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

A PhD thesis proposal presented to the

Business Information System - Helwan University

In Partial Fulfillment of the requirements for the Degree

PhD of Business Information System

Presented by

Ashraf Saied Abd El-Naby
Business Information Systems
Faculty of commerce
Helwan University

Supervised by

Prof. Yahia Mustafa Helmy
Business Information Systems
Faculty of commerce
Helwan University

Prof. Emad Helmy Khalil
National Telecom Regulatory Authority , Egypt

1-Abstract

ERP systems using many business applications like accounting, billing, payroll, financing, purchasing, suppliers, and legacy applications most of these applications are designed to work isolated databases and applications, they are implemented at different platforms and different operating systems. Each program has its own databases and data entry rules, which leads to frequent data entry, the high cost of operating information technology resources. The main problem is the difficulty in sharing and exchange information between ERP applications which becomes a very crucial point for business continuity.

This paper introduces the most attractive solution to achieve integration between heterogonous systems and legacy systems are using Service-Oriented Architecture (SOA) based Web Application to achieve integration between ERP systems.

2-Introduction

The ERP system has a variety of subsystems. Some subsystems are developed by various organizations using different technologies .

When the systems are not designed to work together, it is hard to reuse functionalities from other subsystems. Software developers have to understand multiple different subsystems to make changes to one part of the application and updating one part of the application can have undesirable effects to the performance of the whole system.

This is one of the problems that the service-oriented architecture (SOA) was developed to solve. SOA is a software architecture that informs the way software can be designed. It is not a technology standard and does not rely on particular protocols. It is an architecture blueprint that can be applied using a wide variety of various technologies.

Many studies have identified SOA on the based on various perspectives (such as technology, industry, and architecture), so there is no perfect definition of this concept. SOA is not a technology, a commodity or a simple solution capable of solving the complexities of IT (Niknejad , Ismail , Ghani, Nazari , & Bahari, 2020)

3-Enterprise Resource Planning (ERP)

The exponential development of information and communication technology (ICTs) led by microelectronics, computer hardware and software systems has affected all aspects of computer applications through organizations. It is widely understood by large and medium-sized companies (SMEs) that the opportunity to have the right knowledge at the right time offers immense benefits to companies in a highly strategic world with diverse business practices (Hossain, Patrick, & Rashid, 2002).

Some ERP
software or subsystems exist in separation or legacy systems, making it very challenging to integrat e current systems modules that are
not initially designed to work together The major challenge that is faced organisations is to build the ir application to integrate ERP components. (Saied, 20014)

Enterprise Resource Planning (ERP) applications are made up of various components, including sales, banking, manufacturing and human capital. In order to facilitate data exchange by connecting business processes, these modules are interconnected. (Kähkönen, 2017)

