EVALUATION OF INTEGRATION APPROACH

by

Nattanicha Rittammanart

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

Examination Committee: Dr. Mathew Dailey (Chairperson)
Dr. Vatcharaporn Eaichaisakul
Dr. Donlaya ..
Dr. Prachya Chalermwat (External committee)

Nationality: THAI
Previous Degree: Bachelor of Computer Science
Walailak University , Thailand
Scholarship Donor: Haadthip Public Company Limited

Asian Institute of Technology
School of Engineering and Technology
Thailand
July 2007
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Title Page</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Table of Contents</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>List of Figures</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>List of Tables</td>
<td>iv</td>
</tr>
<tr>
<td>1</td>
<td><strong>Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 <strong>Overview</strong></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 <strong>Statement of the Problem</strong></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.3 <strong>Objectives</strong></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.4 <strong>Limitations and Scope</strong></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.5 <strong>Thesis Outline</strong></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td><strong>Literature and Technology review</strong></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1 <strong>Enterprise Resource Planning</strong></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1.1 <strong>Overview of ERP</strong></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1.2 <strong>ERP Advantages and Disadvantages</strong></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2 <strong>Method to implement an ERP system</strong></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.2.1 <strong>Development ERP software individually</strong></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.2.2 <strong>Integrated Best of breed choices</strong></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.3 <strong>Open Source Software</strong></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.3.1 <strong>Definition of Open Source software</strong></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.3.2 <strong>Comparison open source ERP</strong></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.3.3 <strong>Open Source Software’s License</strong></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2.3.4 <strong>OFBiz</strong></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2.4 <strong>Java EE Technology</strong></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2.4.1 <strong>JMS API</strong></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2.4.2 <strong>JNDI</strong></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>2.4.3 <strong>Message-Driven Bean</strong></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>2.5 <strong>Integration Technology</strong></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>2.5.1 <strong>Approach to Integration</strong></td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td><strong>Case Study: Asset Management at HTPCL</strong></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>3.1 <strong>Recent Business Architecture</strong></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>3.1.1 <strong>Asset Management Related Systems</strong></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>3.1.2 <strong>Current Asset Management</strong></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3.2 <strong>Asset Management System</strong></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>3.2.1 <strong>Overview of Asset Management</strong></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>3.2.2 <strong>Standard Business Model of Asset Management</strong></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>3.2.3 <strong>HTPCL Asset Management: New system</strong></td>
<td>23</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>This figure show ERP Enterprise Architecture. The source of this figure came from <a href="http://kazman.shidler.hawaii.edu/619ch04.ppt">http://kazman.shidler.hawaii.edu/619ch04.ppt</a>, 2006</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>This figure show Modules in an ERP based integration approach. The source of this figure came from Thomas Herzog, 2006</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>This table show size and functionality subcriteria. The source of this picture came from Thomas Herzog, 2006</td>
<td>6</td>
</tr>
<tr>
<td>2.4</td>
<td>This table show flexibility subcriteria. The source of this picture came from Thomas Herzog, 2006</td>
<td>7</td>
</tr>
<tr>
<td>2.5</td>
<td>This table show support, continuity and maturity subcriteria. The source of this picture came from Thomas Herzog, 2006</td>
<td>7</td>
</tr>
<tr>
<td>2.6</td>
<td>This table show other subcriteria. The source of this picture came from Thomas Herzog, 2006</td>
<td>8</td>
</tr>
<tr>
<td>2.7</td>
<td>This figure shows IBM’s WBI Message Broker. The source of this picture came from Jude Perira, 2006</td>
<td>13</td>
</tr>
<tr>
<td>2.8</td>
<td>This figure shows Bus Architecture. The source of this picture came from Jude Perira, 2006</td>
<td>13</td>
</tr>
<tr>
<td>2.9</td>
<td>This figure shows JCA Architecture. The source of this picture came from Dave Chappell, 2004</td>
<td>15</td>
</tr>
<tr>
<td>2.10</td>
<td>This figure shows Bus Architecture. The source of this picture came from Jude Perira, 2006</td>
<td>15</td>
</tr>
<tr>
<td>2.11</td>
<td>This figure shows ServiceMix Architecture. The source of this picture came from <a href="http://incubator.apache.org">http://incubator.apache.org</a>, 29 September 2007</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>This figure show PC Use-Case Diagram</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>This figure show RS Use-Case Diagram</td>
<td>19</td>
</tr>
<tr>
<td>3.3</td>
<td>This figure show BC Use-Case Diagram</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>This figure show CA Use-Case Diagram</td>
<td>20</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>This table recaps pros and cons of ERP systems. The source of this table came from David L. Olson and David Olson, 2004</td>
<td>4</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Overview

Nowadays, integration technologies can make the benefit in the way of business strategies in
the company. Actually, almost all companies need to use the information systems to make
convenience for working process in organization. Each information system was created by
requirement from user. Developer then starts working on the project to create the information
system which serves the user’s need. For each company, in the different time often have the
different developers and technologies. This main reason makes the information system for
each time be created by different methodology. So the main reasons that the information
systems are very separately together because:

1. Different skills: The different developers created the different information systems.
   Some of the information system can extend from the existing system. But the devel-
   opers have the different skills for creating information system. So, they want to create
   the information system in their familiar ways although the new information systems
   which they need to create can extend to the old one.

