



SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF POST GRADUATE STUDIES

Integration of Enterprise Resource Planning Systems with Java Based Application

تكامل أنظمة تخطيط موارد المؤسسة باستخدام برنامج الجافا

**This research is submitted in partial fulfillment of the academic requirement
for the degree of Master in Computer Science**

PREPARED BY

Alsheikh Ahmed Anwar Altayeb

THE SUPERVISOR

Dr. Nisreen Beshir Osman

September 2018

الآية

(وَلَا تَقُولَنَّ لِشَيْءٍ إِنِّي فَاعِلٌ ذَٰلِكَ غَدًا إِلَّا أَنْ يَشَاءَ اللَّهُ ۗ وَادْكُرْ رَبَّكَ إِذَا نَسِيتَ وَقُلْ
عَسَىٰ أَنْ يَهْدِيَنِّي رَبِّي لِأَقْرَبَ مِنْ هَٰذَا رَشَدًا)

(18 الكهف آية 23-24).

Dedication

I dedicated this thesis:

To my interest and reasons of my presence ... Dear Mother Sumaya mohamed El Hassan which gave more than what should be ... Dear Father Ahmed Anwar Al Tayeb

To the spirit of the martyr brother ... Muhammad Ahmad Anwar al-Tayeb ... We ask God to grant him the highest Paradise and that does not deprive us of the reward and do not Sedition after his death

To the most precious thing I have in my life ... The sisters Walaa and Aisha

To my friends and colleagues who are happy to be near them

And finally, to our teachers and supervisors who supervised the completion of our education and did not spare us the advice ... And to all those who contributed to completion this work ...

Acknowledge

After achieve this degree Master in Software Engineering and Quality assurance, I have to thank all those who contributed to this success and thank all the brothers, sisters, friends, colleagues and thank a lot also to University of Sudan for Science and Technology and its platforms, which never gave up to provide students with science and knowledge and more thanks also to the direct supervisor Dr. Nisreen Beshir Osman gave me the advises many thanks and gratitude to the dear mother and father.

Abstract

All (Erp) systems work primarily on the restructuring of companies and institutions and the elimination of old systems, but recently became the concept of integration between systems is common, because building new systems and canceling the old one waste the effort and money There is also a lack of information exchange between ERP and legacy systems in one organization, which results in the lack of reporting of the decision makers.

Old systems can be linked with (Erp) in order to avoid loss of time and effort and reduce the cost of building comprehensive systems. in addition, an integration model can be used to ensure that the integration process has been done correctly, the main objective of this study is to propose a model that can be used for link ERP system with the old systems already in place. In this framework, the model defines the steps necessary for integration between existing systems such as how to identify the points of integration, as well as the selection of appropriate techniques and other steps.

Two systems were selected and then linked using SOA technology to achieve the effectiveness of the proposed model, the implementation results showed a high rate of scalability, compatibility and performance.

the result of the implementation of the model are the integration of the two systems and the exchange of information with high performance using restful technology as the best technique for performance.

المستخلص

جميع أنظمة (ERP) تعمل في الأساس على إعادة هيكلة الشركات والمؤسسات وإزالة انظمتها القديمة ولكن مؤخراً أصبح مفهوم التكامل بين الأنظمة شائعاً لأن بناء الأنظمة الجديدة وإلغاء القديمة به هدر للجهد والمال، هنالك نقص أيضاً في تبادل المعلومات بين (ERP) والأنظمة القديمة في المؤسسة الواحدة والذي يتولد منه نقص إستخراج التقارير لمتخذى القرار. يمكن ربط الأنظمة القديمة مع (ERP) وذلك لتفادي ضياع الوقت والجهد وتقليل تكلفة بناء الانظمة الشاملة، بالاضافة الى ذلك يمكن استخدام نموذج تكامل لضمان أن عملية التكامل قد تمت بشكل صحيح. الهدف الرئيسي من هذه الدراسة هو إقتراح نموذج يمكن إستخدامه لكيفية ربط نظام (ERP) مع الأنظمة القديمة الموجودة مسبقاً. وفي إطار ذلك يحدد النموذج الخطوات اللازمة لعملية التكامل بين الانظمة الموجودة مثل كيفية تحديد نقاط التكامل وأيضاً إختيار التقنيات المناسبة وغيرها من الخطوات. تم إختيار نظامين ومن ثم تم ربطهما بإستخدام تقنية البنية الخدمية (SOA) وذلك للتحقق من فعالية النموذج المقترح، نتيجة لتنفيذ النموذج ، حصلنا على تكامل بين النظامين وتبادل المعلومات مع الأداء العالي باستخدام تقنية (RESFUL). اظهرت نتائج عملية تنفيذ النموذج معدل عالي في كل من قابلية التوسع والتوافق والأداء.

Table of contents

الآية	i
Dedication	ii
Acknowledge	iii
Abstract	iv
المستخلص	v
Table of contents	vi
List of Figures	ix
List of Abbreviatwns.....	x
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Problem Statement.....	1
1.3 Significance	2
1.4 Objectives	2
1.5 Scope	2
1.6 Thesis Layout	2
Chapter 2: Literature review	4
2.1. Introduction.....	4
2.2. Literature Review.....	4
2.2.1. Enterprise Recourse Planning - DEFINITION.....	4
2.2.2. (ERP) Architectures.....	4
2.2.3. Two-tier Implementations	5
2.2.4. Enterprise Recourse Planning Features (ERP).....	6
2.2.5. Definition of Open Source.....	7
2.2.6. Evolution Of Enterprise Resource Planning.....	8
2.2.7. Open Source Software (ERP) Systems.....	9
2.2.8. Enterprise Integration	10
2.2.9. Enterprise Application Integration	10
2.2.10. Integration Standard	12
2.2.11. Service Oriented Architecture (SOA)	12

2.2.12.	Application Programming Interface (API).....	14
2.2.13.	API-management.....	15
2.2.14.	APIs vs. SOA	15
2.2.15.	Web services	16
2.3.	Related work	16
2.3.1.	Integration of Distributed Enterprise Applications A Survey (Wu He, Li Da Xu 2014)	16
2.3.2.	Using SOA with Web Services for effective data integration of Enterprise Pharmaceutical Information Systems (Kester, Ajibade 2013)	17
2.3.3.	Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics (Tihomirovs, Juris 2016).....	18
2.3.4.	Integration of an ERP System a case study on integration challenges with Microsoft Dynamics AX (Xu, Annie 2017).....	18
2.4.	Conclusion	19
2.5.	Summary	19
Chapter 3:	Methodology	20
3.1.	Introduction.....	20
3.2.	Methodology	20
3.2.1.	Identify the Systems to Be Integrated.....	21
3.2.2.	Selecting the Integration Architecture	21
3.2.2.1.	EAI	21
3.2.2.2.	SOA.....	22
3.2.2.3.	ESB.....	22
3.2.3.	Choose an integration method and technology	23
3.2.4.	Design of the proposed model	24
3.2.4.1.	Apply the Model on target Systems	25
3.3.	Summary	25
Chapter 4:	Model implimintation.....	26
4.1.	Introduction.....	26
4.2.	Tools.....	26
4.2.1.	NetBeans IDE	26

4.2.2.	Notepad++	26
4.2.3.	WampServer	27
4.2.4.	Odoo ERP	27
4.3.	Programming Language and Technology	28
4.3.1.	Web Services	28
4.3.1.1.	Web Service Description Language (WSDL).....	28
4.3.2.	Java 2 Platform, Standard Edition (J2SE)	29
4.3.3.	(JSON) & (XML)	30
4.4.	Implementation	31
4.4.1.	Identify the Systems to Be Integrated.....	31
4.4.1.1.	build services from the systems	32
4.4.1.1.1.	Civil Registry system.....	32
4.4.1.1.2.	Customer registration system.....	33
4.4.2.	Selection The Integration Architecture.....	36
4.4.3.	Integrate services with RESTFUL:.....	36
4.5.	Result.....	36
4.6.	Summary	37
5.1.	Conclusions	38
5.2.	Recommendation	38
5.3.	Reference.....	38

List of Figures

FIGURE 2. 1 TWO-TIER ERP SYSTEMS ARCHITECTURE [2].	5
FIGURE 2. 2 SUMMARIZES THE 3-TIER ARCHITECTURES OF ERP [2].	6
FIGURE 2. 3 SUMMARIZES THE HISTORICAL EVENTS RELATED WITH ENTERPRISE RESOURCE PLANNING (ERP) [4].	9
FIGURE 2. 4 EAI INTEGRATION VS TRADITIONAL INTEGRATION [6].	11
FIGURE 2. 5 API VS. SOA [7].	16
FIGURE 3. 1 SHOW THE STEPS THAT ADOPTED FOR INTEGRATION NEED.	20
FIGURE 3. 2 PROPOSAL MODEL OF INTEGRATION ODOO ERP WITH LEGACY	24
FIGURE 4. 1 REPRESENT THE LOGO OF LEGACY CUSTOMS SYSTEM	31
FIGURE 4. 2 REPRESENT THE MODULES OF LEGACY CUSTOMS SYSTEM.	31
FIGURE 4. 3 REPRESENT THE ADDING DATA MANUAL IN LEGACY SYSTEM.	32
FIGURE 4. 4 REPRESENT THE INFORMATION OF PEOPLE IN JSON	33
FIGURE 4. 5 REPRESENT THE INFORMATION OF PEOPLE IN JSON	33
FIGURE 4. 6 INFORMATION ABOUT WEB SERVICES AND WSDL.	34
FIGURE 4. 7 LOGGING OF ODOO ERP.	35
FIGURE 4. 8 MODULES OF ODOO ERP.	35
FIGURE 4. 9 THE NEW MODULES OF ODOO ERP TO GET INFORMATION ABOUT CUSTOMER	35
FIGURE 4. 10 REPRESENT THE DATA AFTER GET FROM WEB SERVICE.	35