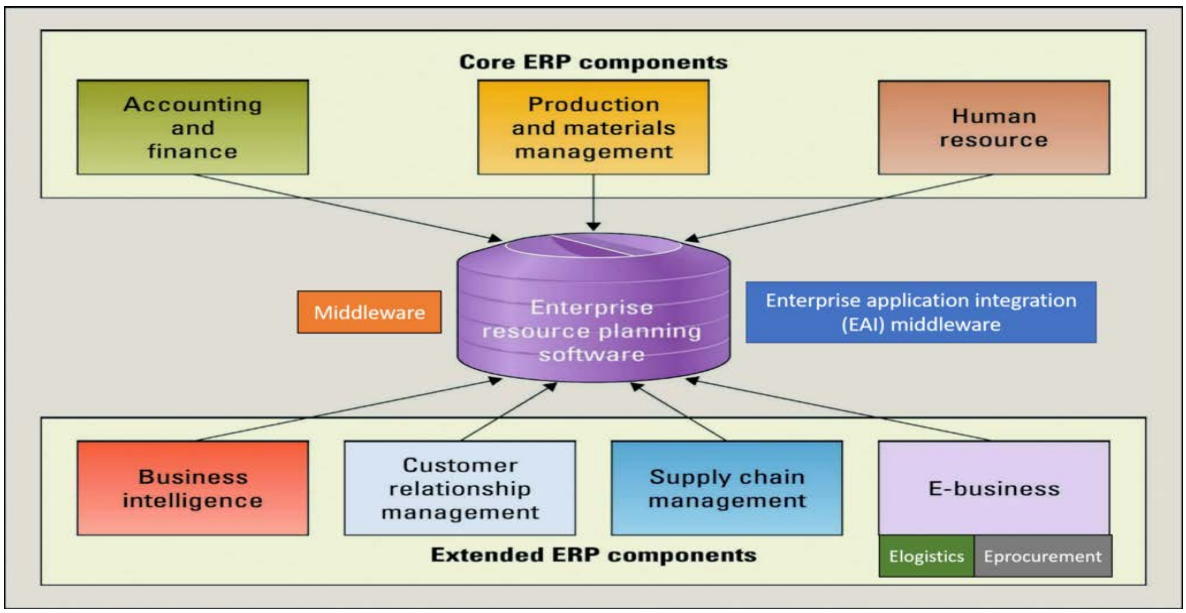

Figure 1-1 -ERP Systems

Over the past decade, a new generation of packaged application software has grown, mostly seeking to consolidate under a different title, a multi-billion dollar industry that comprises the world's fourth largest software vendor, a variety of major technology companies and the world's largest management consulting firms. Commonly referred to as Enterprise Resource Planning Systems (ERP), these hierarchical, streamlined management applications aim to integrate a number of business processes and functions in order to present a holistic view of the business from a single information and IT architecture (Klaus, Rosemann, & Gable, 2000)..

3-1 Evolution Of ERP Systems

The ERP system did not appear suddenly, it appeared since 1960 and has developed in several different stages. This will be briefly explained as follows

1- Inventory Control Packages (1960s) :

The organ of ERP can be traced back to the 1960s, when the emphasis of the applications was primarily on inventory control. Inventory control involves defining inventory needs, setting goals, providing restocking strategies, tracking the distribution of items, reconciling inventory balances, and reporting on inventory status.

2- Material Requirements Planning (1970s)

Materials Requirement Planning (MRP) was developed in the late 1960s and 1970s to deal with inventory control and production management. Based on the manufacturing requirements of the final goods, the design of the production environment, the existing inventory levels and the inventory level process for each operation, and the time limits, MRP establishes production and raw material procurement schedules. This approach has helped to turn the master production plan into individual unit requirements, such as sub-assemblies, parts and other raw material planning and acquisitions.

3- Material Requirements Planning (MRP II) (1980s)

In the early 1980s, MRP was redesigned under the label of Manufacturing Resources Planning or MRPII, MRP II was established. MRP II utilizes SW software to manage production methods, from product planning, parts purchase, inventory management to final product shipment

and distribution. Although MRP-II was an expansion of MRP in the beginning to include shop floor and distribution management operations, MRP-II was further enlarged in later years to include fields such as banking, human resources engineering, project management,

4- Enterprise Resource Planning (ERP) (1990s)

The first ERP started in the 1990s, by using multi-module software SW to maximize the productivity of existing business operations, the key principle of ERP. In order supply visibility and maintenance and transportation, ERP can consolidate business activities through different units such as manufacturing, delivery, administrative, accounting, human resource management, inventory management, project management, service and maintenance, and transportation to provide visibility and control over all business components

5- Extended Enterprise Resource Planning OR ERPII (2000s)

"ERP providers attached more components and functions to the existing systems during the 1990s, giving birth to "extended ERPs." These ERP enhancements include advanced planning and scheduling (APS), e-business solutions such as Customer Relationship Management (CRM) and Supply Chain Management (SCM) solutions The expansion of SCM and CRM helps the company, vendors and consumers to have productive tri-party market partnerships.

3-2 Advantages of ERP systems

1. Reliable access to information: common DBMS, reliable and accurate records, better reporting.
2. Avoid data and processes redundancy Modules view the same data from the central database, avoid different data entry and upgrade operations.