2. Technology changing: The technology for creating the information system can change
every time. Developers like to study the new technologies to replace the old. And then
they use those new technologies to create the project in the company. This can make
the problem about inconsistency technologies in the organization.

3. The required software for some applications: This point relates to software that did
   not developed from in-house. For example, maybe each of them uses several database
   technologies.

The main reasons above are the nature problems that can cause with almost all organi-
zations. So the solution to solve those problems is to think about how to make the consistency
among the information systems by do not change original systems.

For this time, almost all organizations use many information systems for working. Some
information systems need to link together for retrieving data. So, data consistency is very
importance. The main problem for the organization is data management. The concept of
ERP solved this problem already.

ERP, which will be chosen, should provide the service for accessing such as web service,
RMI, messaging-service, etc. Importantly, it should be an open source ERP because the cost
of the systems should low as much as it cans. The test-bed system will be developed on ERP
which will be chosen.

This research aims to develop the application on ERP and use ERP framework for
integrating with the several existing systems. Evaluate to find the best way for integration.
An open source ERP can be use for the test-bed of evaluation. The test-bed application
which will be developed is Asset Management System (AMS) for Haadthip Public Company
Limited. Then AMS will be integrated with each legacy system which has different database
technology.

1.2 Statement of the Problem

Recently, almost all business use information system to support business process. Several
information systems are used for different functions of work. Consistency of data is very
important for business. Integration method can make data consistency. The appropriate integration technology can improve the quality of the system in several ways. Moreover, many companies need to use ERP. The first thing that needs to do before install new ERP or migrates ERP with the existing systems is finding the way to integrate the systems. This research will study the several integration methods to integrate the systems and evaluate each technology by developing the test-bed application by using ERP framework and integrate that new system with other legacy systems.

1.3 Objectives

This research will illustrate how to integrate several legacy systems with new application which develop on ERP framework. Integration technologies which are appropriate with the legacy system can improve the quality of process and make low cost development process. Moreover, an ERP which will be chosen can reduce production cost. To achieve this goal, this research will be implemented on these objectives:

1. To integrate several legacy systems with ERP framework.
2. To compare and evaluate the appropriate integration technology for each legacy system.
3. To select and implement the test-bed system on the appropriate open source ERP.
4. To implement Thai language on the system.

1.4 Limitations and Scope

This research will implement the system on the open source ERP and then integrate it with several legacy systems. The open source ERP must be well-known and has the documentation. The legacy systems are now running in the process. So implementing and testing will be difficult in some manners.

This research relates with many legacy systems. The problem is those legacy systems have very less analysis/design documents. Some systems don’t include the architecture. Basically, each system has only database design and reports. So, it’s very difficult to understand the processes of each system.

1.5 Thesis Outline

The organization of this thesis is as follows.

Chapter 2: This chapter reviews the technologies background for implementing the integrated application. It describes the comparison of ERP and why do we chose that ERP, ERP architecture, comparison of integration approach, JavaEE technology, area of integration technology and recent business architecture of each legacy system.

Chapter 3: This chapter describes how to create basic application on ERP framework, How ERP will be integrated with other applications.
CHAPTER 2
LITERATURE AND TECHNOLOGY REVIEW

In this chapter, literatures related to overview of integration technology, current business process of legacy systems, ERP and other related development technologies. It introduces the comparison among integration approaches and comparison among ERP for the information to decide.

2.1 Enterprise Resource Planning

2.1.1 Overview of ERP

“Enterprise Resource Planning (ERP) software programs are at the cutting edge of information systems technology. ERP (pronounced “E-R-P”) programs help to manage company-wide business processes, using a common database and shared management reporting tools. ERP software supports the efficient operation of business processes by integrating business activities, including sales, marketing, manufacturing, accounting and staffing. Today’s business people (and tomorrow’s) should understand what an ERP program can do for a company.” 
(Joseph Brady, Ellen Monk, and Bret Wagner, 2001)

The famous ERP is SAP. The first time of SAP is the software product line. SAP has many modules which are integrated. As we know, SAP is very high cost. Therefore, the companies are still trying to find other ERP for supporting the business process. Even though buying some specific module can decrease cost, maybe it cannot work properly. The table below shows he business functions potentially supported by ERP.

![ERP in the Enterprise diagram](http://kazman.shidler.hawaii.edu/619ch04.ppt)

2.1.2 ERP Advantages and Disadvantages

- Integrated system – ERP is the integrated system which can use by many user in the same time. Data consistency can be solved. Less communication between each other because overall systems can process and flow with the discipline. ERP is hard to install and manage in the first phase. All user must conform about the business process. Cost of installation and conformity are very high. Although installation and conformation
is very high cost, benefit of integration are usually much greater than the costs of installation and conformity.

- Data integration – ERP share data among modules. All user use the same data which can be entered once. Therefore, the source which enter data should correct the entered data. If data is wrong from the source of data, any process can encounter the errors. Even though the data correction make an inconvenience, data integration and consistency can be added to the system.

- Best practice, less effort – ERP can provide the best way to do the business. Organization can choose the appropriate practice which can be the best for them such as the way which can make the less works for the members.

  “Further, as with any theory, what is considered best by one is often not considered best by all.”

