A SALES ORDERING SYSTEM FOR A STEEL TRADING FIRM

by

Mr. Ekapoj Jitpattanarat

A Final Report of the Six-Credit Course
CE 6998 - CE 6999 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

November 2002

151'35-1
MS (CET)
St. Gabriel's Library, Au

A SALES ORDERING SYSTEM FOR A STEEL TRADING FIRM

by
Mr. Ekapoj Jitpattanarat

A Final Report of the Six-Credit Course
CE 6998 - CE 6999 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

November 2002

Project Title	A Sales Ordering System for a Steel Trading Firm
Name	Mr. Ekapoj Jitpattanarat
Project Advisor	Dr. Chamnong Jungthirapanich
Academic Year	November 2002

The Graduate School of Assumption University has approved this final report of the six-credit course, CE 6998 — CE 6999 PROJECT, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer and Engineering Management.

Approval Committee:

04h40:-	
(Dr. Chamnong J gt rapanich)	(Prof.Dr. Srisakdi Charmonman)
Dean and Advisor	Chairman

(Assoc.Prof. Somchai Thayaniyong)
MUA Representative

November 2002

ABSTRACT

This project develops an analysis and design of the Sales Ordering System for steel trading company in order to increase the effectiveness and efficiency of the sales order processing's operations.

The system is a function of an order processing for steel products. The information system, application portfolio, and system operation are provided in order for a comprehensive understanding of an existing system's operations. In addition, system requirements are provided for the analyst in the analysis of existing system and proposed system. In system development, the proposed Sales Ordering System is designed on areas as process modeling, logic modeling, and conceptual data modeling. The development of forms, reports, interfaces, and dialogue are included in the proposed system. Since the company was periodically invested in the computer technology, the development of proposed system was a better utilization of existing resource. Also, system feasibility assessment economically suggests the financial benefits with the development system. System evaluation was accomplished to inspect the system's verifications and validations. The results of the system evaluation suggest that the system performs accurately and satisfactorily according to the design specifications and objectives.

Therefore, the development of a new computerized of Sales Ordering System can effectively and efficiently improve the sales ordering's operations of the company.

ACKNOWLEDGEMENTS

I am indebted to the following people and organizations. Without them, this project would not have been possible.

I wish to express sincere gratitude to my advisor and dean of the Advisory Committee, Dr. Chamnong Jungthirapanich. His patient assistance, guidance, and constant encouragement has led me to the project completion.

I would like to thank sales staff's team of Sahaviriya Panich Company for their kindness, helpful, and useful, and useful information for use in project development And I would like to thank my best friend Miss Orawan Limchaiyawat who always speed up and advise me in doing this project.

Special appreciation is due to my family for their fervent and continuous encouragement. Above all, I am forever grateful to my parents whose willingness to invest in my future has enabled me to achieve my educational goal.

St. Gabriel's Library, Au

TABLE OF CONTENTS

ASSUMPTION UNIVERSITY OF THAILAND
SINCE 1969

LIST OF FIGURES

ASSUMPTION UNIVERSITY OF THAILAND
SINCE 1969

LIST OF TABLES

ASSUMPTION UNIVERSITY OF THAILAND
SINCE 1969

I. INTRODUCTION

1.1 Background

Sahaviriya Panich Company, or SVP, a steel trading company, is owned and operates by the Sahaviriya Group, Inc. As of January 2002, SVP has more than 200 customers across Thailand and is a representative for 20 steel product's manufacturers. There is at least one SVP customer in every part of Thailand. There are also 10 customers, five in Lao, and five in Burma. The company is currently struggling to expand the market in Vietnam within three years. SVP office and warehouse are located in Bangkok.

SVP offers for sale two product lines, long-shape steel product and flat-shape steel product. Long-shape steel product is sold in ten formats, and also flat-shape steel product is sold in ten formats, according to the customer usage. Long-shape steel product includes round bar, light lips channel, slitter, wire mesh, wire rod, flat bar, angle, equal angle, Z-bar, and bright bar. Flat-shape steel product includes coil, plate, sheet, zinc, and scab. In calendar year 2001, both product lines accounted for 40% of the total industrial market share. There is totally three million tons of steel consumption in Thailand. The company's total sale is accounted for 1.2 billion baths in 2002.

The steel industry is strong and growing, both domestically and internationally. The company's sales volume in domestic last year exceeded 0.84 million tons (over 8.4 billion baht). In addition, the company's export last year exceeded 60,000 tons (ove72 million baht). In 2002, domestic revenue is expected to exceed 12 billion baht and export is expected to exceed 3.6 billion baht.

To get a good idea of the industry in which Sahaviriya Panich completes, we can use Porter's Five Forces Model of Industry Structure (1980) which shown in Figure

1.1. According to Porter, any industry can be analyzed by focusing on five aspects of its structure: (1) suppliers, (2) buyers, (3) substitutes, (4) barriers to entry, and (5) rivalries among firms completing in the industry. Porter's Five Forces Model of Industry Structure is depicted in Figure 1.1. For suppliers, the company deals with the manufacturers, which is within the Sahaviriya Groups. Customers (buyers) consist entirely of wholesalers or redistributors. The company does not provide product for individual consumer or retailers and does not deal with the groups or associations. Substitutes for the company offerings include imported finished products and domestically produced products.

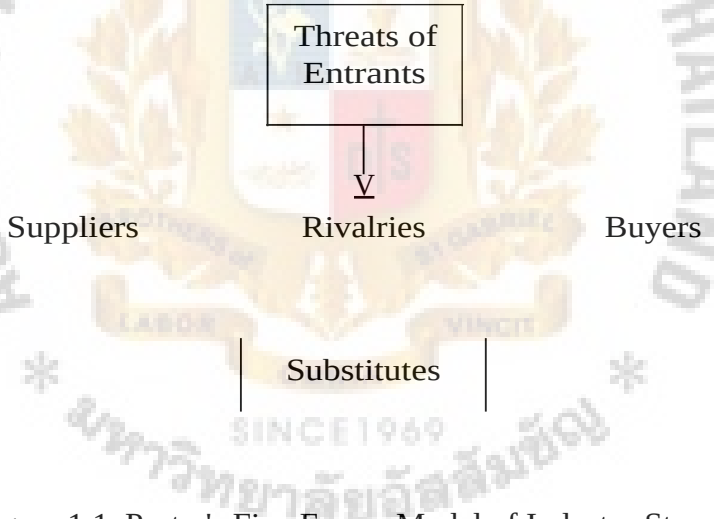


Figure 1.1. Porter's Five Forces Model of Industry Structure.