3. Delivery and Period Time Improvement Minimizes recovery and monitoring of delays
4. Cost reduction Time savings, improved control by enterprise-wide analysis of organizational decisions.
5. Easy adaptability Changes in business processes easy to adapt and restructure.
6. Improved scalability Structured and modular design with “addons.”
7. Improved maintenance Vendor-supported long-term contract as

4-Service-Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) is at the heart of a revolutionary computing platform that is being adopted world-wide and has earned the support of every major software provider. In Service-Oriented Architecture: Concepts, Technology, and Design (Erl, Ogrinz, Bieberstein, & Karmarkar, 2009)

Organizations can't afford to jot down off or replace them over night, as a result of they're mission critical additionally they can not afford to develop their entire data systems from scratch in today's business condition.

Many studies have identified SOA on the based on various perspectives (such as technology, industry, and architecture), so there is no perfect definition of this concept. SOA is not a technology, a commodity or a simple solution capable of solving the complexities of IT (Niknejad , Ismail , Ghani, Nazari , & Bahari, 2020) .

A service-oriented architecture (SOA) is a collection of principles and techniques for designing and developing software in the form of interoperable services. in operational behavior and policies like adding new business rules and regulations, in types of business-related events (R. Ibrahim, Kadry, Akil's, & Kalakech's, 2016).

SOA logic is decomposed into smaller, distinct units of logic. Collectively, these units comprise a larger piece of business automation logic. Individually, these units can be distributed (Erl, Ogrinz, Bieberstein, & Karmarkar, 2009).

A service-oriented architecture (SOA) is an architectural style in computer software development in which components of applications provide services to other component parts, typically over a network, through a communication protocol. The principles of service orientation are independent of any company, manufacturer or technology. SOA makes it easier for software components to work with each other across different networks.

The IT technologies are very difficult and dynamic, each application can have its own hardware and software components with its own unique characteristics. As a way , when a system wants to speak or interact with other system(s), there's a problem that happens at the horizontal level, at the same time as the system is modified to new devices and needs to use some functionality from legacy applications, there are different issues that occur at the vertical level. These applications have normally been created on various platforms, technologies and programming languages. Since they have been produced after some time, distinctive applications utilize diverse programming models.

4-1 SOA Principles

A service-oriented architecture (SOA) is a collection of principles and techniques for designing and developing software in the form of interoperable services. in operational behavior and policies like adding new business rules and regulations, in types of business-related events (R. Ibrahim, Kadry, Akil's, & Kalakech's, 2016).

4-1.1 Standardized Service Contract

Standardized contract Standardized service contracts are one of the basic components of SOA. They ensure that the beneficial results of services in the same inventory of services remain consistent with international contract design requirements. Services during the SOA are in a position to define their capabilities and the overall purpose of the contract. It evaluates the char at the same time as the public technical interface of the service is being designed. (Vennapoosa, 2007)

Origin Of Service Contract Contracts are used by various types of automation systems for about as long as information technology has developed, interactions between two standalone software programs are facilitated by a simple design in which the information needed to invoke and share information between the programs is pre-defined and a proper technical specification is provided.

4-1.2 Loose Coupling

Loose coupling is an approach to linking parts of the system or network such certain modules, also referred to as elements, depend upon one another to the smallest amount degree possible. The word "coupling" are often a reasonably straight-forward feature of the IT vocabulary contract that's decoupled from the encompassing environment.

Loose coupling—Services maintain a relationship that minimizes dependencies and only requires that they retain an awareness of each other (Erl, Ogrinz, Bieberstein, & Karmarkar, 2009).

4-1.3 3-Service Abstraction

This principle concerns the abstraction of various service particularities in order to reduce the interaction between consumers of services and therefore the execution of services. In addition, these abstractions make it possible for service providers to switch the library or algorithm in topic without affecting other parts of the service, which in turn contributes to the improvement of the service logic.

4-1.4 4- Service Autonomy

Services have a high degree of control over their internal runtime execution environment. Services are expected to possess influence of the concepts they encapsulate. during which to be autonomous, the service must be as independent as practicable from the other services with which it communicates, both technically and from the purpose of view of the runtime environment)Erl ، Tost ،Roy و ،Thomas(2014 ،.

4-1.5 5-Service Reusability

Definition of the re-use service as having the capacity to communicate in several service assemblies (compositions). Service reuse is closely connected to software composability (Feuerlicht & Lozina, 2014)

The concept of service reusability can be a development process applicable to the development of services within the context of a business orientation architecture that can be reused in the business. Reusability of Services-Concept is separated into resources with the goal of maximising reuse.