List of Abbreviations

Abbreviation	Terms
SOA	Services oriented architecture
IEEE	Institute of Electrical and Electronics Engineers
EAI	Enterprise Application Integration
HTTP	Hypertext transfer protocol
XML	Extensible Markup Language
WSDL	Web Services Description Language
SaaS	Software as a Service
CC	Cloud Computing
APIs	Application-programming interfaces

Chapter One

Introduction

Chapter 1: Introduction

1.1 Overview

Enterprise resource planning (ERP) is a term denoting software that automates a broad range of activities that help a business manage its responsibilities [23]. Many enterprises acquire disparate systems and applications over the years. This application may be having a different architecture, run in different environment and built by different language. There is a lack of existing systems to provide some needs of workflow, the development of existing systems to meet the business requirements need high cost. Lack of information sharing between existing systems because the lack of connectivity between them so they need to implement (Erp) to achieve business requirements and needs quickly and efficiently. Integrating of(ERP) with other systems has attracted much attention in the enterprises and information system area.

Advantages of research provide proposal model to integration (ERP) with other systems already exist.

1.2 Problem Statement

The problems of this research are as follows:

- There is a Lack of information sharing between existing systems and (ERP) in the enterprises.
- Different systems and applications with different architecture, run in different environment and built by different language without connectivity (ERP) causes lack of information or integrated reports for the management and the decision makers.
- The change and develop existing systems to satisfy business processes Loss of effort time and money.

1.3 Significance

This Research help to Take the capabilities of Enterprise resource planning (ERP) with no cancellation of existing systems by provide a model to integration these systems with the (ERP).

Ease of linking different systems developed in different languages and working in different operating environment.

Ease of dealing with the Enterprise resource planning (ERP) and link with other systems. Save time and effort exerted in the development of systems and linking them. Include all systems that institutions and companies needs in to a unified program

1.4 Objectives

The aim of the work is the integration (ERP) with other systems to achieve the following objectives:

1. design a model to link the different system with the (ERP) system.
2. selection of appropriate method for integrating legacy systems with (ERP)
3. Avoid modification in legacy programs that cost money and effort.

1.5 Scope

In this research, the focus will be on building the proposed model for open ERP (Odoo ERP) and other systems based on java.

1.6 Thesis Layout

Chapter One: Contain introduction about the project, defining the problem, objectives, methodology and scope.

Chapter Two: Contain a Literature Review and related work.

Chapter Three: Contain the methodology and how we built the proposal model.

Chapter Four: Contain the implementation of proposal model and result.

Chapter five: is the conclusion and recommendations.

Chapter Two

Literature Review & Related Work

Chapter 2: Literature review

2.1.Introduction

In this chapter, relevant theoretical concepts, Literature Review and related work to the research problem will be presented.

2.2.Literature Review

2.2.1. Enterprise Recourse Planning - DEFINITION

An Enterprise resource planning system is a fully integrated business management system covering functional areas of an enterprise like Logistics, Production, Finance, Accounting and Human Resources. It organizes and integrates operation processes and information flows to make optimum use of resources such as men, material, money and machine [1].

Enterprise resource planning promises

- one database
- one application
- one user interface
- for the entire enterprise, where once disparate systems ruled manufacturing, distribution, finance and sales.

Other definition “Enterprise Resource Planning” is a term originally coined in 1990 by The Gartner Group to describe the next generation of Material Requirements Planning (MRP) software. The purpose was to integrate all facets of the business enterprise under one suite of software applications. The definition of ERP would be broadened to include almost any type of large integrated software package [2].

2.2.2. (ERP) Architectures

When talk about architecture of ERP it is also necessary to know how an ERP architecture works and how the system has been deployed in an organization. While the

servers may be centralized, the clients are usually spread to multiple locations throughout the enterprise, although ERP applications are most commonly deployed in a distributed and often widely dispersed manner. ERP software and the results are presented to the end user through some user interface. The architecture that supports the connection between the database, processing, and presentation is called Client/Server architecture. The two most commonly implemented architectures are outlined below [2].

2.2.3. Two-tier Implementations

In typical two-tier architecture, the server handles both application and database duties. The clients are responsible for presenting the data and passing user input back to the server. While there may be multiple servers and the clients may be distributed across several types of local and wide area links, this distribution of processing responsibilities remains the same. Figure [2.1] summarizes the two-tier architecture of ERP.

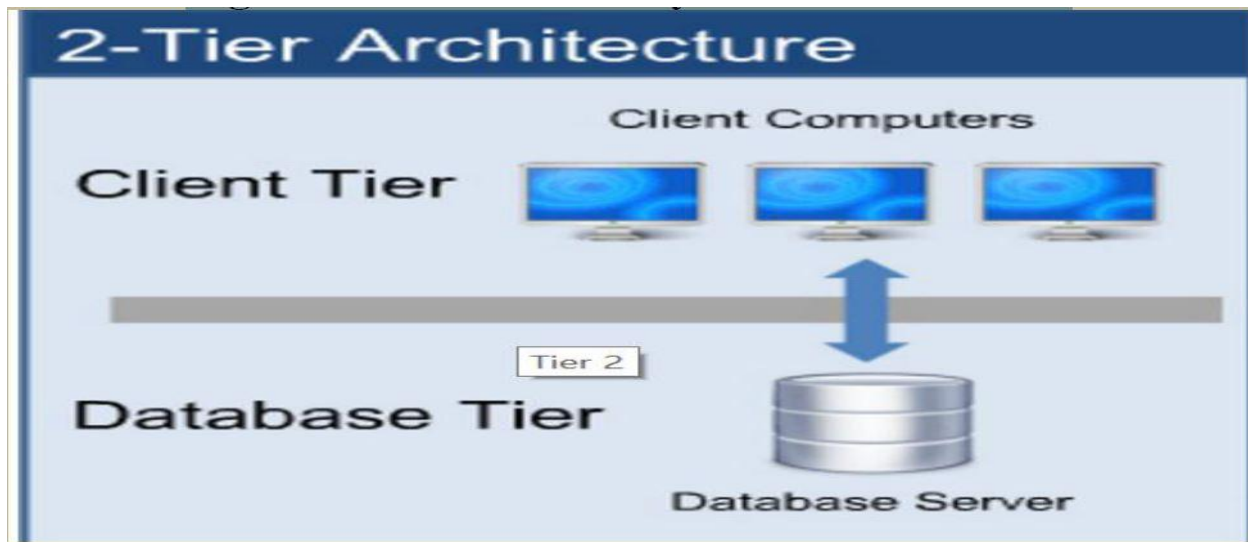


Figure 2. 1 two-tier ERP systems architecture [2].

In three-tier architectures, the database and application functions are separated. This is very typical of large production ERP deployments. In this scenario, satisfying client requests requires two or more network connections. Initially, the client establishes communications with the application server. The application server then creates a second connection to the database server.