On one level, the barriers to entry into the company business are relatively low. To open a mill and factory warehouse that process and stock raw materials and finished products require a large amount of capital or experience. On the other hand, to import a bit of finished product for abroad is not so hard. The company's management is not worried about the domestic mills and factories. They are worried about such things as dumping from foreign which has government subsidiary in production capacity and

financial position. That thing would cause the domestic price to be down. The reason behind is that after the world economic crisis, the demand of the steel product was decreasing. In the other hand, the supply of steel product was increasing. There are an over producing of both raw material and finished product over the world. Most of countries begin to protect the domestic market by imposing the tariffs and the anti-dumping regulations.

As for rivalries in the industry, the company is most concerned about two types of rivals: the large importers and the large domestic producers. Nowadays the government begins to concern about the domestic market conditions and impose some regulations to the importers. The tariff regulation imposed to the importers who imported semi-finished product and finished product was up to 30% on imported price. That would increase the cost of the imports by 30%. Recently, the government imposed the anti-dumping regulation to the importers, which specified to some countries such as Korea, Japan, Indonesia, Russia, and India. In the past, without the government's tariff and anti-dumping regulations, the steel product's domestic market share was dominated over 50% by imported product. The company, at that time, is in the trouble.

The company market share, this year, is expected to 80% for domestic only. The domestic consumption for steel product this year is expected to more than 3 million tons. With the several factors such as the company's forty-year knowledgeable in steel industry, the high power in supply bargaining, and the subsidiary from the government, the company's revenue has increased steadily.

1.2 Objectives of the System

The objectives of the development of proposed system are to increase the effectiveness and efficiency of sales order processing. The development of proposed system will automate the company's sales department transactions including storing,

retrieving, and manipulating which is existingly done by manual. Next, the development of the new proposed system will reduce the work redundancies and incorrectness resulting from the existing manual system. Finally, the development of new proposed system will improve the customer service through the fast of the system operations.

The new proposed system should; prepare the reusable and sharable information within the department, provide the data entry screen that easy to implement, reduce workload such as paperwork in the process, generate the update and accuracy reports, and provide the efficient and sufficient software and hardware.

1.3 Scope of the System

The scope of the development of sales ordering system include an analysis of the existing manual sales ordering system and a development of the new computerized of sales ordering. The development of sales ordering system includes the module to process all related data in sale order processing.

1.4 Company History

Sahaviriya Panich Company started as one of the trading companies that Sahaviriya's management no longer fears. At first, the company opened in the Yaowarach Mall, Bangkok, in 1955. Wit and Prapa Viriyaprapaikit, brother and sister, who are the founder and the current chairman of the board. Wit Viriyaprapaikit had immigrated to Bangkok from his native China in 1940. At that time, he studied in high school and work at his sister shop as a worker. He did all everything in the shop. In 1954, after the dead of his sister's husband, they began to settle their own company for survival. With nine years of experience in trading steel products, Wit began with trading the steel's scrap. Next, he began to import steel products from Asia countries to sell in domestic. The imports included the semi-finished steel products, which further

processed to be finished steel products for final usages. Until the time of World War 2, there was an under-supply of steel products in Thailand. Although huge amount of steel products were imported from abroad in that time, the domestic consumption's demands were not fulfilled. With the situation of steel product's shortage in that time, Wit caught the opportunity to grow. Also with the initial importer who imported steel products from abroad, he had a better chance that the other's domestic traders. In that time, not only the used steel products that he sold, but also the new steel products that were imported and domestically produced. The imported steel products were imported from Asia countries such as Indonesia, Malaysia, Korea, and China.

The company's product lines were increasing. The company's knowledgeable in steel industry can help the company in doing business with the other partners from foreign. Many foreign steel product's manufacturers selected the company to be their product representative agent In Thailand. The East Asiatic and Louis-E-Lenovan, the two well known steel product's manufacturers who selected Sahaviriya Panich Company to trading products in Thailand.

The company also joined in the auctions of used-steel products. In the auction, the used-steel products were came from the scrap iron of planes, rails, tanks, and factory's machines, which were destroyed during the Second World War. After biding the used-steel products from the auction, the scrap iron was stripped and resold to retailers in the market. The company's opportunities in the auctions enhanced the company's deep knowledge and experiences in steel industry such as steel specifications, dimensions, properties, and chemical compositions. In the auction, the company learned not only steel knowledge, but also how to quote the price, and forecasting the steel demand.

In 1962, Wit and Prapa Viriyaprapaikit registered the Sahaviriya Panich to be the company and moved to the new location. The previous location was small in area. The

new location on Rama 3 road, Yannawa, Bangkok, has an area of twenty rais. The company location at Rama 3 road consisted of ten rais of warehouse, three rais of office place, seven rais of parking area. For the company's machines and equipments in the warehouse, there were twenty cranes of loading and unloading the product, the two of scales for weighing the loaded and unloaded truck.

During that time, sales were steady and profits increased. And for the first time the company visited the Singapore's manufacturer of the round bar steel product, the company negotiated and was able to be the representative for their product in Thailand. Next, the company could negotiate to be the representative of many Asian countries steel product's manufacturers and the company expanded the line of steel products in the latter year. As a result, cost was decreased and profits increased.

Next two years, in 1964, Wit Viriyaprapaikit built the round bar mill in Prapadeang. By imported raw material from abroad, the company could process the raw material and sold the finished round bar steel to the customer. Predicting that his mill had already met the domestic consumption's demand., he decided to built the other two mills in nearby. The mills were produced Light-Lips Channel product and Angle bar product. The company continued to grow and Sahaviriya Group expanded beyond existing product lines.

Although the company offered the steel products at a lower quality comparing with the steel products offered by the other importers, but the prices of steel products the company offered also cheaper than the other offered. Sahaviriya Group's mills imported raw materials from abroad such as billets, slabs, ingots, and cobble plates. The mills refurnanced that raw materials, and formed into semi-finished and finished steel products. The semi-finished and finished steel products were sold through the Sahaviriya Panich Company, the sole agent.

Also, the Sahaviriya Group's mills were inefficient in the production due to the old model of the machines. That pushed up the production cost of the mills. Therefore, Wit Viriyaprapaikit took over the Bangkok Steel Work Mill and their operations in 1968. That could improve the operations of the mill and decreased the production cost.