  (David L. Olson and David Olson, 2004)

- Cost – The expectation to use ERP is the lower costs in the long run. Anyway, it is very high cost for maintenance, training, and system failure.

The table 2.1 recaps these pros and cons of ERP systems.

- Technology - more powerful, integrated computer systems,
  Greater flexibility.
  Lower IT cost.

- Business practices - better ways of accomplishing tasks.
  Better operational quality.
  Greater productivity

- Strategic - cost advantages gained through more efficient systems.
  Improve decision making.
  Support business growth.
  Build external linkages.

- Competitive - Keep up with competitors adopting ERP. Greater cost efficiencies.
  Better customer service.

Table 2.1: This table recaps pros and cons of ERP systems. The source of this table came from David L. Olson and David Olson, 2004
2.2 Method to implement an ERP system

Thomas Herzog (2006) discusses about methods to implement an ERP system. About this topic, this research will concentrate on two subtopics, development of ERP individually and integrated best of breed choices.

A small or medium enterprise (SME) has several possibilities to implement an ERP system, but this research will describe only development of ERP software individually and integrate Best of Breed Choices of ERP packages.

The figure shows how ERP can support IT processes. It shows Modules in an ERP-based integration approach.

**Figure 2.2:** This figure shows Modules in an ERP-based integration approach. The source of this figure came from Thomas Herzog, 2006

2.2.1 Development ERP software individually

If ERP system is not covering the specific requirement, individual module will be developed and integrated. But the development will base-on ERP framework or other favorite frameworks. A framework is a software library that makes up reusable design for a specific class of software. So it’s not necessary to develop system from scratch.

ERP is very interesting in aspect of changeable and reusable. Developers can use the framework as the template for development, provided you have the source code or good interface documentation. With this reason, developers can have opportunities to adapt the framework and share improvement easily.

2.2.2 Integrated Best of breed choices

Because you have the limit condition to support IT. Size of company and budget are the main conditions you have. The more you have, the more you can select the standard approach which is convinience and easily maintainable customization.

Integrate Best of Breed choices mean you can select the best module of different ERP systems and integrate them together. So, the recommend company which could be done this process should be big enough because the cost of integration is very high. Moreover, it can cause the points of failure in the system and introduce the complexity. The system requirement change in the future make the difficult to update the system because the system
has many integrated modules. Anyway, before you choose this way, “you have to find balance between the number of systems, the degree of customization, adaption to process, complexity, cost and maintenance.” (Thomas Herzog, 2006)

2.3 Open Source Software

2.3.1 Definition of Open Source software

2.3.2 Comparison open source ERP

This section describe about the comparison of each open source ERP. This section is importance because it provides the information for the decision to choose an appropriate ERP for the research. The comparison table shows about the evaluation criteria for each open source ERP.

The comparison adopted from Herzog’s paper, Comparison of Open Source ERP. He done the evaluate among different open source ERP. The referenced paper use the word “Opentaps” as a synonym for “OfBiz”. He evaluate the open source ERP in various aspects. For this paper, OFBiz is an appropriate one which this research will use.

Legend:

- ? yes
- x no
- n/a not available
- ? unknown
- + above average
- average
- - below average

(Average refers to the other evaluated open source ERP systems)

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>Open Source ERP System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SQL Ledger</td>
</tr>
<tr>
<td>SIZE</td>
<td></td>
</tr>
<tr>
<td>micro</td>
<td>+</td>
</tr>
<tr>
<td>small</td>
<td>+</td>
</tr>
<tr>
<td>medium</td>
<td>+</td>
</tr>
<tr>
<td>large</td>
<td>+</td>
</tr>
<tr>
<td>FUNCTIONALITY</td>
<td>45</td>
</tr>
<tr>
<td>e-Commerce</td>
<td>x</td>
</tr>
<tr>
<td>Accounting</td>
<td>√°°°°°°°°</td>
</tr>
<tr>
<td>MRP</td>
<td>√°°°°°°°°</td>
</tr>
<tr>
<td>POS</td>
<td>√°°°°°°°°</td>
</tr>
<tr>
<td>Inventory &amp; Warehouse</td>
<td>√°°°°°°°°</td>
</tr>
</tbody>
</table>

Figure 2.3: This table show size and functionality subcriteria. The source of this picture came from Thomas Herzog, 2006
## 2.3.3 Open Source Software’s License

### 2.3.3.1 Overview

From wikipedia (28 September 2007),

> "An open-source license is a copyright license for computer software that makes the source code available under terms that allow for modification and redistribution without having to pay the original author. Such licenses may have additional restrictions such as a requirement to preserve the name of the authors and the copyright statement within the code. One popular (and sometimes considered normative) set of open-source software licenses are those approved by the Open Source Initiative (OSI) based on their Open Source Definition (OSD)."

Licenses that are popular and widely used or with strong communities:

- Apache Software License 2.0
- new BSD License
- GNU General Public License (GPL)
- GNU Lesser General Public License (LGPL)
Figure 2.6: This table show other subcriteria. The source of this picture came from Thomas Herzog, 2006

- MIT License
- Mozilla Public License (MPL) 1.1
- Common Development and Distribution License
- Common Public License 1.0
- Eclipse Public License

This research give only 3 examples of open source license which are related or widely used.