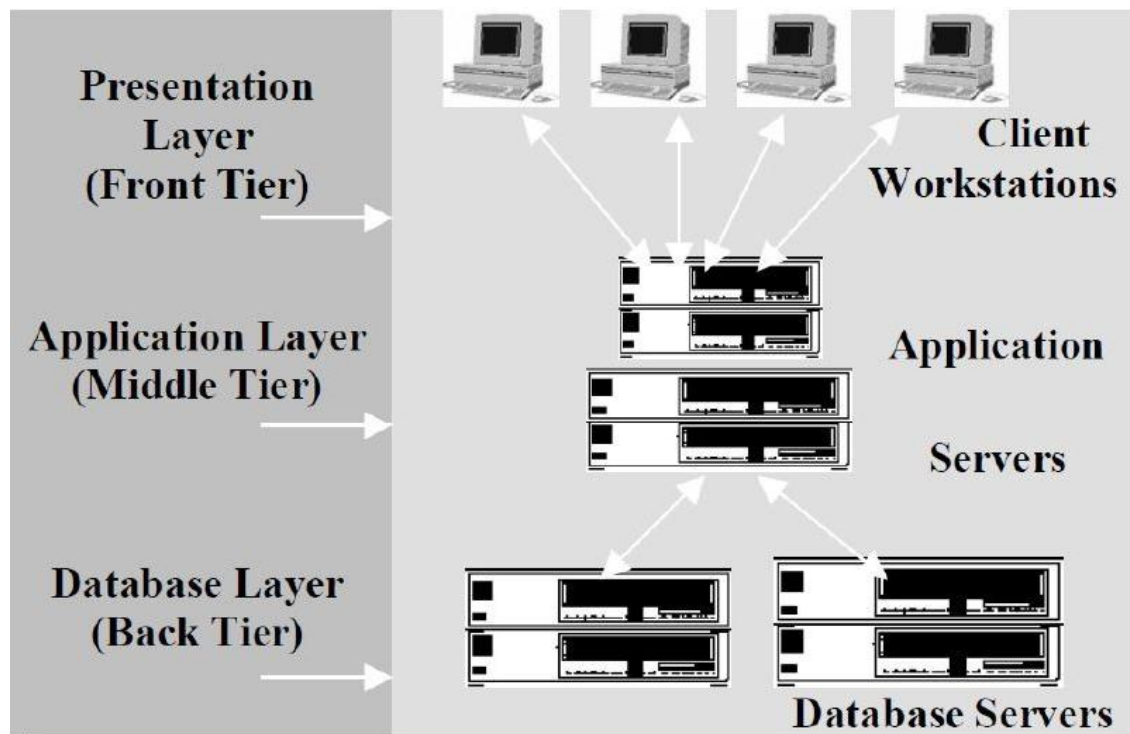


Figure 2. 2 summarizes the 3-Tier architectures of ERP [2].

2.2.4. Enterprise Recourse Planning Features (ERP)

Some of the major features of ERP and what ERP can do for the business system are:

- ERP provides multi-platform, multi-facility, multi-mode manufacturing, multi-currency, multi-lingual facilities.
- It supports strategic and business planning activities, operational planning and execution activities, creation of Materials and Resources.

- ERP covering all functional areas like manufacturing, selling and distribution, payables, receivables, inventory, accounts, human resources, purchases etc.
- ERP performs core activities and increases customer service, thereby augmenting the corporate image.
- ERP bridges the information gap across organizations.
- ERP provides complete integration of systems not only across departments but also across companies under the same management.
- ERP is the solution for better project management.
- ERP allows automatic introduction of the latest technologies like Electronic Fund Transfer (EFT), Electronic Data Interchange (EDI), Internet, Intranet, Video conferencing, E-Commerce etc.
- ERP eliminates most business problems like material shortages, productivity enhancements, customer service, cash management, inventory problems, quality problems, prompt delivery etc.
- ERP provides intelligent business tools like decision support system, Executive information system, Data mining and easy working systems to enable better decisions [2].

2.2.5. Definition of Open Source

the open source definition of the Open Source Initiative (OSI) is used. According to OSI this means that software must comply to the following conditions (shorted):

1. Free redistribution, including selling or using as component without fee.
2. The Source code must be available in readable form.
3. Derived work must be allowed under the same license conditions.
4. Integrity of the author's source code (licenses may require that modifications are redistributed only as patches).

5. No discrimination against persons or groups.
6. No discrimination against fields of endeavor.
7. Distribution of license (license applies to all whom the program is redistributed to, closing up software is forbidden).
8. License must not be specific to a product.
9. License must not restrict other software.
10. Licence must be technology neutral.

Licenses that conform to the above definition can get certified by OSI and may use its certification mark.

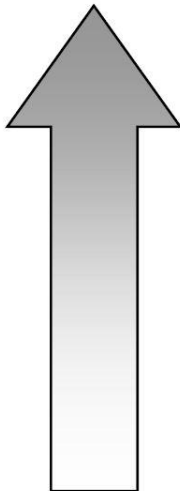
The availability of the source code reduces investment risk as the development cannot be abandoned easily. Furthermore, you have the possibility to adapt the software to your needs [3].

2.2.6. Evolution Of Enterprise Resource Planning

The evolution of ERP systems closely followed the spectacular developments in the field of computer hardware and software systems. During 1960s most organizations designed, developed and implemented centralized computing systems, mostly automating their inventory control systems using inventory control packages (IC). These were legacy systems based on programming languages such as COBOL, ALGOL and FORTRAN. Material requirements planning (MRP) systems were developed in the 1970s which involved mainly planning the product or parts requirements according to the master production schedule. Following this route new software systems called manufacturing resources planning (MRP II) were introduced in the 1980s with an emphasis on optimizing manufacturing processes by synchronizing the materials with production requirements. MRP II included areas such as shop floor and distribution management, project management, finance, human resource and engineering. ERP systems first appeared in the

late 1980s and the beginning of the 1990s with the power of enterprise-wide inter-functional coordination and integration. Based on the technological foundations of MRP and MRP II, ERP systems integrate business processes including manufacturing, distribution, accounting, financial, human resource management, project management, inventory management, service and maintenance, and transportation, providing accessibility, visibility and consistency across the enterprise.

During the 1990s ERP vendors added more modules and functions as “add-ons” to the core modules giving birth to the “extended ERPs.” These ERP extensions include advanced planning and scheduling (APS), e-business solutions such as customer relationship management (CRM) and supply chain management (SCM). Figure 2.3 summarizes the historical events related with ERP [4].



2000s	Extended ERP
1990s	Enterprise Resource Planning (ERP)
1980s	Manufacturing Resources Planning (MRP II)
1970s	Material Requirements Planning (MRP)
1960s	Inventory Control Packages

Figure 2. 3 summarizes the historical events related with enterprise resource planning (ERP) [4].

2.2.7. Open Source Software (ERP) Systems

An ERP is an integrated software package composed of a set of standard functional models including but not limited to: production, sales, human resources, and finance. OSS benefits are greater for ERP systems than for any other kind of applications due to increased adaptability, decreased reliance on a single supplier, and reduced costs. Full

access to the source code is of benefit when implementing an OSS ERP when the ERP system needs to be adapted to business processes and local regulations, and it reduces reliance on proprietary product builders and distributors. License and implementation costs for ERPs can be exorbitant. The cost of ERP implementations is estimated between one and six percent of a firm's revenue and this has always been a major barrier to SME companies. OSS is thought to improve cost effectiveness, however that costs have a secondary role in the adoption decision making of open source ERPs in spite of the high level of attention the cost perspective receives^[4].

Despite the stated advantages of OSS, an apparent increase in the adoption rate of OSS ERP systems, and the high costs of proprietary ERP systems, the adoption of OSS applications has been relatively limited in general, with an even slower uptake in South Africa. Contrary to OSS ERP uptake, investment in proprietary ERP systems has continued strongly on the back of proven operational improvement and streamlined data and process flows^[4].

OSS ERP applications are different from other OSS applications. This leads to the question of whether the barriers to adopting Open Source Software Enterprise Resource Planning Systems are different from those that apply to OSS software in general. to a large extent, provided counter arguments for the OSS ERP benefits suggested by Serrano and Sarriegi in 2006^[5].

2.2.8. Enterprise Integration

As the demand for managing information has increased, researchers have focused their efforts on integrating business processes and data. The term enterprise integration (or system integration) reflects the capability to integrate a variety of different system functionalities.

2.2.9. Enterprise Application Integration

In the mid-1990s, a new approach to system integration known as Enterprise Application Integration—EAI—was introduced. The basic concept of EAI is mainly in its

externality of enterprise integration with lower costs and less programming using existing applications. EAI is a business computing term for plans, methods, and tools aimed at modernizing, consolidating, and coordinating the overall computer functionality in an enterprise. Typically, an enterprise has existing legacy applications and databases, and wants to continue to use them while adding or migrating to a new set of applications that exploit the Internet, e-commerce, extranet, and other new technologies. EAI may involve developing a totally new outlook of an enterprise’s business and its applications, determining how existing applications fit into the new view, and then devising ways to efficiently reuse what already exists while adding new applications and data. Previously, integration of different systems required rewriting codes on source and target systems, which in turn, consumed much time and money. Unlike traditional integration, EAI uses special middleware that serves as a bridge between different applications for system integration. All applications can freely communicate with each other through a common interface layer rather than through point-to-point integration. Thus, EAI eliminates extensive programming. Figure 2.4 compares the EAI approach with traditional integration [6].