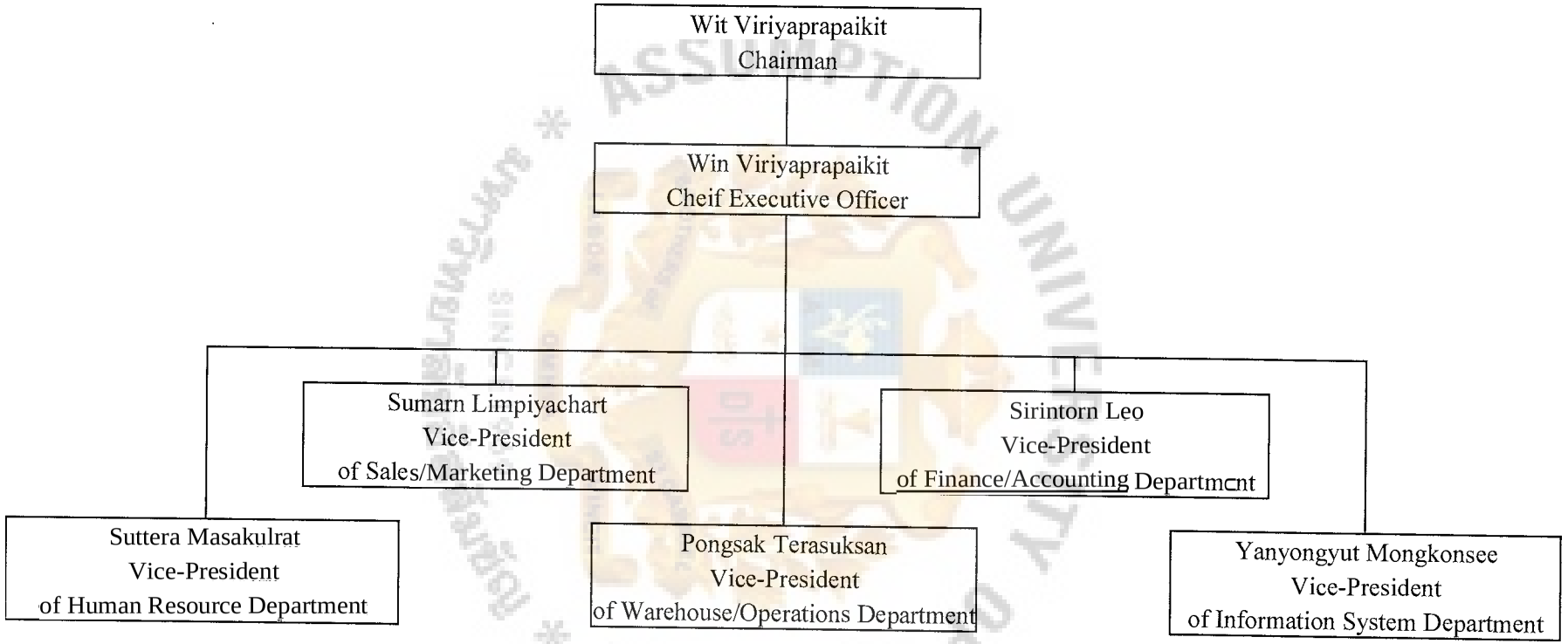
When Wit Viriyaprapaikit first started his business in 1962, he had the insight to not trading only few kinds of steel product. His insight paid off as emphasizing few kinds of steel product soon would not grow. Other steel trading companies had imported the same steel products as Sahaviriya Panich offered, but the were too small in volume and higher in imported prices. Soon some began to built their own mill and imported raw materials from abroad.

In 1990, Wit Viriyaprapaikit saw the opportunity to combine varieties of steel products into one place. Sahaviriya Panich Company was the only one who offered a wide variety of steel product in Thailand. Sales steadily grew and the customer's response was overwhelming. Profits increased until 1995 and revenues were over one billion bahts.

1.5 Organization Chart of the Company

Since 1962 when Wit Viriyaprapaikit founded the company, he has been intimately involved in the growth and the development of Sahaviriya Groups. In 1992, when the company was about to expand its product lines. He decided that he no longer wanted to be the chief executive officer of the company. He decided to fill only the position of chairman and he promoted his son, Win Viriyaprapaikit to the officer of president and CEP. The Organization chart of the company is shown in Figure 1.2.

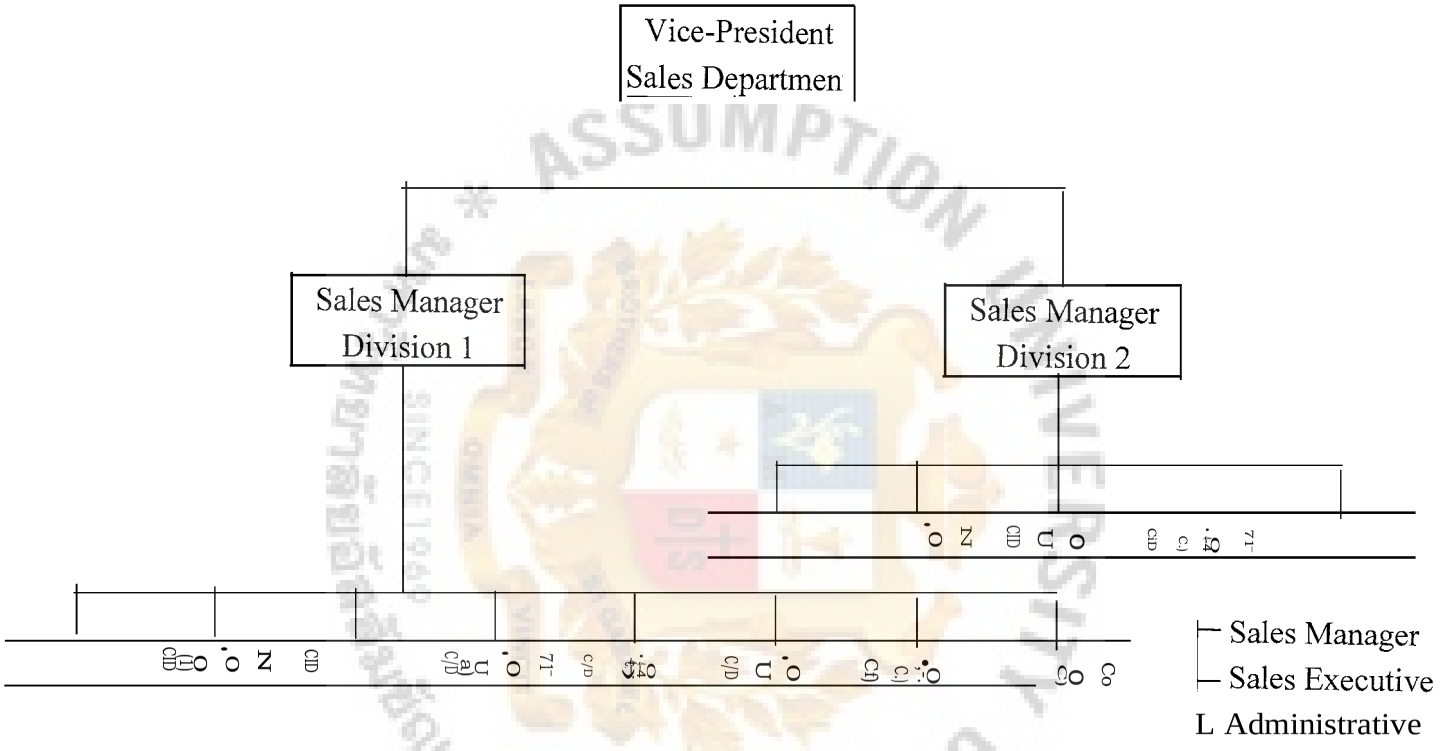
Most of the company's other senior officers have also been promoted from within. Sinrintorn Leo, the chief financial officer, had been with the company from the beginning, starting out as the fledgling company's bookkeeper and accountant.