4-1.6 6-Service Statelessness

This means that the services do not carry information from one state to the another. We can get a web service giving you the value of a given object. Services decrease the usage of money and eliminate state data. Ideally, the service is to be stateless. Programs of Operation Statelessness would preferably be stateless.

4-1.7 7-Service Discoverability

The principle of discoverability should be applied to the design of a private service in such a way as to make it as discoverable as possible if the extension of discoverability or product is actually taking place within the current setting of implementation. The idea of discoverability should be extended to the planning of a private service in such how that it's as discoverable as possible whether the extension of discoverability or the merchandise currently occurs within the current implementation setting.

4-1.8 8-Service Composability

Composition of services means the creation of latest services by the mixing of existing services. Services must be compassable so as to integrate smaller services with larger value-added services. Composability integrates design principles to make sure that programmers are ready to communicate during a number of variations so as to unravel variety of biggest problems.

4-2 Benefit Service Oriented Architecture

1- -Integrating and linking to other structures is a dominant approach
2- The issue of data transmission between systems is overcome through integration

3- -Enterprise architects claim that the SOA will help companies adapt more efficiently and cost-effectively to changing market conditions.
4- The sophistication of applications concealed from consumers
5- -This type of architecture uses macro level reusability instead of micro level integration and SOA
6- -Flexible structures are not an obstacle to the evolving and rapid evolution of processes;
7- -Dividing the project into smaller parts that can be completed separately is easy.
8- -Monitor improvement in each sub-project is determined
9- -Systems are easy to satisfy the specifications of users
10- -Solving the dynamic integration of large networks
11- New products or procedures are simply carried out

5- Applying Service Oriented Architecture on Enterprise Resource Planning Environment

Enterprises are generally made up of hundreds, if not thousands, of custom-built software, purchased from third parties, part of legacy systems, or a variety of them, which run on multiple-tiered architectures of multiple operating systems (Solving Integration Problems using Patterns, n.d.).

In order to run their enterprises, companies have invested heavily in large-scale software applications such as ERP (enterprise resource planning), SCM (supply chain management), CRM (customer relationship management) and other such systems. Corporation and IT administrators are frequently asked to develop enhanced service functionality while utilizing current IT resources (MAHMOOD, 2007).

ERP applications are usually consisting of hundreds of applications that work in various configurations, run on different operating systems and platforms, which are created custom, obtained from third parties, part of or a mixture of these systems. On the mainframe, in databases, in unknown legacy systems, in spreadsheets on desktops, in enterprise resource planning (ERP) systems, in message queues, in flat files, data is distributed everywhere.

To solve the interoperability, reusable problems, an effort was started to create standards for different systems to communicate in a well-known way, detached from any proprietary technology. Nowadays, Web Services are the most commonly used standard to perform this task

Data integration enables businesses to access all their scattered data, to establish reliable and clear views of their key information properties and quickly use these data for strategic decisions and activities within the organization. Data integration is also used to synchronise information through operating processes and systems and to create scalable and reusable data resources.

Service-oriented architecture (SOA) is an architectural approach that makes use of resources accessible inside the network by applications. In that same architecture, services are accessed by a web-based communication by the internet (Garg, 2020).

6-SOA Technologies & Web Services

new concept to classify SOA. In the late 1990s, Sun defined SOA as Jini, an ecosystem for dynamic exploration and use over a network of services. The concept of services introduced by Jini technology has been embraced by web services and adopted as web services using technologies such as XML, Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP) and Universal Description, Discovery and Integration (UDDI).

SOA is not a new idea. In the late 1990s, Sun described SOA to characterise Jini, which is an environment for dynamic discovery and usage over a network of services. The definition of services introduced by Jini technology has been picked up by web services and adopted as web services using technologies such as XML, Web Services Description Language (WSDL), Basic Object Access Protocol (SOAP) and Universal Description, Discovery and Integration (UDDI).

SOA can help enterprises streamline processes so that they can more efficiently transact business and adapt to shifting demands and competition, facilitating the definition of software as a service. For example, eBay is opening its API for its web services for its online marketplace. (Mahmoud, 2005) .