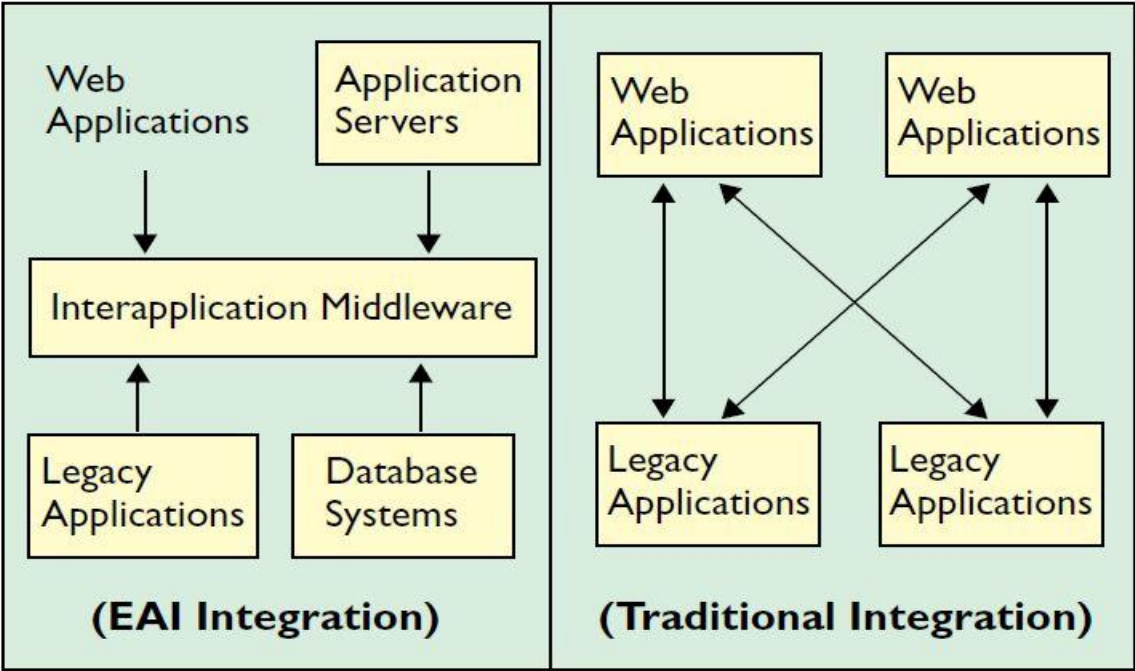


Figure 2. 4 EAI Integration vs traditional Integration [6].

2.2.10. Integration Standard

Integration standard defines common languages, file formats and communication protocols that are essential for coexistence and interoperability of the different systems, enabling the connection of systems with different vendors and origins. the need of integration standards has increased due to the following aspects:

- Growing complexity of the integration technologies
- Decrease costs for IS integration
- The need to integrate either external or internal processes
- The need to integrate all applications in an organization
- Expose services outside an organization

In large organizations, many legacy systems, including proprietary systems exist. These systems complicate the integration process due to several reasons; the communication between these systems may be inadequate, different technologies are not compatible or many systems are characterized by a very traditional governance model.

The need of integration standard has increased as the executives realize the benefits as well as the need to perceive integration as a long-term strategy for an organization. Hence, in a study conducted by Delphi group, 31% perceived standards as very relevant since it can increase the value of current and future investment on IT ^[7].

2.2.11. Service Oriented Architecture (SOA)

The basic idea of SOA is addressing systems as services and abstracts those services into single domains where they are formed into solutions. SOA can be described as “representing a set of design principles that enable units of functionality to be provided and consumed as services”. With a SOA, software and IT infrastructure can be designed to

allow applications and systems communicate and exchange data regardless of the operating systems or programming languages. The fundamental building block of SOA is a service. A SOA is composed by a set of services that communicates through a communication protocol. A service is a program that interacts through well-defined message-exchanges. An application's functionality is exposed through a collection of services. Through a contract that defines their interchanges and a defined policy of how they should be exchanged, the services are interconnected. The services must be built to enable stability and availability, which can be achieved by using standards-based interfaces and well-defined messages.

The concept of SOA has captured interest by the emergence of web services and service-oriented computing. Loose coupling is the fundamental principle and the architectural concept associated with SOA, whereas the benefits with SOA can be described with the word agility. The idea of loosely coupled systems can result in loosely coupled business processes, as the business processes are not constrained by the underlying infrastructure. To achieve this, the interfaces associated with the services must remain stable and be able to re-configure to meet the ever-changing needs of the business.

The aim of SOA services is to design re-usability and interoperability between systems and platforms. In traditional systems that do not rely on services, the function

that is provided by services need to be built into the systems. The issue with this is that the functions have to be coded or recoded every time it is needed. Consequently, different organizational units might need to build functionality from scratch, e.g. point-to-point integration need to be built, making the architecture fragmented since new integration is built for each case. However, a SOA could be a possible solution to overcome this kind of issue. A SOA can provide the mechanism whereby a function is written as a stand-alone service that is called when needed. Standardized protocols and formats are used to enable the interfaces to be loosely coupled to the system providing the service, meaning that the rest of the IT landscape is not affected when a change in a particular system is made.

Furthermore, SOA is an architectural philosophy that promises to help IT respond to market conditions in time and enable agile business processes through open, standards-based interoperability. However, describing SOA can be a challenge for organizations due to different business needs and strategies that affect the requirements and expectations of SOA. To ensure that SOA will provide a value, alignment of SOA with an organization's business drivers are required. The organizational benefits are assumed that the organization has implemented a well-designed architecture. This is for many organizations difficult due to lack of experience and knowledge of how to do it or how to manage and control the processes. Some studies suggest that the first step should be to identify the critical business problems and challenges.

2.2.12. Application Programming Interface (API)

The term "API" can have different meanings depending on the context. There are APIs for applications, operating systems and the Web. Nowadays when API is mentioned, it is often referred to web APIs. In this report, it is referred to web APIs when talking about API, the definition of an API is as follows:

"An API defines the contract of a software component in terms of the protocol, data format, and the endpoint for two computer applications to communicate with each other over a network."

Through APIs, services and assets are exposed and can easily be consumed by another application over the Web. When cloud computing emerged and cloud-based solutions and software were introduced to organizations and industries, APIs moved from being only used for social interaction to running real business and the need for APIs has ever since grown in a rapid pace. When applications want to use an API, the applications are dependent on a contract that defines the API. The contract describes how two applications communicate with each other.

2.2.13. API-management

API-management can be defined as:

“API-management is the process of publishing, documenting and overseeing APIs in a secure, scalable environment.”

For an organization that publishes an API, API management aims to monitor the interfaces’ lifecycle and ensure that the needs of the ones using the API are met. An

API management platform provides with capabilities for an organization to successfully receive business insights, analyze, manage and protect APIs in a scalable environment.

2.2.14. APIs vs. SOA

There is a huge confusion regarding the difference between APIs and SOA. APIs can be considered as an evolution of SOA, advocating for a lot of the same concepts and principles of creating and exposing re-usable services. Figure 2.5 briefly presents the main differences between SOA and APIs. SOA has an extensive and well-defined description language and focuses on control while APIs focuses on easy consumption.

Compared to SOA, APIs are more open, developer centric, easier to consume and also supports human-readable formats, e.g. JSON. From a technical perspective, the major difference between SOA and APIs is the objective behind them. While SOA focuses on reusability, agility and helps in the pace of the delivery of a service, APIs helps in the pace of innovation of building applications rapid development, time-to-market and support for cloud-based solutions and services. A service, which is the fundamental building block of a SOA can be defined as the capability while APIs define how that capability can be repackaged, productized and shared in an easy consumable format. In that sense, APIs and services are complementary rather than contradictory. When applied together, the overall effectiveness of enterprise innovation can be dramatically increased.

Seen from governance perspective, SOA focuses on re-usability of enterprise services that enable integration within the enterprise. Access to the services is controlled

and only available for trusted and well-known partners, whereas APIs are open services for developers and can be accessed on the Web [7].

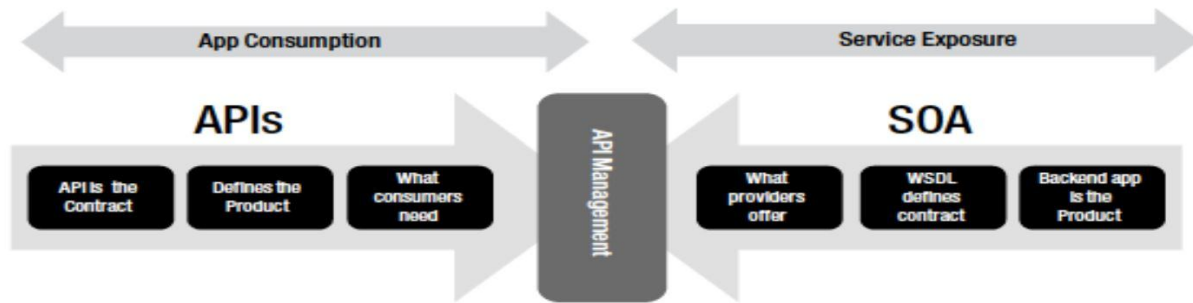


Figure 2. 5 API vs. SOA [7].

2.2.15. Web services

“A web service, in very broad terms, is a method of communication between two applications or electronic devices over the World Wide Web (WWW). “Word Wide Web Consortium defines web services as “a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine process able format. Other systems interact with the web service in a manner prescribed by its description using messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards.”

An API (web API) can be regarded as a subset of web services. It is a special kind of web service that uses different architectural patterns to communicate, compared to other web services.

2.3.Related work

2.3.1. Integration of Distributed Enterprise Applications A Survey (Wu He, Li Da Xu 2014)

This paper examined the architectures and technologies for integrating distributed enterprise applications, illustrated their strengths and weaknesses.