Wit Viriyaprapaikit
Chairman
Win Viriyaprapaikit
Cheif Executive Officer
Sumarn Limpiyachart
Vice-President
of Sales/Marketing Department
Sirintorn Leo
Vice-President
of Finance/Accounting Department
Suttera Masakulrat
Vice-President
of Human Resource Department
Pongsak Terasuksan
Vice-President
of Warehouse/Operations Department
Yanyongyut Mongkonsee
Vice-President
of Information System Department

Yanyongyut mongkonsee had been part of the outside consulting team that had built the company's first information system for the company in 1992. He became the vice president in charge of information system for the company in 1995. Suttea Masakulrat, the vice president for human resources, had been with the company since 1980 and had worked her way up to the vice president level. An exception to the promote-from-within tendency, Pongsak Terasusan, the vice president of warehouse and operation, had been recruited from Bangkok Steel Bar Mill in 1991. Sumarn Limpiyachart had been hired in 1980 to be the administrative staff for the first time, and had been promoted in 1985 to be the sales executive, and in 1990 to be the vice president of sales and marketing.

1.6 Organization Chart of Sales/Marketing Department

The Sahaviriya Panich Company's sales and marketing department was primarily separated into two main divisions; Sales division one and sales division two. Each sales division was assigned to supervise each group of product lines, which had hot-rolled steel products and cold-rolled steel products respectively. There were also eight product sections in sales division one and five product sections in sales division two. For each section, there were one or more of product lines. In each section, there also were one sales manager, one assistant manager or sales executive, and one administrative staff. The organization chart of Sales department is shown in Figure 1.3.



Sales Department Chart.

MS (CEM)
St. Gabriel's Library, Au
2140

II. AN EXISTING SYSTEM

2.1 Information System

Sahaviriya Panich Company operated from 1962 until 1990 without any information system support. During the first five years of business, the company generated the revenue of 120 million bahts. Like many businesses this size, Wit Viriyaprapaikit, the company owner, did not have the expertise or the capital for developing the company's own information systems. Sirintorn Leo did all of the company business, including the management of inventory, by hand until she contacted the personal computer distributors in local area to support some general application software such as accounting, inventory handling software. After she had successfully converted the company accounting and other business functions to work by using prepackaged software purchased from a local computer distributors, she wondered how she had ever worked without it. She told anyone who would listen that computerizing the company had made the expansion much easier. In 1995, Wit Viriyaprapaikit expanded the company's product lines to more than 10 product lines, and by 1996, the company had expanded to fifteen product lines and was considering its first public offering. The change in business scope would clearly swamp Sirintorn Leo and her computerized potential. The dramatically increased demands on inventory alone seemed to warrant increased computerized.

Sahaviriya Panich Company had nobody trained in information systems on staff and all the company managers were quite busy coping with the business expansion. Wit Viriyaprapaikit and Sirintorn Leo considered hiring a small information system department from outside, but they were reluctant to do so since they did not know how

a new group should be managed, how to select quality staff, or what to expect from such employees.

From their limited experience with computers, Wit and Sirintorn realized that computer software could be quite complicated, and building systems for a rapidly changing organization could be quite a challenge. They also knew that building information systems required discipline and this discipline needed to be a part of systems development from the beginning. Thus, they wanted experienced people, but had no idea how much effort was involved or what expertise would be required. So he, after contacting several other local businesses, contacted the information-consulting firm of local area, about designing and building a custom computer-based information system for the company.

Wit and Sirintorn wanted the new system to perform all of the functions Sirintorn's current system could handle, i.e., accounting, payroll, inventory, and customer tracking. Wit wanted the system to be readily expandable as he was planning for the company's rapid growth. At the operational level, Wit realized the steel business would require new features in their information system, features not need for the old system. For one thing, the company's customers would not only be buying product from the company, they would also be tracking the product status. Further, customers would be required to track with the system and report the product status.

At a management level, Wit wanted the movement of the products in and out of the company and all customer account's computerized and easy to track. Being new to the steel trading business, Wit also wanted to be able to search through the data on the company's customers describing their buying habits.

He wanted to know which products were the most selling, and he want to know who were the most buyer, not only in Bangkok city but in every province where the

company did business. Acer Tech, Inc., was happy to get the company's account. They assigned Yanyongyut Mongkonsee to head up the development team. Yanyongyut led a team of his own staff of analysts and programmers, along with participation by several company's managers, in a thorough analysis and design study. Acer Tech. had recently adopted a CASE tool (Computer-Aided Software Engineering) and associated methodology. This methodology provided the discipline needed for such a major systems development effort. The methodology began with information planning and continued through all phases of the systems development life cycle. Fortunately for the development team, the company had been doing annual and long-term business planning and was very cooperative at every step of the total process.

After working for almost two years on the project, Yanyongyut and his team delivered and installed the system the company had requested. The system was centralized, with the two sever of Window NT installed in the company, and twenty personal computer terminals, three centralized printers. One of the server was used to store, retrieve, edit database. The database stored the information of the customer. The printers generated documents within the company. In addition, personal computers and printers were included to handle a few office functions such as the generating receipts. The primary software product Acer Tech. Built for the company, then, monitored and updated the credit account of the customers. Another software product transferred exchanged the electronic mails within the company. By the time the system was installed and running at the end of 1995, the company revenue was increased to 0.8 billion baths, and the system was able to handle the company's business demands with few problems.