6-1.1 Web Service Components

One of the main solutions is to build architectures that allow fast integration of current and new enterprise applications. Web Services (WS) technologies and SOA provide better options for enterprise application integration with the additional advantages of lower costs, faster servicing, increased reliability and enhanced scalability. With its loosely coupled structure, SOA helps companies to connect to new services or update existing services and to create opportunities to improve market flexibility and react to demand.

The Web Services (WS) technologies and SOA have improved opportunities for business application adoption with the added advantages of lower costs, quicker development, greater stability and improved scalability. SOA helps businesses to connect to emerging services or upgrade existing services through its loosely linked existence, providing ways to increase business resilience and respond to demand.

Web services are software networks developed to facilitate machine-to-machine connectivity over a network that is interoperable. This interoperability is gained from a series of open standards based on XML, such as WSDL, SOAP, and UDDI. These principles offer a popular approach for web resources to be established, published and used (Mahmoud, 2005)

Three specific operations are then required for a Web Services architecture: publish, find, and bind. Providers of services publish services to the broker of the service. Service requesters use a service broker to find the services needed and bind them. The following figure illustrates these theories.

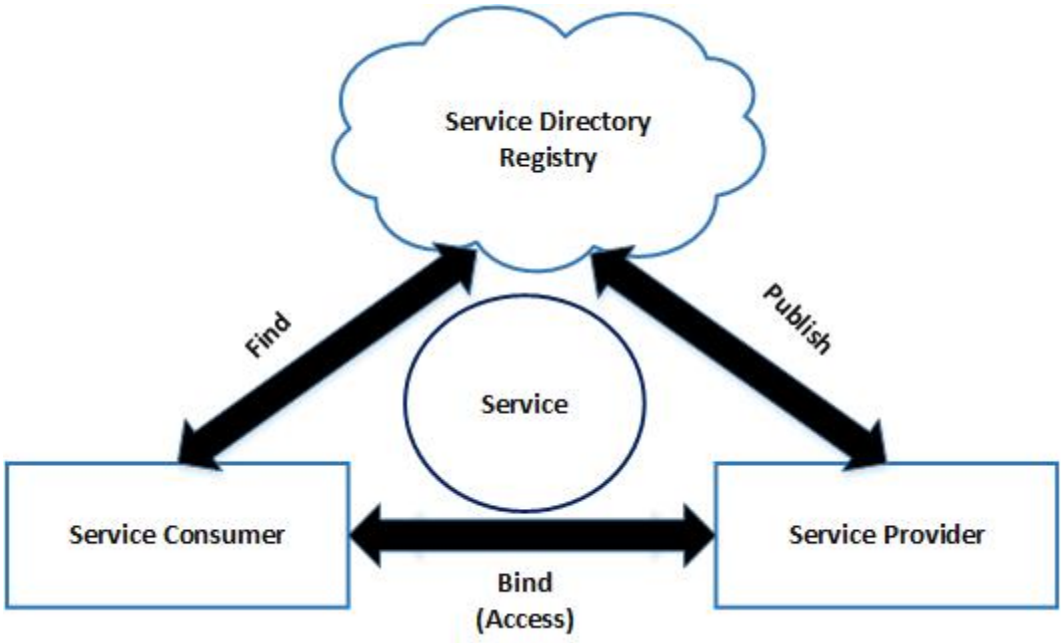

Figure 2 Service Provider, service consumer and service directory partnership

- **Service Provider:** produces services he needs to deliver to the outside world and publishes such services to the UDDI registry. The service provider uses a Universal Discovery, Description, and Integration (UDDI) directory to create and publish a service description as the generic requirement for services to be discovered over the network by its applicants.
  A service provider
- **A Service Requestor:** looks at the UDDI register for the services and communicates with the Service Provider for the services he needs to use.May be pointed to as a service consumer/requestor which uses SOAP (Simple Object Access Protocol) messages to find a service description stored inside a UDDI by a service provider and binds or invokes the service.
- **Services Register**: This acts as an interface between the service provider and the requesting service provider by connecting the two services together. Service Register UDDI is the directory where the service provider publishes the overview of the service. The UDDI registry is a central repository that manages the Published Services Registry.