- **Apache Software License 2.0:** Apache Software License is the Permissive free software licences\(^1\). The purpose of Apache Software foundation to use license are to distribute software and documentation(it has the terms of the documents list which any project can license), to accept regular contributions from individuals and corporations(all contributors complete, sign, and submit and Individual Contributor License Agreement: CLA), and to accept larger grants of existing software products(Before adopting a body of existing software of documentation to one of The Apache projects, they need to execute a formal Software Grant).

The condition to use Apache license are allowing the freedom for any purpose(distribute, modify and distributed modified). The Apache License is not require the same license for the modified versions. It’s only require to keep informing recipient that Apache licensed code has been used. Therefore, the top directory of redistributed software packages should be put 2 files as follows:

- LICENSE - a copy of the license itself.
- NOTICE - A “notice” text document listing the names of licensed libraries used, together with their developers.

All version of the Apache License incompatible with the version 2 of the GNU General Public License. The Free Software foundation says “The Apache Software License is incompatible with the GPL because it has a specific requirement that is not in the GPLv2: it has certain patent termination cases that the GPL does not require. (We don’t think those patent termination cases are inherently a bad idea, but nonetheless they are incompatible with the GNU GPL.)”

---

\(^1\)are free software licences for a copyrighted work that offer many of the same freedoms as releasing a work to the public domain. In contrast, copyleft licences like the GNU General Public License require copies and derivatives of the source code to be made available on terms no more restrictive than those of the original licence. Well-known examples of permissive licences include the MIT License and the BSD licenses. – WikiPedia
From the content of Apache License, in term of “it’s not require the same license for the modified versins”, it means Apache supports users to create their own contribution by only referring to Apache. Therefore, many organization contribute many product from Apache to be the commercial products. For this research also uses the product from Apache (Apache OFBiz) to create the new product on top of it.

- **new BSD License**: BSD License is the Permissive free software licences. It have been referred to as copycenter\(^2\). WikiPedia(28 September 2007) provide an example about the text of license which is considered to be in the public domain\(^3\) and thus may be modified without restriction. The detail looks like the agreement for using software which the holder can modify the content to relate with the software. BSD license is widely used for several free or open source licenses. For example, NetBSD and FreeBSD.

- **GNU General Public License (GPL)**: GPL License is the permissive free software licenses which is used by Linux kernel. It’s also the family of the strong copyleft license (the same to BSD licenses)

In sum, the comparison table shows the differentiation among open source license. This research adopted the comparison only 9 licenses which are popular and widely used. Table will be put here .............

### 2.3.4 OFBiz

The ERP started from SAP. Many companies are using some modules from SAP. But SAP is very high cost both establishing and maintenance. Therefore, companies need to use the low cost ERP software. OFBiz is also ERP system but it is an open source. Recently, many developers are interested in OFBiz to make several contributions. The existing systems can be integrated with OFBiz easily because OFBiz provides the service for accessing such as web service and RMI.

#### 2.3.4.1 Introduction to OFBiz

"The Apache Open For Business Project is an open source enterprise automation software project licensed under the Apache License Version 2.0. By open source enterprise automation we mean: Open Source ERP, Open Source CRM, Open Source E-Business / E-Commerce, Open Source SCM, Open Source MRP, Open Source CMMS/EAM, and so on."(www.ofbiz.org, 19 Sep 2006)

OFBiz is the enterprise solutions which is used by many organization. It can use for creating specialized applications for use OOTB (out of the box) by other organizations. Moreover, OFBiz can cause benefit about the deployment and maintenance costs which is lower price than commercial ERPs. OFBiz is an open source under the Apache 2.0 and driven

---

\(^2\)The way it was characterized politically, you had copyright, which is what the big companies use to lock everything up; you had copyleft, which is free software’s way of making sure they can’t lock it up; and then Berkeley had what we called ‘copycenter’, which is ‘take it down to the copy center and make as many copies as you want.’ – Kirk McKusick, BSDCon 1999

\(^3\)comprises the body of knowledge and innovation (especially creative works such as writing, art, music, and inventions) in relation to which no person or other legal entity can establish or maintain proprietary interests within a particular legal jurisdiction.
by a community Apache OFBiz. Thailand also has many communities of OFBiz. Therefore, you can communicate with many people who get things done.

Apache OFBiz's official site (www.ofbiz.org, 19 Sep 2006) says about the OFBiz's functions, including:

- advanced e-commerce
- catalog management
- promotion and pricing management
- order management (sales and purchase)
- customer management (part of general party management)
- warehouse management
- fulfillment (auto stock moves, batched pick, pack and ship)
- accounting (invoice, payment and billing accounts, fixed assets)
- manufacturing management
- general work effort management (events, tasks, projects, requests, etc)
- content management (for product content, web sites, general content, blogging, forums, etc)
- and much more all in an open source package

2.3.4.2 OFBiz Framework introduction

2.3.4.3 How to implement the application on OFBiz

2.3.4.4 Related Technology to OFBiz: BeanShell

BeanShell is the scripting language which can be embedded to the application. BeanShell is written in Java. Therefore it can run with the same JVM in the machine.

BeanShell can help to develop OFBiz framework by generating action or script for the screen section actions in process layer. Normally, Screen Section Actions in OFBiz can use service *.xml (in user interface layer), entitymodel*.xml (in data source layer) and BeanShell script (in process layer) to create them.