In 1996, Yanyongyut Mongkonsee left Acer Tech. To start his own consulting firm. In 1997, he joined Sahaviriya Panich as the head of its information system group.

He led the effort to expand and enhance the company's information system as the company's revenue grew to one billion bahts in 1999. Sahaviraya Panich Company now relies on Window NT servers and Intel Celeron 600 computer terminals. Yanyongyut and his team have been thinking seriously about designing and developing the company's sales ordering system and changing to a client/server network. None of these issues, however, have been resolved.

2.2 Application Portfolio

Sahaviriya Panich Company (SVP) has developed computer applications for both sales department as well as other functional departments. The company has system development and support groups, which responsible for Sales department's applications and other functional department's applications. The other functional departments are further divided by functional areas, such as Accounting, Marketing, Human Resources, and Operations. Each development group also has liaison staff with sales department group, since data in other functional departments feed or are fed by sales department's applications. The company creates both one-year and three-year information system plans that encompass both Sales and other departments. Domestic and export management operations are supported within the company information plan and development group.

For the company, the Sales department's original functions have changed very little since 1982, customer and inventory tracking is done by manual system and basic counting. Daily transactional data (purchasing, selling, other changes in inventory as well as employee time in and out) are all captured and transmitted in the manual system.

As shown in Figure 1.3, each sales section has personal computer that serves as a terminal for application software. The computer also serves as the section manager's

workstation for generating reports and performing other administrative in-section application. Some managers have also learned how to use spreadsheet, word processing, and other packages to handle functions not supported by systems provided by the company for in-section use. Each computer in the sales section includes a modem to allow communication with the other terminal computer within the company via a communications front-end processor. The front-end processor at the company offloads communications traffic from the Oracle servers so that it can concentrate on data processing applications. The company databases are managed by the Oracle's relational.

All inventory control (e.g. purchasing of new items, retirement of items) is done centrally and, since each sales section is the company owned, sales manager, sales executive, and administrative staff are paid by the company. Today, each sales section is an isolated section; that is, each section has manual records of only its own activity, including inventory. Profit and loss, balance sheets, and other related reports are produced for each section by decentralized systems. Each section manager is responsible for all in-section personnel matters (hiring, evaluating, scheduling, and firing employees), section property management (day-to-day dealings with property owner if the property is leased), managing section promotions within a budget set by the company for each section, and customer relations. The section manager also influences stocking decisions, but final decisions are made at the company conference based on section transaction history and market trends. Practically all other functional departments are managed centrally.

A rapidly expanding business, the company has created significant growth for the information system group. To date, the company's Sales department and support staff, but the company is considering reorganizing its staff t provide focused attention to this

important areas. Since the company wants to invest as much as possible in expansion, managers have been changed with increasing productivity without adding full-time staff. The company still use services of Acer Tech, but on a very selective basis, such as when Information System department's resources are fully committed. Information System department includes four developers (programmer, analyst, specialist in database, and net working) plus data center staff and is now large and technically skilled enough to handle almost all requests. The company has created as professional as environment in the company's information system group as exist at Acer Tech. For example, the company recently acquired a CASE tool and adopted a standardized system development methodology consistent with this tool.

The company's current challenge in managing the information system plan is keeping its staff current in the skills they need to successfully support the systems in a rapidly changing and competitive business environment. Besides adapting to new technical skills, information system staff needs to be excellent project managers, to completely understand the business, and to exhibit excellent communication with clients and other. The company is also concerned information systems literacy among the company management and that technology is not being as thoroughly exploited as it could be.

To deal with this situation, Information System department is considering several initiatives. First, Information System department has requested a sizeable increase in its training budget and has asked the company to consider expanding the benefit of its college tuition reimbursement program. Second, Information System department is considering instituting a management development program that will better develop junior staff members and will involve various user departments. As part of this program, the company personnel will rotate in and out of the information system group

as part of normal career progression. This program should greatly improve relationships with user departments and increase end-user understanding of technology. The development of this set of technical, managerial, business, and interpersonal skills in and outside information system is a critical success factor for the company in responding to the significant demands and opportunities of the company's business area.

The company has a hardware and software environment that is similar to that used by many other business companies. The company's sales section has a computer terminal with software program that run mainly application programs, such as Microsoft offices. Other function's application systems are executed on a main server at a center. The company's other function applications handle all accounting, banking, property, inventory tracking, and other applications that involve data from all sections.

The company is a rapidly growing business with significant demand for information service. The company has divided its staff into functional area groups for both sales and other functions needs. The company uses modern database management and programming language, technologies, including a CASE tool. The company information system is challenged by keeping current in both business and technology areas.

2.3 Operation System

Sahaviriya Panich Company (SVP) sells steel products to customers. Whenever a customer needs to purchase steel product from the company, it issued a purchase order to the company's Sales department. In the purchase order issued by the customer, it contained info!! iation of product descriptions. The product information included product dimensions, descriptions, prices, amounts, weight, condition of payments, delivery date, and shipment place. Another information of customer that necessarily

contained in the purchase order are customer name, address, telephone number and other necessary information such as date of order and approval signature. After receiving the purchase order from the customer, sales staff needs to check the information in the enquiry form whether its conform with the company standard of information. For example, there may be some error in the product dimension's information in the order form that occurs as a result of the difference in the understanding of standard dimensions. Another example is that the customer may send the wanted prices instead of the negotiated prices. Therefore, if there is some information incomplete in the order form, sales staff suddenly needs to contact the customer to correct it. After checking the correctness of information in purchase order, the sales staff has to check whether the warehouse and operations department has enough amount of product as the customer required. By checking the availability of inventory that the warehouse and operations department prepared for the sales department on daily basis, sales staff would know whether the company can fulfill the customer needs. As soon as sales staff has known the availability of inventory status, sales staff needs to inform the inventory status to the customer. After knowing that the information in product enquiry form conformed to the standards and the warehouse and operations has the available inventory, finally, sales staff needs to check the credit line of the customer if the customer wants to purchase on the credit basis. Customer's credit line checking is done by enter into the Customer Credit Line Checking which is prepared and updated by the financial and accounting department. In checking the customer credit line, sales staff is only inserting the customer account into the box, the system will automatically process and show the outstanding balance of that customer. The system prints the outstanding amount credit that customer can use for further purchase. If there are no outstanding allowance of credit, the system will print in minus

or red number. For the customer who has the over-credit allowance, it needs to transfer the cash into the company account before receiving the product. Sales staff will inform customer the company's account number. For the customer who has an outstanding balance, sales staff will accept and issue the sales confirmation. For the customer who has not any credit line with the company, they need to transfer cash before receiving the product.