6-2Web Services Technologies

Web Services are based upon three main technologies WSDL (Web Service Description Language), UDDI (Universal Description and Discovery Integration) and SOAP(Simple Object Access Protocol)

6-2.1 **Web Service Description Language (WSDL)**

The WSDL text was also XML-designed based on W3C-specified specifications. The WSDL specifies what the service is doing, how to use it, and where it is based.

WSDL (Web Service Definition Language) is an abstract interface that allows a web service to connect to any programming language. Thus, if Java implements the procurement component and C Sharp implements the accounting component, the purchasing order will be exposed as a WSDL

request that will be an abstract interface to the accounting component to communicate without respecting the technologies used.

The Web Services Description Language (WSDL) is an XML grammar used to describe Web services as a set of network endpoints that operate on messages. A overview of the service WSDL provides an abstract collection of operating and message definitions, a concrete linking protocol and network endpoint parameters for binding operations and messages (Christensen, Curbera, Meredith, & Weerawarana, 2001) .

The WSDL document has different components, but they are found within these three key elements, which can be established as separate documents and can then be integrated or reused to form total WSDL files.

The three key elements of the WSDL that can be identified separately are-:

- Types :(data types of parameters to be passed to and to be retrieved from the Web Service)
- Operations: (an abstract description of an action supported by the service)
- Binding: (a concrete protocol and data format specification for a particular port type.)

6-2.2 Universal Discovery, Description and Integration UDDI

Universal Discovery, Description and Integration (UDDI): The UDDI was created for the purpose of discovering services published to the registry. The UDDI serve as pointer to where to locate a particular Web service

UDDI includes a mechanism for publishing and locating service descriptions, which is invoked if a text (usually from a requesting service) conforms to the explanation provided in any of the directories stored in the UDDI.

6-2.3 Simple Object Access Protocol SOAP

SOAP is the most commonly used technology for online services available. SOAP is not limited to a single transportation protocol such as HTTP, unlike most web server technologies. While, due to its more features and less drawbacks, SOAP can be considered to be the most powerful web service technology, it is often considered to be more complex and slower than other web service technologies (Sud, 2010).

For sharing XML data over the Network, SOAP offers a simple and lightweight protocol. SOAP helps client programmers to quickly connect and invoke remote approaches for remote services. SOAP messages are written exclusively in XML and are thus special relative to the protocols used for CORBA, DCOM, and Java RMI, applications, and language-independent platforms.. Machines will quickly understand the data being sent from a service requestor with SOAP messages in XML format and respond by sending SOAP messages..

7- Applying Service Oriented Architecture on Enterprise Resource Planning Environment

In terms of software terminology, there is a lot of confusion when it comes to software terminology. ERP and SOA are very different but may be used in the same sentence when referring to enterprise systems, which adds to the confusion.

7-1 The Difference Between ERP and SOA

ERP and SOA are quite distinct but can be found in the same context when referring to enterprise systems, which leads to the misunderstanding.

There is tons of confusion about software terminology . ERP and SOA are quite distinct, but when pertaining to enterprise systems, they will be found within the same statement, which contributes to the misunderstanding (Bortolus, n.d.)

There is a lot of confusion when it comes to software terminology. ERP and SOA are very different but may be used in the same sentence when referring to enterprise systems, which adds to the confusion.

SOA ERP: Service-oriented architecture can be located at the center of the ERP system. Similar data, shared by different modules or functional areas, is often stored in a single place in ERP systems. They are not always in the same environment, but they are more likely to find SOA in integrated systems. SOA is a facilitator for any ERP module and SOA usage is relevant not only to ERP, but also to other business functions such as Legacy Programs, CRM, SCM …etc. .In general, SOA may be referred to as a software development tool. The basic function behind the use of SOA is reusability . (Behera, Awasthi, Qureshi, Shanbhag, & Jain, 2011).