The paper Surveyed the current status of the integration of distributed enterprise applications and discussed the research directions. The publishers said as enterprises continue to add and/or deploy more applications, integrating distributed enterprise applications becomes inevitable for enterprises that need to achieve competitive advantages. The study said as industry enterprises are increasingly adopting multitier client/server, Internet and service-oriented architectures for their enterprise applications and industrial devices. So we have to address the need and requirement for interoperability in the industrial enterprise environment.

2.3.2. Using SOA with Web Services for effective data integration of

Enterprise Pharmaceutical Information Systems (Kester, Ajibade 2013)

study illustrates that the Medicines exist for combating illnesses and drug stores provide an outlet for consumers to gain access to these drugs. However, due to various factors such as cost of production, accessibility and legal issues as well as distribution factors, availability of these drugs at any location cannot be guaranteed at all times. There is need for individuals and organizations such as hospitals to be able to locate required drugs within given geographical vicinity. This is of immense importance especially during emergencies, travel, and in cases where the drugs are uncommon. This research work is aimed at solving this problem by designing a system that integrates all drugstores. The integration of the drug stores will be based on SOA concepts with web services via a central service bus. The database systems of the drug stores will be integrated via a service bus such that drugs can easily be searched for and the results will be displayed based on its availability. Drugs can easily be searched for within geographically distributed pharmaceutical databases as well as consumption of drugs with relation to geographical locations can easily be monitored and tracked. This will make it easy for health institutions to research on drug consumption patterns across geographical areas and also control their usage.

Research concludes that with the proposed system, individuals as well as health institutions can now search for drugs from databases that have been web enabled and

have their services registered within the service repository which is discoverable via the service bus. Using SOA with web services makes it easy for heterogeneous database platforms to be integrated and interoperate. Services created can be reused in multiple and also new services and applications can be created quickly and easily used with a combination of new and old services [9].

2.3.3. Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics (Tihomirovs, Juris 2016)

Researcher said that the most common implementations are based on SOAP (Simple Object Access Protocol) and REST (Representational State Transfer Protocol) styles. Maintainability of REST and SOAP Web services has become an important issue as popularity of Web services is increasing. Choice of the right approach is not an easy decision since it is influenced by development requirements and maintenance considerations. The paper presents the comparison of SOAP and REST based Web services using software evaluation metrics. The paper contains a systematic literature review, where a conceptual comparison method is applied to compare the SOAP and REST approaches. Results do not clearly show which of the approaches should be selected to make integration among the systems. To select the right approach, functional and nonfunctional requirements should be analyzed before making the choice about SOAP or REST. For example, if the project is about to integrate two simple information systems, the right choice will be REST. However, if the systems are complex and they should have additional security levels, the right choice will be SOAP, which supports many different standards (for example WS-Security).

2.3.4. Integration of an ERP System a case study on integration challenges with Microsoft Dynamics AX (Xu, Annie 2017)

Researcher said that system integration of ERP and other existing IT systems within an organization has always been a complex challenge. To understand the challenges, a case study at Company X has been performed. The ERP system

investigated in the case study is Microsoft Dynamics AX.

The purpose with this study is to investigate challenges that exist in large organizations during system integration of ERP system and other IT systems that exist in the organization and thereby understand how integration can be developed in a more efficient way. The study entails an extensive literature study and two rounds of data collection through qualitative interviews.

The results showed that the challenges that exist at Company X are mainly organizational. Technical challenges coexist but they have a much less significant impact than the organizational challenges. This can be explained by that there has been a heavier focus on solving the technical related problems and maybe to some extent neglected the impact organizational problems can have on system integration. Possible actions include central strategy concerning how to work with Dynamics AX, it's growth and future initiatives, policies and framework regarding working methods, increased authority and decision rights to the group working with AX as well as minimize customizations and own development.

2.4. Conclusion

Using Service oriented architecture to integrate systems most focus on the core features which are depend on requirements. All of above study's the most requirement where Scalability and performance.

2.5. Summary

This chapter addresses the definition of open source software concepts as well as the definition of ERP systems. It also discusses how to connect different systems with each other by collecting previous studies related to this matter.

Chapter Three

Methodology

Chapter 3: Methodology

3.1.Introduction

In this chapter, built a Proposed Model for integrated (Odoo ERP) with java-based application by selection the integration architecture and choose an integration method. These methods are related to SOA concept.

3.2.Methodology

There're Several studies discussed the integration of ERP systems with legacy systems exist. But a little research talk about ERP integration with a legacy system. Therefore, the research idea concerned about proposal model for the integrated ERP systems with other systems already exist.

to build the proposal model for the integrated ERP systems with other systems already exist we need the following steps. figure (3.1) show the steps that we need.

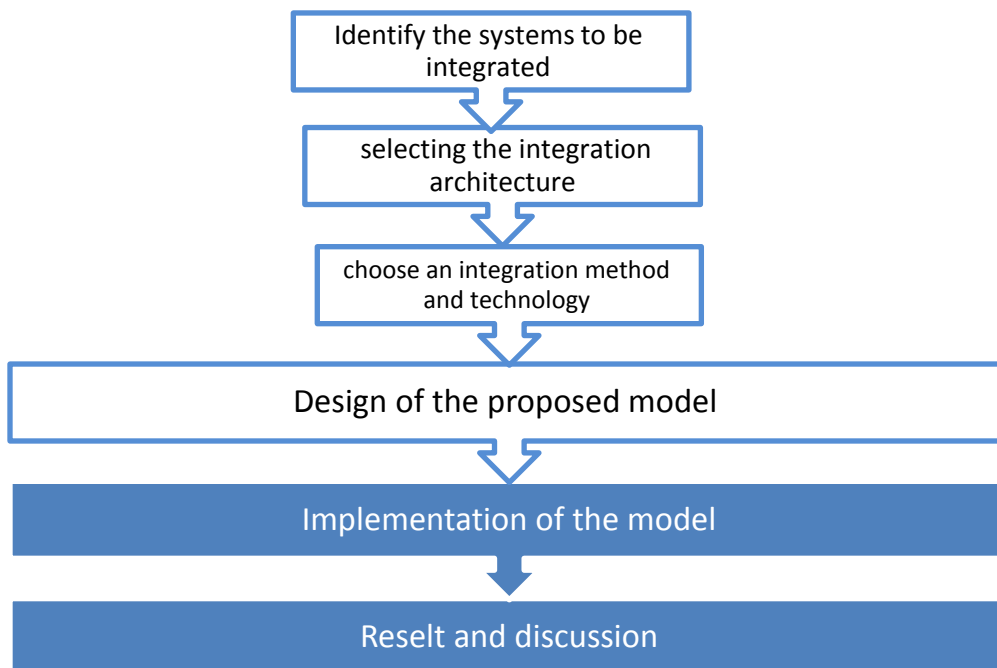


Figure 3. 1 show the steps that adopted for integration need

3.2.1. Identify the Systems to Be Integrated

System integration is the process of joining different subsystems into one large system and ensuring that each integrated subsystem functions as required. Commonly, new functionality must be integrated with existing applications, other packages and data sources without replacing legacy systems due to lack of time. The purpose of system integration is to build applications that are adaptable to business and technology changes while retaining legacy technology and systems. To achieve rapid response and protect existing investments, dealing with changes on the organizational level.

So, from this concept selected java-based application which contain information about people to integrated with CUSTMER module of (Odo ERP). We want to integrate these two systems to build one system that can increase the connectivity between the two sides. We need that the module in Odo ERP to post and get the data from java application (e.g. information ID, first name and last name) for the particular people in civilian application.

3.2.2. Selecting the Integration Architecture

There are several approaches or architecture for the enterprise application integration these approaches have a different concepts and technologies example of these concepts:

3.2.2.1. EAI

short for “enterprise application integration,” is a framework that makes use of software and computer systems to enable data integration across applications within a single organization, while also simplifying business processes among connected applications and data sources.

There are three main types of EAI:

1. The Presentation Integration Model
2. The Data Integration Model
3. The Functional Integration Model

The sharing of data and business processes between applications are its primary purposes. However, EAI also defines a set of principles for integration of multiple systems for communication architectures, such as message-oriented middleware (MOM).

EAI serves as a system that can provide a business service to simplify information data between diverse applications, which makes it possible to easily integrate them when needed. This discipline of integrating applications and data within the enterprise has been a critical component of today's enterprise strategies. In fact, many vendors offer EAI suites that provide cross-platform, cross-language integration solutions.

In EAI There are two basic architectures to achieve the enterprise application integration, bus and hub/spoke architecture. Both of these can be used to develop services and then it also becomes service orientated architecture ^[12].

3.2.2.2. SOA

Service oriented architecture is an approach to have software resources in an enterprise available and discoverable on network as well-defined services. Each service would achieve a predefined business objective and perform discrete units of work. The services are independent and do not depend on the context or state of the other services. They work within distributed systems architecture. Earlier SOA used COM or ORB based on CORBA specifications and recent SOA stress on web services using standard description (WSDL), discovery (UDDI) and messaging (SOAP) ^[12].