When a purchase order is received by a sales staff, the number referenced on the purchase order is manually entered into the customer purchase order's sheet. Next, sales staff needs to issue a sales contract and send it to customer. Also, sales staff needs to issue a product requisition bill and sends it to warehouse/operation department. Warehouse/operations department received a product requisition bill from sales staff prepared the product and delivered it to the customer on delivery time stated in product requisition bill. Warehouse/operations also issue delivery note or delivery order which contain exact amount of product's weight and send it along with product to customer. When the customer received the product, the customer need to check whether the product received conformed with the information contained in delivery note or delivery order.

If conformed, the customer need to sign in the delivery note or delivery order, and keep the copy of that document. If the product received is not conformed with the information in the document, the customer need to contact sales staff for further decision without signing any documents. When the customer signing the delivery note or delivery order, it needs to return it back to the company. Warehouse/operations department after receiving delivery note or delivery order, they need to send back with the product requisition bill to sales department. Sales staff receives and records the delivery note or delivery order infolination in the delivery note or delivery order sheet

manually. Sales staff records product requisition bill. Both delivery note or delivery order and product requisition bill need to be recorded for two objectives. One is for generating necessary reports such as shipment reports, sales reports, inventory outstanding reports. Another is for generating tax invoice to collect bill from the customer. In issuing tax invoice, sales staff needs to be carefully since info'_iation or items contained in tax invoice must be correct. And after approval by sales manager, it cannot be changed. When sales staff completed in issuing tax invoice, it will be send to the customer for billing. Customer accept and sign tax invoice, sends back to sales staff for issuing receipt.

There are many documents generating along the sales order processing's transactions. These documents include recording documents, transaction documents.and report documents. For different purposes, record document needed for recording necessary information which make it easy for tracking and producing reports. Transaction documents necessary for keeping transaction flow. Report documents generate during transaction for high level managers.

There are several findings incorporated into topic that influences the effectiveness and efficiency of the company's sales order processing. First, sales staff manually records the data (customer data and product data) for their own department, inaccuracy and outdate information certainly will cause delays and errors in the processing. Second, inflexibility occurs due to the transfer of the documents through the channels of people, not by electronically transfers. Third, the data (customer and product) is separately stored or kept in each section, then there are redundancy of information and delays in preparing reports.

From the findings, it found that if the company use the existing manual of sales ordering system in coping up with the changing environment, the company would lack

of effectiveness and efficiency in operating to capture the company competitiveness. In order to coping up with the changing environment and the current's situation of growth rate and also bringing the company to the wealth creation, the company must introduce the new development system and implement it with existing operation of the sales order processing as fast as possible.

III. A PROPOSED SYSTEM

3.1 Process Modeling

3.1.1 Overview

By studying from the existing operation system, the development of proposed system's process modeling can be done by drawing a data flow diagram. Data flow diagram is a picture of the movement of the data between external entities and the processes and data stores within a system. There are many different tools developed for process modeling, but the reason in selecting on data flow diagram technique is the most often used- today technique' by using data flow diagram technique, it can save for the cost of reworking to fix requirement flows. This represents a doubling of system developers' productivity and helped in avoiding costly system mistakes.

During requirement structuring, it must organize the information into a meaningful representation of the information system that exists and to the requirements desired in a proposed system. In addition to modeling the processing elements of an information system and how data are transformed in the system, it must also model the processing logic and the timing of events in the system and the structure of data within the system. Thus, a process model is only one of three major complementary views of an information system. Together, process, logic and timing, and data models provide a thorough specification of an information system and, with the proper supporting tools, also provide the basis for the automatic generation of many working information components.

3.1.2 Definition and Symbols

A data flow can be best understood as data in motion, moving from one place in a system to another. A data flow could represent data on a customer order or a pay check.

A data floe could also represent the result of a query to a database, the contents of a printed report, or data on a data entry computer display form. A data flow is data that move together, thus, a data flow can be composed of many individual pieces of data that are generated at the same time and flow together to common destinations. A data store is data at rest. A data store may represent one of many different physical locations for data , for example, a file folder, one or more computer-based file, or a notebook. To understand data movement and handling in a system, the physical configuration is not really important. A data store might contain data about customers, students, customer orders, or supplier invoices. A process is the work or actions performed on data so that they are transformed, stored, or distributed. When modeling the data processing of a system, it does not matter whether a process is performed manually or by a computer. Finally, a source/sink is the origin and destination of the data. Sources/sinks are sometimes referred to as external entities because they are outside the system. Once processed, data or information leave the system and go to some other place. Since sources/sinks are outside the system, there are many characteristic of sources/sinks that are unimportant as follows:

(1) Interactions that occur between sources and sinks.

(2) What a source or sink does with information or how it operates (that is, a source or sink is a black box).

(3) How to control or redesign a source or sink since, from the perspective of the system, the data a sink receives and often what data a source provides are fixed.

(4) How to provide sources and sinks direct access to store data since, they can not directly access or manipulate data stored within the system; that is,

process within the system must receive or distribute data between the system and its environment.

For the conventions, a data flow is depicted as an arrow. The arrow is labeled with a meaningful name for the data in motion; for example, customer order, sales receipt, or sales confirmation. The name represents the aggregation of all the individual elements of data moving as part of one packet, that is, all the data moving together at the same time. A square is used in a convention for sources/sinks and has a name which state what the external agent is, such as customer, other departments or manager. The symbol for a process is a rectangle with rounded corners. The round rectangle has a line drawn through the top. The upper portion is used to indicate the number of the process. inside the lower portion is a name for the process, such as generate bills, calculate minimum quantities, or compute total orders. The symbol for a data store is a small box used to number the data store and inside the main part of the rectangle is a meaningful label for the data store, such as customer file, purchase order, or product requisition file.

Sources/sinks are always outside the information system and define the boundaries of the system. Data must originate outside a system from one or more sources and the system must produce information to one or more sinks. If any data processing takes place inside the source and sink, it is not interested as this processing take place outside of the system. A source/sink might consist of the following:

(1) Another organization or organization unit which sends data to or receives information from the system of analyzing.