2.4 Java EE Technology

2.4.1 JMS API

Java Message Service is a Java API that allows applications to create, send receive, and read messages (Sun Microsystems). JMS API architecture consists of 4 main factors. There are JMS provider, JMS clients, Message and administered objects.

JMS provider “is the messaging system that is used by JMS client. It implements the JMS interface and provides administrative and control features.” (Sun Microsystems, 2007)

JMS clients can be producer or consumer program that are written in the Java program.
Messages “are the objects that communicate information between JMS clients.” (Sun Microsystems, 2007)

Administered objects, “the two kinds of JMS admixture objects are destinations and connection factories” (Sun Microsystems, 2007).

- Destination is the object used by a client for specifying the target and the source of the messages.
- “Connection factory is the object used for creating connection to a provider. It is an instance of the ConnectionFactory, QueueConnectionFactory, or TopicConnectionFactory interface” (Sun Microsystems, 2007).

To describe about the messaging domain of JMS, JMS API supports both point-to-point and publish/subscribe domain.

Point-to-point domain has the concept about queue, message, sender and receiver. Each message has only one consumer. PTP has no timing dependencies. It means receiver can fetch the message although the client that sent message is not running.

Subscript domain has the concept of topic. Each message can have multiple consumers. Pub/Sub has timing dependencies. It means receiver cannot fetch the message if the client that sent message is not running. But JMS API solves this problem by allowing the subscribers to create durable subscriptions which can receive message when the client that sent the message is not active.

In each way of message domain, consuming message has different way. Message can be consumed in either of 2 ways.

Synchronously: A receiver fetches the messages by calling the receiver method. During the time that a message still not arrives, the receiver method can block. But if a message does not arrive within a specified time limit, the receiver method can time out.

Asynchronously: client use message listener to notify the JMS provider, to delivers the message by calling the listener’s onMessage method, when a message arrives at the destination.

This research only talks about asynchronously. The example from Java EE 5 Tutorial will show about how to write JMS API program in the kind of asynchronously message consumption.

AsynchronousConsumer.java is the consumer’s program. It needs to define JNDI name for the related variables.

```java
@Resource(mappedName = "jms/ConnectionFactory")
private static ConnectionFactory connectionFactory;
@Resource(mappedName = "jms/Queue")
private static Queue queue;
@Resource(mappedName = "jms/Topic")
private static Topic topic;
```

It needs to create connection before receiving information:

```java
connection = connectionFactory.createConnection();
```

After create connection, you use it to create a session. The first argument means that the session in not transacted. The second means that it will be acknowledge automatically when it successfully received the message:

```java
session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
```

And then it uses the session to create consumer. It’s up to dest variable which enter from user which it will be either queue or topic.

```java
consumer = session.createConsumer(dest);
```
That’s from the mention before, because this program is about asynchronously message consumption, it will use the message listener:

```java
listener = new TextListener();
```

After that, set the listener to consumer

```java
consumer.setMessageListener(listener);
```

Finally, start the connection:

```java
connection.start();
```

AsynchronousConsumer.java is the consumer’s program. It needs to define JNDI name for the related variables. The codes are look like the consumer’s program for defining JNDI name. And then the producer also needs to create the connection and session. Moreover, because of it’s the server, it needs to create the messages to sent to consumer:

```java
TextMessage message = session.createTextMessage();
```

### 2.4.2 JNDI

(Java Naming and Directory Interface) is the API for directory service which allow client accessing by using the name. JNDI is used by RMI and Java EE to lookup objects in a network. JNDI has 3 major components include naming model, directory models, and service provider model. First, we need to define initial context to identify service provider to use. Next, we can look up the objects.

JNDI is usefule in the way for looking up references to both local and remote components/resources. For example of remote components, EJB also use JNDI’s concept in session bean. Session bean can remote look up the object from server by identifying resource to JNDI.

### 2.4.3 Message-Driven Bean

MDB is a kind of Enterprise bean. “it’s similar to session bean, except it responds to a JMS message rather than an RMI event. MDBs were introduced in the EJB 2.0 specification which is supported by Java 2 Platform, Enterprise Edition 1.3 or higher. The message bean represents the integration of JMS (Java Message Service) with EJB to create an entirely new type of bean designed to handle asynchronous JMS messages.”(Message-Driven Bean on WikiPedia, 26 September 2007)

Message-driven bean using topic with a durable subscription, and it also using a message selector.

### 2.5 Integration Technology

#### 2.5.1 Approach to Integration

Pereira (2006) organizes the approaches to enterprise application integration as follows:

#### 2.5.1.1 Broker Based (Hub-and-Spoke) Approach

Centralized server is used by application to send and receive data
2.5.1.2 Bus Based Approach

All nodes link to the communication backbone. The data is sent via the bus to adapter. This adapter use for data transformation, translation and subsequence routing to the receiving node. The bus architecture is different from the broker based approach since the data transformation and routing is distributed in each of the application adapters.