SOA stands for Service-Oriented Architecture. SOA is a software development tool. SOA's main concept is to write programming code as few times as possible. Unique functions done in various programmers are set up as "Service" objects. A programme that handles a larger purpose may "Call" that particular task by sending input and receiving output, or by having that output stored in a database. Simple example: "Address" is a data set common to several business functions. Address data for staff, clients, suppliers and contacts may all use the same logic of the software to validate and format the data . https://itstillworks.com/difference-between-erp-amp-soa-6106695.html

The implemented ERP system would be effective in carrying out business transactions regarding human resource, payroll, attendance management, accounting, procurement, etc. The key purpose of ERP is to integrate all roles/functions in an organization so that they share information with each other.

7-1.1 Differences:

- SOA is an integration paradigm that is based on a fundamental design principle and provides architectural interoperable services - hence its name..

-SOA is a tool that is simple to device developers is an architecture that connects several functions

- SOA is an ERP subset, providing a key reusability functionality on hand, ERP is more like linking functionality on its own, such as Accounting, Vacancy Management, Banking, etc.

7-1.2 ERP programmers are communicating with and without SOA

ERP programmes need to communicate at all levels such that synchronised information is accessible for use in all agencies, and this may be achievable with with and without SOA. There are some conditions before the two vendor's ERPs communicate with each other and they are, Report Style, the format in which the report is to be submitted from one vendor to another should be appropriate to the target vendor's ERP.

Field Concepts All ERP structures can either run using the same fields or have an analogue present in the target ERP referring to the ERP source.

With SOA: ERP systems should be more disintegrated so that there could be a standard input feature. The employee id is an example that is being used globally so that the employees can be handled on all channels.There are also several monitoring instruments, such as Company Object Files, SQR, which could easily be integrated with various ERP programmers, such as Oracle PeopleSoft, SAP, and Siebel.

8-Conclusion

This study presents a summary of ERP characteristics, stages of development, advantages, disadvantages, and problems resulting from the multiplicity of its applications and their differences in databases, operating systems, connectivity and integration.

The study contributes to explaining SOA, its stages of development, and the principles it follows when integrating ERP applications using the web service.

The Web Services enable a ERP applications components to exchange information seamlessly d between heterogeneous applications. The data integration between the present running systems and legacy systems can be achieved by the web services .

Service-oriented architecture is used web services allow for a distributed environment in which any number of applications or system components easily communicate between non-platform-dependent and non-language-dependent organizations. Data integration between existing systems and legacy systems can be achieved in a cost-effective manner through web services

9-References

AbdAllah, M. M. (213). A Quick Introduction to SOA. *Software Engineering Competence Center*.

Bah, A., André, P., Attiogbé, C., & Konaté, J. (2019). Federated Access Control in Service Oriented. *Semantic Scholar*.

Beal, V. (2003). *loose_coupling*. Retrieved from Webopedia.com: https://www.webopedia.com/TERM/L/loose_coupling.html

Bean, J. (210). *SOA And Web Services Interface Design Principles , Techniques And Standard.* Morgan Kaufmann publications.

Behera, D., Awasthi, I., Qureshi, I., Shanbhag, S., & Jain, V. (2011). *Does SOA adds redundancy to an ERP Implementation*. Retrieved from soavolution: https://soavolution.wordpress.com/about/

Bortolus, D. (n.d.). *What Is the Difference Between ERP & SOA?* Retrieved from itstillworks: https://itstillworks.com/difference-between-erp-amp-soa-6106695.html

Brodkin, J. (2010, April). *Service-orientation design principles*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Service-orientation_design_principles#cite_note-Pitfalls-

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *Web Services Description Language*. Retrieved from W3C: https://www.w3.org/TR/wsdl.html#:~:text=WSDL%20is%20an%20XML%20format,format%20to%20define%20an%20endpoint.

Chu, P., Park, M., Chevtsov, S., & Wu, J. (2010). SERVICE ORIENTED ARCHITECTURE FOR HIGH LEVEL APPLICATIONS. *Research Gate*.

Erl, T. (2008). *SOA Principles Of Service Design.* Prentice Hall.

Erl, T., Merson, P., & Stoffers, R. (2017). *Service-Oriented Architecture Analysis and Design for Services and Microservices.* Mark Taub.

Erl, T., Ogrinz, M., Bieberstein, N., & Karmarkar, A. (2009). *SOA_eBook-Inform IT*. Pearson E ducation, Inc.