3.2.2.3. ESB

which is the abbreviation for “enterprise service bus,” is a software architecture that provides integration of enterprise applications and services for complex architectures, such as middleware infrastructure platforms.

An ESB's primary function is to provide the connections between communicating applications - acting much like a router to control the data. It is commonly used in enterprise application integration (EAI) or service-oriented architecture (SOA) principles. The interaction and communication between components are across the bus, which has a

similar function as a physical computer bus to handle data transfer or message exchange between services without writing any actual code.

ESB as an infrastructure software service-oriented model works as a managed message system that provides routing, data transformation, translation upon a client's request and event-interpretation. It is often needed to transform messages into a format that the application can interpret. ESB is also used to change data content or execute services via a rule engine ^[13].

At this study we chosen service-oriented architecture (SOA) to integration Odoo ERP with java base application and we chose methods related to SOA concept. Service oriented architecture may or may not use web services but yes web services provide a simple way towards service-oriented architecture.

3.2.3. Choose an integration method and technology

The task of choosing the right method of integration is predominantly an exercise in constraint-based modelling. It will vary depending on the business, and its current technological situation. The general guideline is to analyses each system and identify all potential interfaces into that application. Should the application not have any API; the backend data store represents the only option. In other cases, APIs and a CORBA infrastructure may exist so application-level integration can be employed. Having chosen the integration method, the next step is to “identify a common integration XML and JSON Schema in order to encompass all integration objects and their associated attributes” ^[14].

Study used the web service. A web service is a software component stored on one computer that can be accessed by an application (or other software component) on another computer over a network. Web services communicate using such technologies as XML, JSON and HTTP. The machine on which a web service resides is referred to as a web service host. The client application sends a request over a network to the web service host, which processes the request and returns a response over the network to the application.

This kind of distributed computing benefits systems in various ways. For example, an application without direct access to data on another system might be able to retrieve the data via a web service. Similarly, an application lacking the processing power to perform specific computations could use a web service to take advantage of another system’s superior resources. In Java, a web service is implemented as a class that resides on a server—it’s not part of the client application. Making a web service available to receive client requests is known as publishing a web service; using a web service from a client application is known as consuming a web service.

3.2.4. Design of the proposed model

The Odoo server also provides an external API, which is used by its web client and is also available for other client applications. Any programming language can be used, as long as it has support for XML-RPC or JSON-RPC protocols. As an example, the official documentation provides code samples for four popular programming languages: Python, PHP, Ruby, and Java.

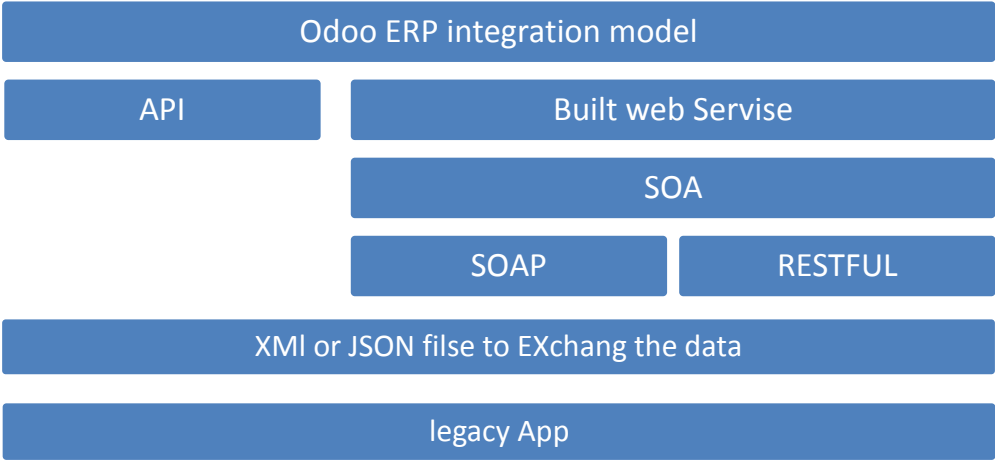


Figure 3. 2 Proposal Model of integration Odoo ERP with legacy Systems

3.2.4.1. Apply the Model on target Systems

The Ministry of the Interior Customs Authority has a great work in facilitating trade. It is necessary to provide the basic information for its customers to give them a customs clearance license. This basic information is mainly found in the Ministry of the Interior (Civil Registry Administration). Each of these parties has their own systems and the customer's information must be invoked from the database of the Civil Registry Program to confirm their authenticity and record them in the customer's information in the Customs Authority's program to give the customs clearance license.

So, we selected the Customs system as first system. On other side, we selected Civil system. We want to integrate these two systems to build one system that can increase the connectivity between the two sides and to achieve the Ministry's objective to exchange information between its bodies and public administrations and thus provide information to decision-makers.

We built web services to integrate the selected system at specific points. We built functions one of them especially for customs to get all information about the customer from civil database. The web service built based on java programming language, the customs system developed using Payson language and the civil system developed by java programming language. NetBeans IDE is a tool that we used to build the web services. Chapter four is going to discuss in more details Programming Languages, Technologies and Tools that will be used.

3.3.Summary

This chapter discussed the types of integration techniques and then chose one of them to construct the proposed models for the integration process as well as to address the design of the models.

Chapter Four

Model implementation & Result

Chapter 4: Model implimintation

4.1.Introduction

In this chapter will explain the tools, programming languages and techniques we used to implement the proposed model and get the result.

4.2.Tools

4.2.1. NetBeans IDE

NetBeans IDE lets you quickly and easily develop Java desktop, mobile, and web applications, as well as HTML5 applications with HTML, JavaScript, and CSS. The IDE also provides a great set of tools for PHP and C/C++ developers. It is free and open source and has a large community of users and developers around the world.

The important benefits of NetBeans:

Best Support for Latest Java Technologies.

Fast & Smart Code Editing.

Easy & Efficient Project Management.

Rapid User Interface Development.

Write Bug Free Code.

Support for Multiple Languages.

Cross Platform Support.

Rich Set of Community Provided Plugins ^[16].

4.2.2. Notepad++

Notepad++ is a free (as in "free speech" and also as in "free beer") source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GPL License. Notepad++ allows you to

author code in a variety of languages and supports various plug-in utilities. In addition, Notepad++ provides a few other niceties.

Based on the powerful editing component Scintilla, Notepad++ is written in C++, php and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment. ^[17]

4.2.3. WampServer

WampServer This handy piece of software gives you Apache, MySQL, and PHP all in one handy, easy-to-install package, which is used to describe any Windows-based Web server setup that uses these three open-source technologies.

WampServer is a web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases. WampServer functionalities are very complete and easy to use so we won't explain here how to use them.

- Manage your Apache and MySQL services.
- Switch online/offline (give access to everyone or only localhost).
- Install and switch Apache, MySQL and PHP releases.
- Manage your servers' settings.
- Access your logs.
- Access your settings files.
- Create alias. ^[18]

4.2.4. Odoo ERP

Odoo is a suite of open source business apps that help in managing businesses and organizations. Formerly it was called Open ERP and was an enterprise resource planning (ERP) software that provided an accomplished, integrated ERP solution. Odoo aims at

conquering new territories and widening its functionality. Grow sales, run operations, organize marketing activities, boost productivity and empower human resources - all with Odoo. As well as Open ERP, Odoo is written in Python. It is easy and lightweight software that offers a fully integrated set of apps and takes care of everything from front to back end.

Odoo is a leading business application software editor. It includes 30 main applications supported by the editor and more than 3000 applications for various business areas. Odoo also provides an open source CMS, a fully-functional eCommerce and a Business Intelligence engine. It gathered all business needs from sales and accounting to manufacturing and recruitment in one place. No other software product has such a level of integration out-of-the-box. ^[19]

4.3. Programming Language and Technology

4.3.1. Web Services

To build the web service we have Select the programming language that will be used to build web service, Understand the two systems and identify the points of integration, Select the IDE based on language, Build the web service as server side using restful techniques and Use the web service interface that built in two systems to transfer the data.

4.3.1.1. Web Service Description Language (WSDL)

To consume a web service, a client must determine its functionality and how to use it. For this purpose, web services normally contain a service description. This is an XML document that conforms to the Web Service Description Language (WSDL)—an XML vocabulary that defines the methods a web service makes available and how clients interact with them. The WSDL document also specifies lower-level information that clients might need, such as the required formats for requests and responses.

WSDL documents help applications determine how to interact with the web services described in the documents. You do not need to understand WSDL to take advantage of it—the Glassfish application server generates a web service’s WSDL dynamically for you, and client tools can parse the WSDL to help create the client-side service endpoint interface

class that a client uses to access the web service. Since Glassfish (and most other servers) generate the WSDL dynamically, clients always receive a deployed web service's most up-to-date description. To access the Welcome SOAP web service, the client code will need the following WSDL URL. [20]

4.3.2. Java 2 Platform, Standard Edition (J2SE)

The java 2 platform, standard edition, is fast, stable and secure for developing and deploying enterprise client-side applications and mission critical applications. The J2SE applications can run on a variety of desktop computers, servers, and other computing devices. The J2SE provides the basic capabilities to develop the enterprise web-based applications.