(2) A person inside or outside the business unit supported by the system of analyzing and who interacts with the system.

(3) Another information system with which the system of analyzing exchange information.

3.1.3 Rules of Data Flow Diagram

There is a set of rules that must follow when drawing data flow diagrams. These rules allow to evaluate data flow diagrams for correctness. For process symbol, process cannot have only inputs or outputs, it is making data from nothing. If an object has only outputs, then it must be a source. If an object has a only inputs, then it must be a sink. Also, a process has a verb phrase label.

For data store symbol, data cannot move directly from one data store to another data store. Data must be moved by a process. Data cannot move directly from an outside source to a data store. Data must be moved by a process, which receives data from the source and places the data into the data store. Also data cannot move directly to an outside sink from a data store. Data must be moved by a process and a data store has a noun phrase label.

For source/sink symbol, data cannot move directly from a source to a sink. It must be moved by a process if the data are any concern to any system. Otherwise, the data flow is not shown on the data' flow diagram. A source/sink has a noun phrase label.

For data flow symbol, a data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however, by two separate arrows since these happen at different times. A fork in a data flow means that exactly the same data stores, or sources/sinks. A join in a data flow means that exactly the same data comes from any of two or more different process, data stores, or sources/sinks to a common location. A data flow cannot go directly back to the same process it leaves. There must be at least one other process, which handles the data flow, produces some

other data flow, and return the original data flow to the beginning process. A data flow to a data store means update and a data flow from a data store means retrieve or use. A data flow has a noun phrase label. More than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Besides the rules described above, there are two data flow diagram guidelines that apply most of the time. First, the inputs to a process are different from the outputs of that process. The reason is that processes, to have a purpose, typically transform inputs into outputs, rather that simply pass the data through without same manipulation. The same input goes in and out of a process but the process also produces other new data flows that are the result of manipulating the inputs. Second, objects on a data flow diagram have unique names. Every process has a unique name. There is no reason to have two processes with the same name. To keep a data flow diagram uncluttered, it needs to repeat data stores and sources/sinks. When two arrows have the same data flow name, it must be careful that these flows are exactly the same. It is easy to use the same data flow name when two packets of data are almost the same, but not identical. A data flow name represents a specific set of data, and another data flow that has even one more less piece of data must be given a different, unique name.

3.1.4 Decomposition of Data Flow Diagram

Functional decomposition is an iterative process of breaking the description or perspective of a system down into finer and finer detail, which creates a set of charts in which one process on a given chart is explained in greater detail on another chart. Each of processes or subsystems are candidates for decomposition. Each process may consist of several sub-processes. Each sub-process may also be broken down into smaller units. Decomposition continues until it has reached the point where no sub-process can

logically be broken down any further. The lowest level of data flow diagrams is called a primitive data flow diagram.

In decomposition data flow diagram, there are no sources and sinks represented. Although it may include sources and sinks, the context and level-0 diagram that is generated from *n* nest decompositions from a level-0 diagram.

Just as the labels for processes must follow numbering rules for clear communication, process names should also be clear yet concise. Typically, process names begin with an action verb, such as receive, calculate, transform, generate, and produce. Often process names are the same as the verb used in many computer-programming languages. Examples include merge, sort, read, write, and print. Process names should capture the essential action of the process in a few words, be descriptive enough of the process' action so that anyone reading the name get a good idea of what the process does.

3.1.5 Balance of Data Flow Diagram

Balance is the conservation of inputs and outputs to a data flow diagram process when that process is decomposed to a lower level. A data flow consisting of several sub-flow on a level-n diagram can be split apart on a level-n+1 diagram for a process which accepts this composite data flow as input. The principle of balancing and the goal of keeping a data flow diagram as simple as possible lead to four additional, advanced rules for drawing data flow diagrams. First, a composite data flow on one level can be split into component data flows at the next level, but no new data can be split into component data in the composite must be accounted for in one or more sub-flows. Second, the inputs to a process must be efficient to produce the outputs (including data placed in data stores) from the process. Thus, all outputs can be produced, and all data in inputs move somewhere, either to another process or to a data

store outside the process or on a more detailed data flow diagram showing a decomposition of that process. Third, at the lowest of data flow diagrams, new data flows may be added to represent data that are transmitted under exceptional conditions; these data flows typically represent error message or confirmation notices. Finally, to avoid having data flow lines cross each other, it needs to repeat data stores or sources/sinks on a data flow diagram. Use an additional symbol, like a double line on the middle vertical line of a data store symbol, or a diagonal line in a corner of a sink/source square, to indicate a repeated symbol.

3.1.6 Guideline for drawing Data Flow Diagram

(1) Completeness

The concept of data flow diagram completeness refers to the extent to which all necessary components of data flow diagram have been included and fully described. If the data flow diagram contains data flows that do not lead anywhere, or data stores, processes, or external entities that are not connected to anything else, the data flow diagram is not complete.

(2) Consistency

The concept of data flow diagram consistency refers to the extent to which information contained on one level of a set of nested data flow diagrams is also included on other levels.

(3) Timing

For the data flow diagram, there is no indication of whether a data flow occurs constantly in real-time, once per week, or once per year. There is also no indication of when a system would run. Therefore, it needs to draw the data flow as if the modeling system has never started and will never stop.

(4) Iterative Development

When drawing data flow diagram, the first data flow diagram will rarely capture perfectly the system of modeling. It should count on drawing the same diagram over and over again, in an iterative fashion. With each attempt, it will come closer to a good approximation of the system or aspect of the system of modeling. Iterative data flow diagram development recognizes that requirements determination and requirements structuring are iterating, not sequential sub-phases of the analysis phase of the system development life cycle.

(5) Primitive Data Flow diagram

One of the more difficult decisions when drawing the data flow diagram is when to stop decomposing processes. It should stop in decomposing the data flow diagram when it has reached the lowest logical level. It has reduced each process to a single decision or calculation or to a single database operation, such as retrieve, update, create, delete, or read. Each data store represents data about a single entity, such as a customer, employee, product, or order. The system user does not care to see any more detail, or when it has documented sufficient detail to do subsequent systems development tasks. Every data flow does not need to be split further to show that different data are handled in different ways. It has been shown each business form or transaction, computer on-line display, and report as a single at flow. There is a separate process for each choice on all lowest level menu option for the system.