![Bus Architecture Diagram]

**Figure 2.8:** This figure shows Bus Architecture. The source of this picture came from Jude Perira, 2006

2.5.1.3 JCA Based Approach

JCA (JavaEE Connector API) is the standardize contract that allow the connection between the legacy system. The contract are the backend EIA, the different application components, the application server, and the resource adapter. The EAI vendor provided their resource adapter to integrate with other EIS. Therefore, the resource adapter for each vendor would be different from another. JCA implement base on Java Technology to connect application servers and EIS. JCA use for connecting to legacy system (including databases).
Sun Microsystems (2007) inform about the system contract defined by Version 1.0 of the J2EE Connector Architecture are described by the specifications as follows:

- **Connection management**: Connection management enables an application server to pool connections to the underlying EIS and enables application components to connect to an EIS. This leads to a scalable application environment that can support a large number of clients requiring access to an EIS.

- **Transaction management**: Transaction management enables an application server to use a transaction manager to manage transactions across multiple resource managers. This contract also supports transactions that are managed internal to an EIS resource manager without the necessity of involving an external transaction manager.

- **Security management**: Security management provides support for a secure application environment that reduces security threats to the EIS and protects valuable information resources managed by the EIS.

The additional system contracts defined by Version 1.5 of the J2EE Connector Architecture are described by the specification as follows:

- **Life cycle management**: Life cycle management enables an application server to manage the life cycle of a resource adapter. This contract provides a mechanism for the application server to bootstrap a resource adapter instance during its deployment or application server startup, and to notify the resource adapter instance during its undeployment or during an orderly shutdown of the application server.

- **Work management**: Work management enables a resource adapter to do work (monitor network endpoints, call application components, and so on) by submitting work instances to an application server for execution. The application server dispatches threads to execute submitted work instances. This allows a resource adapter to avoid creating or managing threads directly, and allows an application server to efficiently pool threads and have more control over its run time environment. The resource adapter can control the transaction context with which work instances are executed.

- **Transaction inflow management**: Transaction inflow management enables a resource adapter to propagate an imported transaction to an application server. This contract also allows a resource adapter to transmit transaction completion and crash recovery calls initiated by an EIS, and ensures that the Atomicity, Consistency, Isolation and Durability (ACID) properties of the imported transaction are preserved.

- **Message inflow management**: Message inflow management enables a resource adapter to asynchronously deliver messages to message endpoints residing in the application server independent of the specific messaging style, messaging semantics, and messaging infrastructure used to deliver messages. This contract also serves as the standard message provider pluggability contract that allows a wide range of message providers (Java Message Service (JMS), Java API for XML Messaging (JAXM), and so on) to be plugged into any J2EE compatible application server with a resource adapter.

Dave Chappell (2004) describe about JCA Architecture (from the figure). An ESB that supports a JCA container allows applications to be plugged into an ESB using a standard set of off-the-shelf adapters. These adapters can be made available from the application vendors themselves, such as SAP, or from vendors who specialize in providing adapters, such as iWay and DataDirect. The vendors who provide the JCA container:

2.5.1.4 ESB Based Approach

The definition from WikiPedia, “In computing, an enterprise service bus (ESB) refers to a software architecture construct, implemented by technologies found in a category of middleware infrastructure products usually based on standards, that provides foundational services for more complex architectures via an event-driven and standards-based messaging engine (the bus).”

Jude Pereira (2006) describe about the ESB’s definition is “a new architecture that exploits Web services, messaging middleware, intelligent routing, and transformation. ESBs act as a lightweight, ubiquitous integration backbone through which software services and application components flow.”

Apache ServiceMix is an Open Source ESB which is in the Apache Project. It combines the functionality of SOA and EDA to create an enterprise ESB. It’s built from the groud of JBI. The gold of Apache ServiceMix is to integrate the components or services from several vendor which is implemented in different ways. Users and vendors can plug and play the application.

---

4 Java.net is the organization which provide the source for Java Technology Collaboration
5 Service-Oriented Architecture
6 Event Driven Architecture
7 Java Business Integration
Figure 2.10: This figure shows Bus Architecture. The source of this picture came from Jude Perira, 2006

Figure 2.11: This figure shows ServiceMix Architecture. The source of this picture came from http://incubator.apache.org, 29 September 2007
Open ESB is an ESB which using JBI which is provide by Sun Microsystem. It allows allows webservice’s integration to create loosely couple application.

2.5.2 Middleware service

Apache ActiveMQ is the Open Source Message Broker which provide by Apache Project. It supports many cross clint languages and protocols. It’s fully supports JMS 1.1 and J2EE 1.4 with support for transient, persistent, transactional and XA messaging. Moreover, it includes JCA resource adapter for inbound and outbound message.

Apache Camel is a subset of Apache ActiveMQ which make a new approach to middleware services and mediation and routing. Apache Camel has established because the people wanted to create and use patterns from the Enterprise Integration Patterns book\textsuperscript{8} in many different scenarios.

\textsuperscript{8}Gregor Hohpe, Bobby Woolf, \textit{Enterprise Integration Patterns}, 2003
CHAPTER 3
CASE STUDY: ASSET MANAGEMENT AT HTPCL

In this chapter, case study related to current business model of legacy systems, requirement of the new asset management system and asset management’s best practices. It introduces some coverage of standard business models and the commercial ERP products implementing those business models for asset management. The chapter also include the comparison of HTPCL situation and the standard business practices.

3.1 Recent Business Architecture

Before now, HTPCL used to have Asset Controlling. A responsible employee controlled all assets in company by using Excel file. She/he did the controlling in manual. The company never had Asset Management System. Now, all assets cannot track and retrieve information about each. The executives cannot know about the asset information. So, benefit calculation cannot be done correctly.