The important benefits of J2SE:

- so it reduces the development time. J2SE is a complete development platform, so it reduces the development cost.
- Object-oriented programming and design: to introduce the basic concepts and terminology of object technology in J2SE.
- Platform independence: because of the platform independence capability, J2SE reduces the development and maintenance costs. Easier to deliver and deploy applications across all targeted platform.
- Network-aware development and runtime platform: J2SE provides the mechanism to develop cost effective distributed internet applications.
- Central administration: J2SE provides a simple version control mechanism, which can be managed centrally.
- Rich and highly functional user interface components: J2SE provide a highly functional and interactive user interface, which results in improving developer's productivity.
- Database and Web Development: Accessing Databases with JDBC covers four ways using. Java Server Faces and Web services, creating and consuming based web. [21]

4.3.3. (JSON) & (XML)

JavaScript Object Notation (JSON) is an alternative to XML for representing data. JSON is a text-based data-interchange format used to represent objects in JavaScript as collections of name/value pairs represented as Strings. It's commonly used in Ajax applications. JSON is a simple format that makes objects easy to read, create and parse and, because it's much less verbose than XML, allows programs to transmit data efficiently across the Internet. Each JSON object is represented as a list of property names and values contained in curly braces.

The XML schema language is a standard for specifying the structure of XML documents. It uses the same syntax rules as regular XML documents, so that the same processors can be used on both. To distinguish the two types of documents, we will use the term XML instance document or XML document for a regular XML document, and XML schema document for a document that specifies an XML schema. The XML schema is based on the tree data model, with elements and attributes as the main structuring concepts. However, it borrows additional concepts from database and object models, such as keys, references, and identifiers. Here we describe the features of XML schema in a step-by-step manner.

There have been several proposals for XML query languages, and two query language standards have emerged. The first is XPath (expression generally returns a sequence of items that satisfy certain pattern as specified by the expression), which provides language constructs for specifying path expressions to identify certain nodes (elements) or attributes within an XML document that match specific patterns. The second is XQuery (permits the specification of more general queries on one or more XML documents), which is a more general query language. XQuery uses XPath expressions but has additional constructs. We give an overview of each of these languages in this section.

[22]

4.4.Implementation

4.4.1. Identify the Systems to Be Integrated

As a discussed in chapter three the following figures illustrates the old model of customer data entry.

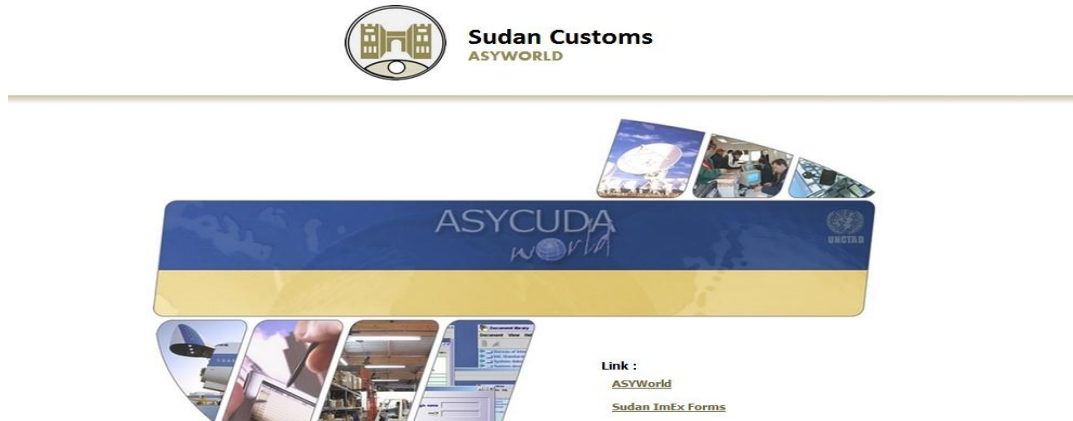


Figure 4. 1 represent the logo of legacy customs system



Figure 4. 2 represent the modules of legacy customs system

After logging and choosing the company module the following figures illustrates how the employ add the data about civil record to give the customers costume clearance license.

The screenshot shows a web browser window with the title 'Company - Create [AL SHAIKH AHMED ANWAR]'. The browser's address bar and menu bar (File, Edit, View, Help) are visible. The page features a header with the ASYCUDA logo and a banner image of shipping containers. Below the header is a 'Company Identification' form with the following fields and values:

- Company Name: AL SHAIKH AHMED ANWAR
- Company Code: 187126142142
- Address: KHARTOUM
- Telephone: 01234567
- Fax: (empty)
- Telex: (empty)
- Status: Active

Figure 4. 3 represent the adding data manual in legacy system

4.4.1.1. build services from the systems

two systems have been integrated together and services have been created as well.

The two systems are below:

4.4.1.1.1. Civil Registry system

Language used java, WampServer DBMS and notepad. Function which chosen from system to build the Services which are determine the Information about the person who wants to apply for a customs license.

All of these added as data to JSON file and XML file to integrate this using message passing interface.

```

- <peoples>
- <Pe>
  <id>1</id>
  <age>28</age>
  <name>alsheikh</name>
  <workplace>custom</workplace>
</Pe>
- <Pe>
  <id>2</id>
  <age>35</age>
  <name>abass</name>
  <workplace>alribat university</workplace>
</Pe>
- <Pe>
  <id>7</id>
  <age>30</age>
  <name>sabir</name>
  <workplace>alneleen univirsity</workplace>
</Pe>
- <Pe>
  <id>8</id>
  <age>47</age>
  <name>mahdy</name>
  <workplace>univiersty of khartoum</workplace>
</Pe>

```

Figure 4. 4 represent the information of people in json

```

{"Pe":[{"id":1,"age":28,"name":"alsheikh","workplace":"custom"},{"id":2,"age":35,"name":"abass","workplace":"alribat university"},{"id":7,"age":30,"name":"sabir","workplace":"alneleen univirsity"},{"id":8,"age":47,"name":"mahdy","workplace":"univiersty of khartoum"},{"id":9,"age":15,"name":"ayman","workplace":"omderman"},{"id":10,"age":23,"name":"morta","workplace":"sudan"},{"id":11,"age":20,"name":"morta","workplace":"sudan"},{"id":12,"age":20,"name":"morta","workplace":"sudan"}]}

```

Figure 4. 5 represent the information of people in json

4.4.1.1.2. Customer registration system

Language used python, PostgreSQL DBMS and Odoo ERP IDE. Function which chosen to build the Services are: show all people, select people to verify their information from Civil Registry system and to post this information to the customer registration systems (Odoo ERP customer module).

All of these added as functions to XML (WSDL) file which its web service has created using restful by building a module in Odoo ERP to get and post these information Integration this using a method call by URL.

Services have built after analyze the systems well and consideration that it will be applied from restful. These services represent system results.

```

- <application>
  <doc jersey:generatedBy="Jersey: 2.0 2013-05-03 14:50:15"/>
  - <grammars>
    - <include href="application.wadl/xsd0.xsd">
      <doc title="Generated" xml:lang="en"/>
    </include>
  </grammars>
  - <resources base="http://localhost:8080/civi_peple/webresources/">
    - <resource path="peoples">
      - <resource path="xml/id/{id}">
        <param name="id" style="template" type="xs:int"/>
        - <method id="retrieveOnePeople" name="GET">
          - <response>
            <ns2:representation element="people" mediaType="text/xml"/>
          </response>
        </method>
        - <method id="apply" name="OPTIONS">
          - <request>
            <representation mediaType="*/*/>
          </request>
          - <response>
            <representation mediaType="application/vnd.sun.wadl+xml"/>
          </response>
        </method>
        - <method id="apply" name="OPTIONS">
          - <request>
            <representation mediaType="*/*/>
          </request>
          - <response>
            <representation mediaType="text/plain"/>

```

p

Figure 4. 6 information about web services and wsdL.