Erl, T., Tost, A., Roy, S., & Thomas, P. (2014). *SOA with Java Realizing Service-Orientation with Java Technologies.* Arcitura Education Inc.

Feuerlicht, G., & Lozina, J. (2014). Understanding Service Reusability. *ResearchGate*.

Garg, S. (2020, 12). *Service-Oriented Architecture*. Retrieved from Geeks ForGeeks: https://www.geeksforgeeks.org/service-oriented-architecture/

Hossain, L., Patrick, J. D., & Rashid, M. (2002). *Enterprise Resource Planning: Global Opportunities & Challenges.* Idea Group Publishing.

Kähkönen, T. (2017). *UNDERSTANDING AND MANAGING ENTERPRISE SYSTEMS INTEGRATION.* Lappeenranta University of Technology.

Klaus, H., Rosemann, M., & Gable, G. (2000). What is ERP? *Research Gate*.

Kohlborn, T. (2008). *A consolidated approach for service (Master's Thesis).* Universitat Munster.

Laird, R., Falkl, J., Cocks, S., & Clark, D. (2018, 08). *SOA Policy*. Retrieved from IBM.COM: bm.com/developerworks/library/ws-soa-policy/ws-soa-policy-pdf.pdf

Lewi, G., Morris, E., Smith, D., Simanta, S., & Wrage, L. (2007). Common Misconceptions About Service-Oriented Architecture. *Springer*.

MAHMOOD, Z. (2007). Service Oriented Architecture: Potential Benefits and Challenges. *11th WSEAS International Conference on COMPUTERS.* Greece.

Mahmoud, Q. H. (2005). *Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI).* Retrieved from ORACLE: https://www.oracle.com/technical-resources/articles/javase/soa.html#:~:text=Web%20services%20are%20software%20systems,publishing%2C%20and%20using%20web%20services.

Merson, P., & Lewis, G. (2008). Service Level Agreements in Service-Oriented Architecture Environments. *ResarchGate*. Retrieved from https://www.researchgate.net/publication/235103135

Miller, D. (1989). *Abstractions in Logic Programs.* University of Pennsylvania Department of Computer and Information Science.

Niknejad , N., Ismail , W., Ghani, I., Nazari , B., & Bahari, M. (2020, January). Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation. *Elsevier*. Retrieved from https://reader.elsevier.com/reader/sd/pii/S0306437920300028?token=8EBDFA74FE090A1B766CC689E7CDBD3584723B060ACC29381B8E5DC3A00AEC5871533226D551D650DD36D9DCDDF3EA93

Perumal, T., Leong, C., & Ramli, A. (2014). SOA-Based Framework for Home and Building Automation Systems (HBAS). *International Journal of Smart Home* .

Poulin, J., & Himler, A. (2006). *The ROI of SOA Based on Traditional Component Reuse.* LogicLibrary, Inc. .

R. Ibrahim, R., Kadry, S., Akil's, K., & Kalakech's, A. (2016). A New Solution for Service Oriented Architecture Versioning. *Asian Journal of Computer Science And Information Technology*.

Ricken, J. (2013). *PhD: SOA Method Engineering Framework for process oriented and model-driven SOA implementation with underlying models of abstraction.* University of Namur.

Saied, A. (20014). *ERP Modeling : A Management Integrated Approach (An Applied Study).* Sadat Academy for management Sciences Faculty Of Management Science Computer and information System Department.

Sarkarati, M., Merri, M., & Spada, M. (2012). Intrinsic Interoperability Of Services A Dream Or A Key Objective For Mission Operation Systems. *Researchgate*.

Solving Integration Problems using Patterns. (n.d.). Retrieved from Enterprise Integration Patterns: http://www.eaipatterns.com/Chapter1.html

Stack, C. (20006). *The future is service-oriented architecture*. Retrieved from Network World: https://www.networkworld.com/article/2308447/the-future-is-service-oriented-architecture.html

Sud, M. (2010). *Master:Sensitivity in Service Design for the Development.* Canada: University of Windsor.

The Open Group. (2009). *SOA Source Book.* Van Haren.

Vennapoosa, C. (2007). Retrieved from http://www.exforsys.com/author/Exforsys.