3.1.1 Asset Management Related Systems

Asset controlling relate to these systems:

3.1.1.1 General Ledger
General Ledger (GL) is the main system that is used by Accounting Department. This system manages about journals. User can post general journals, post standing journals, post standard journals, post spread journals, reverse general journals, reverse standing journals, reverse standard journals, reverse spread journals and past period journals. The processes are actually about accounting processes.

System Architecture of this system can describe below:

- Use-Case Diagram

System Environment of this system can describe below:

- Programming Language – Visual Foxpro
- Database – DBase

3.1.1.2 Purchase Control
Purchase Control System is the system that is used by procurement department. The main task is verifying the expenditure of each department to match with budget. Moreover, it provides the utility for the information about request for ordering, ordering and receiving with the efficient process. However, Cash bills need to send the copy to accounting department for keeping information. The main purpose for Purchase Control System is to develop system which supports procurement for the related person.

System Architecture of this system can describe below:

- Use-Case Diagram

System Environment of this system can describe below:
3.1.1.3 Reserved Store

Reserved Store is the system that is used by Store Department. Received bills are keyed into the system when store department received bills from procurement department. Store department respond to distribute anything which are buy to other departments. Stock maintenance is also important process. Before now, store department respond in asset controlling when they distributed anything. But when a responsible human are resigned, some store department’s process are changed, especially asset controlling process. Before now all assets must be submitted the bill to store department, but process nowadays is not need to pass any assets’s bills to store department, each department can order and receive assets directly. So, store department don’t keep any information about assets in company. It means company cannot track any information of assets.

Recently, reserved store system still is a kind of DOS system. Computer department’ve tried to upgrade the system to visual foxpro system but it’s not finish. The plan to upgrade the system to visual foxpro now obsolete. The new plan is to develop in java.

System Architecture of this system can describe below:

- Use-Case Diagram

System Environment of this system can describe below:

- Programming Language – Foxpro on DOS
- Database – DBase

3.1.1.4 Budget Control

Budget Control is the system that is used by all departments. Any departments need to key in their budget information to the system in the defined time. Each department can manage their budget on this system. Accounting department will revise each department’s budget
in each period. Users can retrieve information about each department’s budget and overall budgets.

System Architecture of this system can describe below:

- Use-Case Diagram

**bc Use Cases**

Figure 3.3: This figure show BC Use-Case Diagram

System Environment of this system can describe below:

- Programming Language – Visual Foxpro
- Database – DBase

3.1.1.5 Marketing Asset

Marketing Asset system is the system that is used by Sale Center department. Normally, marketing asset can be controlled by sale center directly. It means ordering, buying and
keeping the assets are not necessary to be done by purchasing or store department. Sale center respond to distribute marketing asset for each branch and keep track of any marketing assets. However, Cash bills need to send the copy to accounting department for keeping information.

System Architecture of this system can describe below:
- Use-Case Diagram

System Environment of this system can describe below:
- Programming Language – Visual Foxpro
- Database – DBase

3.1.1.6 Computer Asset
Computer Asset System is the system that is used by Computer department. The system keeps information for only computer assets. Normally, ordering, buying and keeping the computer assets are done by computer department itself. Each department who needs to order computer asset must inform computer department to order. Cash bills need to send the copy to accounting department for keeping information.

System Architecture of this system can describe below:
- Use-Case Diagram

![ca Use Cases](image)

**Figure 3.4:** This figure show CA Use-Case Diagram

System Environment of this system can describe below:
- Programming Language – Visual Foxpro
- Database – DBase

3.1.2 Current Asset Management

3.1.2.1 AS-IS Process
Recently, Accounting department responds about managing asset’s information. Asset’s documents are kept in Excel files. In these files, they keep the information which group by
department. The purpose to keep the asset’s information for accounting department is for calculating the depreciation for each asset. Therefore, maybe these Excel files do not contain much information which can benefit by other departments. Asset control now has responsibility only for keeping the information while each department has responsibility to keep their own assets. Therefore, it is the responsibility for each department to inform about broken assets or expired assets.

The assets are divided in 2 groups, the assets which are buyed before year 1990 and the assets which are buyed after year 1990. The reason to divide into 2 groups because of the different depreciated calculation.

Before now, some departments tried to control all assets more than keeping information. Store department had an employee who responded about tracking assets. He/she has the excel files to keep information not only the detail about the asset, but also keeping the tag of each asset. Therefore, he/she could know the age of each asset and inform the the department to amortize some assets. But this point of business process changed when this employee resigned. Finally, store department does not keep the information about asset until now. Moreover, Accounting departement and internal audit office used to participate for asset controlling. But it did not successfull for doing with all assets. Recently, the internal audit office still keeps the asset information for the storehouse branches (does not include the head office). Anyway, all of them never defined the code for the assets.

This paragraph describes about the flow of asset control process. First, the assets are buyed by procurement department. The information about buying is kept in the PC system (Purchase Control System). And then the receipts are sent to Accounting department. For the storehouse branches which use the petty cash, although the information is not kept in the PC system, the receipt still be sent to Accouting department. Therefore, Accounting department is the central to which all receipts must be sent. All receipt informations are entered to GL system (General Ledger) by accouting system for keeping the depreciations. Please note that depreciated calculation cannot done by GL system. Accouting department need to calculate the depreciation before enter to the system in each month.