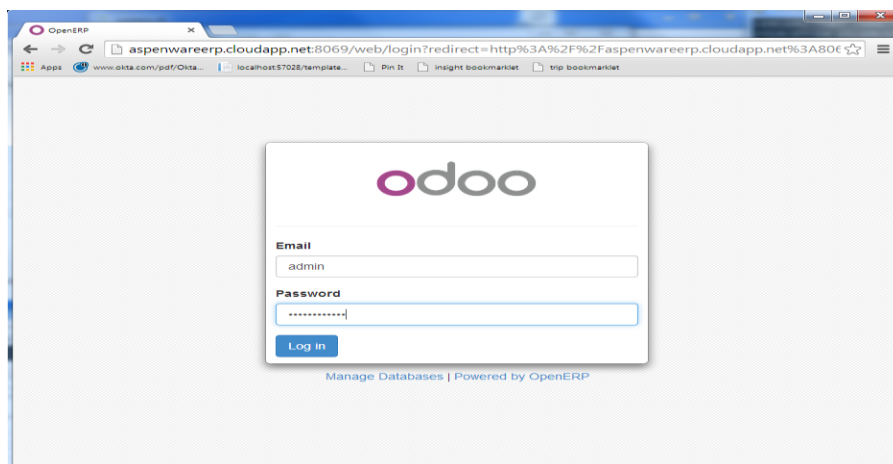


Figure 4. 7 logging of Odoo ERP

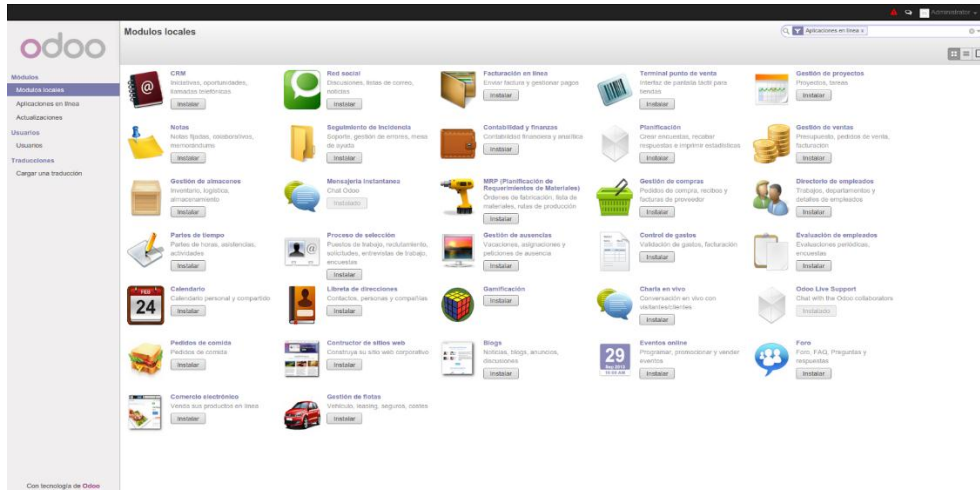


Figure 4. 8 modules of Odoo ERP



Figure 4. 9 the new modules of Odoo ERP to get information about customer

20022558502	الرقم الوطني
الشيخ أحمد أنور الطيب الشيخ	الاسم
الخرطوم , الخرطوم , الشهداء وسوبا , جيرة شمال م٣	العنوان
01/01/1987	تاريخ الميلاد
07/12/2017 13:51:59.000000	مكان العمل

Figure 4. 10 represent the data after get from web service

4.4.2. Selection The Integration Architecture

As a discussed in chapter three the integration architecture used SOA technology.

4.4.3. Integrate services with RESTFUL:

Representational State Transfer (REST) refers to an architectural style for implementing web services. Such web services are often called RESTful web services. Though REST itself is not a standard, RESTful web services are implemented using web standards. Each method in a RESTful web service is identified by a unique URL. Thus, when the server receives a request, it immediately knows what operation to perform. Such web services can be used in a program or directly from a web browser. The results of a particular operation may be cached locally by the browser when the service is invoked with a GET request. This can make subsequent requests for the same operation faster by loading the result directly from the browser's cache. Amazon's web services (aws.amazon.com) are RESTful, as are many others. RESTful web services are alternatives to those implemented with SOAP. Unlike SOAP-based web services, the request and response of REST services are not wrapped in envelopes. REST is also not limited to returning data in XML format. It can use a variety of formats, such as XML, JSON, HTML, plain text and media files.

4.5.Result

As a result of the implement of the model we obtained the integration of the two systems and the exchange of information with high performance using restful technology as the best technique for performance. After understanding and analysis of the two systems and knowledge of the aspects and requirements of the points of integration and then build the webservice will be introduced by restful technology ("If the project requires greater scalability, compatibility and performance, it is better to choose REST. REST implementation complexity, execution speed, consumed memory resources and performance were better compared to SOAP protocol. Thus, if the project requires a simple point-to point integration or large-scale availability from the mobile devices, REST is the

right choice. Based on these reasons, the major Web service providers use exactly REST: Twitter, Yahoo, Flickr, del.icio.us, etc.”). ^[10]

The final version of the system after integration considered as case study to experiment of proposal model.

4.6.Summary

This chapter discussed the nature of the programs for integration. What also discussed the tools that helped the integration and the programming languages used and the result reached by the implementation of the proposed model

Chapter Five

Conclusions and Recommendations

Chapter 5

5.1. Conclusions

The result of this research is a proposal model dedicated to connectivity the Odoo ERP with java base application systems after being integrated by the SOA technique. The model includes a set of steps to selected a right choice technology based on the concepts of integration and SOA. The Webservice was used to exchange information between two systems after being identified by the model as implement to test the model

The final system as a result of the implement of the model we obtained the integration of the two systems and the exchange of information with high performance, scalability and compatibility using restful technology as the best technique for performance and avoid modification in current programs that cost money and effort. however, the limitation of the study that not use the API exist in ERP (XML-RPC and JSON-RPC) and the restful technology not good if the integration between two systems need more of security.

5.2. Recommendation

Finally, there are some recommendations as a complement to this Study:

- Use the external API for Odoo ERP (RPC-XML& RPC-JSON) to build the web service and compare the result.
- If the project requires security and reliability, easier maintainability on the client side, as well as a lower number of possible errors, SOAP is a better choice to build the web service.
- Apply the model in different ERP and systems to see if it suits them too.

5.3. Reference

[1] Abdullah, Mr Antar Mugahed Ahmed. "Evolution of Enterprise Resource Planning." Evolution 1.11 (2017).

- [2] Moon Young B, "Enterprise Resource Planning (ERP): a review of the literature" Department of Mechanical and Aerospace Engineering Institute for Manufacturing Enterprises Syracuse University, Syracuse, NY 13244.
- [3] Herzog, Thomas. A comparison of open source ERP system. na, 2006.
- [4] Rashid, Mohammad A., Liaquat Hossain, and Jon David Patrick. "The evolution of ERP systems: A historical perspective." (2002): 2-6.
- [5] Tome Mr, Leo, et al. "Barriers to open source ERP adoption in South Africa." The African Journal of Information Systems 6.2 (2014): 1.
- [6] Lee, Jinyoul, Keng Siau, and Soongoo Hong. "Enterprise Integration with ERP and EAI." Communications of the ACM 46.2 (2003): 54-60.
- [7] Xu, Annie. "Integration of an ERP System:: a case study on integration challenges with Microsoft Dynamics AX." (2017)
- [8] He, Wu, and Li Da Xu. "Integration of distributed enterprise applications: A survey." IEEE Transactions on Industrial Informatics 10.1 (2014): 35-42.
- [9] Kester, Quist-Aphetsi, and Ajibade Ibrahim Kayode. "Using SOA with Web Services for effective data integration of Enterprise Pharmaceutical Information Systems." arXiv preprint arXiv:1307.8179 (2013).
- [10] Tihomirovs, Juris, and Jānis Grabis. "Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics." Information Technology and Management Science 19.1 (2016): 92-97.
- [11] Xu, Annie. "Integration of an ERP System:: a case study on integration challenges with Microsoft Dynamics AX." (2017).
- [12] Goel, Anurag. "Enterprise integration EAI vs. SOA vs. ESB." Infosys Technologies White Paper 87 (2006).
- [13] Key Differences between ESB, EAI and SOA,
<https://www.innovativearchitects.com/KnowledgeCenter/business-connectivity/ESB-EAI-SOA.aspx> . 19/11/2017.5:54pm.
- [14] Enterprise application integration using J2EE,

<https://www.javaworld.com/article/2074488/enterprise-java/enterprise-application-integration-using-j2ee.html> . 19/11/2017.7:47pm.

[15] Odoos Consulting Services

<https://www.safaribooksonline.com/library/view/odoo-10-development/9781785884887/ch12.html> . 17/9/2018.2:51pm.

[16] netbeans.org, tool in java, [Online]. Available:

<https://netbeans.org/features/index.html>[Accessed:18/9/2018 8:50 am

[17] Andrew Troelsen, Pro C# 2010 and the .NET 4 Platform Fifth Edition: Publisher by Apress 2010.

[18] Matt Doyle, Beginning PHP 5.3: Publisher by John Wiley & Sons, 2010.

[19] Exceeding OpenERP: Odoos with Python, :8/9/2018 9:50 am

<https://quintagroup.com/cms/python/odoo>

[20]J. Hurwitz, R. Bloor, C. Baroudi and M. Kaufman, Service oriented architecture for dummies. Hoboken, NJ: John Wiley & Sons,2007.

[21] Marcia J. Horton, Michael Hirsch and other, Java How to Program, Ninth Edition: Publisher by Pearson Education, 2012.

[22] Ramez Elmasri and Shamkant B. Navathe FUNDAMENTALS OF Database Systems SIXTH EDITION: Publisher by Pearson Education, 2011.

[23] ERP system integration poses ongoing challenges,

<http://searchmanufacturingerp.techtarget.com/essentialguide/ERP-consolidation-and-integration>(9:30 AM 12/1/2018).