The law of depreciation mention that the ledger must be has 1 Bath left for each asset which already expired. Amortization can be done by the approval from the department who is an owner of the assets. Accounting department does not have the permission to amortize all asset without the consent from the asset owner. Therefore, the company now has the responsibility to keep many one Bath’s ledger which cannot amortize. The solution for this problem is verifying all assets in the company. This way need to discuss with the executive.

Depreciated calculations are difference among each group of asset. The group of asset can shows below:

1. Equipment: depreciation rate 20%.
2. Engine: depreciation rate 10%.
3. Construction: depreciation rate 5%.

3.1.2.2 Limitation of current asset management

- Because of keeping information in excel file, it’s hard to maintain the file.
- Difficult for complex calculation.
- Time consuming.
- Other departments cannot use information because the files are not distributed.
- Lacking of data relationship.
3.2 Asset Management System

3.2.1 Overview of Asset Management

From the definition of bestpricecomputers.co.uk (29 September 2007),

“Asset Management is the management of physical assets (their selection, maintenance, inspection and renewal), which plays a key role in determining the operational performance and profitability of industries that operate assets as part of their core business. Asset Management is the art and science of making the right decisions and optimising these processes.”

The goal of Asset Management Software is to help maintaining record of asset. It shows the information which we need such as depreciation of each asset. Asset Management Process is very important process because it can let the company know about the profits in each year.

3.2.2 Standard Business Model of Asset Management

Business Model is the process which link to related persons for reaching the gold of customer satisfaction. This section introduct the business model which use for implementation of Asset Management from several vendors.

3.2.2.1 Hackett Group’s Best practices

Ivy McLemore (1996) discuss about World-Class Fixed Asset Management. The information from the Hackett Group, an Ohio-based consulting firm, has identified the following in fixed management:

- Use a single fixed asset system with standard capitalization policies.
- Automatically post depreciation charges and asset activity to the general ledger.
- Use one policy for depreciation lives.
- Use one approval chart for capital appropriation requests.
- Link an automated capital project tracking system directly to purchasing, accounts payable, engineering and labor systems for tracking purchased and constructed assets.
- There should be an automatic setup of fixed assets based on data accumulated in the capital project tracking system.
- Routinely circulate bar-coded asset tags throughout your company.
- Implement an automatic linkage to the corporate tax system.
- The system should meet tax requirements for all tax authorities.
- Conduct physical inventories routinely on a cycle count basis using bar-code scanners.
- Specialize functional tasks to appropriately match activities with required skills; clerks perform data entry while accounting professionals perform complex capitalization/disposition calculations.

Cost of Product

23
3.2.2.2 SAP: AA module

From SAP site (27 September 2007), Asset Accounting is one component in Financial Accounting module. The SAP Asset Accounting (FI-AA) component is used for managing and supervising fixed assets with the SAP R/3 System. In SAP R/3 Financial Accounting, it serves as a subsidiary ledger to the FI General Ledger, providing detailed information on transactions involving fixed assets.

- Rules for posting depreciation
- Changing the YTD depreciation
- Transfer of Fixed Asset
- No depreciation for a period of time
- Asset with zero depreciation

HTPCL has a plan to adopt Basis On PC (bOnPC). From the presentation details, bOnPC is a new software package in the BASIS family which support the business about Bottling operations at small bottlers, depots/sales centers at large bottlers and wholesalers, Distributors. It can be operated on a single stand-alone personal computer, or in a network for several users. Moreover, It uses the same business and system concepts as BASIS II Release 53.

bOnPC use open database concepts which allowing for easy access of data by commercial tools. MySQL and JDBC are used for database technology. For the details of bOnPC architecture, the program allows easy hardware extension if the performance is not satisfactory or the number of users is too high, several application server computers can be used, and dedicated database server(s) can be used if necessary.

bOnPc functionalities includes the functions relate to master data, sales and distribution, stock management, finance, and system administrator. Asset management is one module in finance function. Finance function relates to accounting management, accounting receivable and sales accounting. But for the current plan, HTPCL did not plan to buy asset management module because of waiting for Asset Management System on the open source.

Cost of Product

3.2.2.3 ICT Business Solution (IBS): Asset Management System

Information and Communication division, Merchant Partners Securities Limited, introduce about the principle of Asset business transaction which Asset Management System should be done:

- Asset Acquisition and Sell of Asset
- Depreciation Calculation
- Dispositions of assets
- Amortization
- Transfer of Fixed Asset

Cost of Product
3.2.3 HTPCL Asset Management: New system

3.2.3.1 Requirement Description

As we know that the current asset management documents are in Excel format. Therefore, it is very difficult to manage the large amount of data which increase dramatically. The purpose of the system can summarize below:

- Use a single fixed asset system with standard capitalization policies.
- Automatically post depreciation charges and asset activity to the general ledger.
- Link an automated capital project tracking system directly to purchasing and accounts payable.
- Routinely circulate bar-coded asset tags throughout your company.
- Asset Acquisition and Sell of Asset
- Amortization
- Transfer of Fixed Asset
- To track the asset’s status about age, position and etc.
- To calculate the depreciation for each asset.
- To summarize all asset value of the organization in several aspect.
- To inform the business profits of the organization.
- To improve the business process to reach the quality and efficiency.