3.1.7 Development of Data Flow Diagram

(1) Context Data Flow Diagram

Context data flow diagram, a highest-level view, is an overview of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system. In the context data flow diagram, it contains only one process, no data stores, some data flows and sources/sinks. The single process labeled "0", represent the entire system; all context data flow diagrams have only one process labeled "0". The sources/sinks represent its environmental boundaries. Since the data stores of the system are conceptually inside the one process, no data stores appear on a context data flow diagram. The context data flow diagram of the company's sales ordering system is shown in Figure 3.1.

For the context data flow diagram of the company's sales ordering system, there are three sources/sinks, one process, twelve data flows, and no data store. Since it is the rule of data flow diagramming. It has not a data store in the context data flow diagram. For the context data flow diagram's sources/sinks, there are: (1) Customer (2) Warehouse/Operations, and (3) Finance/Accounting. For the context data flow diagram's process, it has only one process which is the company's sales ordering system itself. According to the rules of data flow diagramming, it can has only one process in the context data flow diagram. For the data flows in the context data flow diagram, there are totally twelve data flows that can divide into five input's data flows and seven output's data flows. For five input's data flow in the context data flow diagram, there are: (1) Customer's Purchase

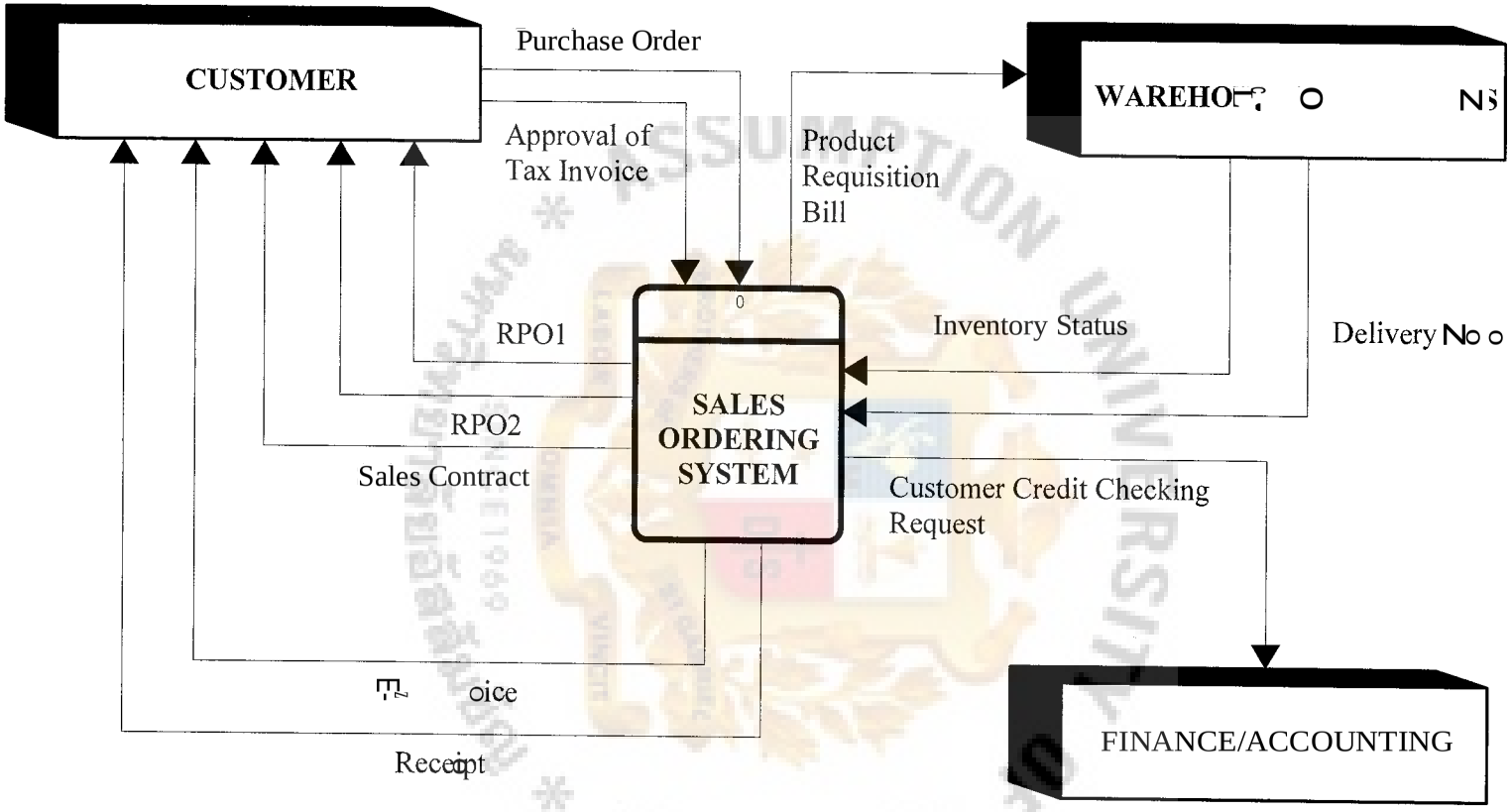


Fig 3.1. Context Data Flow Diagram of Sales Ordeing System.

Order, (2) Inventory Status, (3) Customer Credit Checking Result, (4) Delivery Order/Delivery Note, and (5) Approval of Tax Invoice. For seven output's data flows in the context data flow diagram, there are: (1) Rejection of Purchase Order (Unavailable Inventory), (2) Customer Credit Checking Result, (3) Rejection of Purchase Order (Unavailable credit), (4) Sales contract, (5) Product Requisition Bill, (6) Tax Invoice, and (7) Receipt. The purpose of this process in the context data flow diagram is to perform the sales ordering's operations.

(2) Level-0 Data Flow Diagram

The next step for the analysis is to think about which process are represented by the single process in the context diagram. Level-0 data flow diagram is a data flow diagram that represents a system's major processes, data flows, and data stores at a high level of detail. This data flow diagram represents the primary individual processes in the system at the highest possible level. Each process has a number, which ends in 0 (corresponding to the level number of the data flow diagram). The level-0 diagram of the company's sales ordering system is shown in Figure 3.2.

For the level-0 data flow diagram, there are eleven processes, twenty-five data flows, three sources/sinks, and six data stores. For the processes in the level-0 data flow diagram, there are eleven processes to perform system functions. These processes are decomposed from the single process in the context data flow diagram. For the processes in the level-0 data flow diagram, there are: (1) Get Valid Purchase Order Data and Valid Inventory Status Data, (2) Match Valid Purchase Order Data and Valid Inventory Status Data, (3) Access Customer Credit Checking System, (4) Get Valid

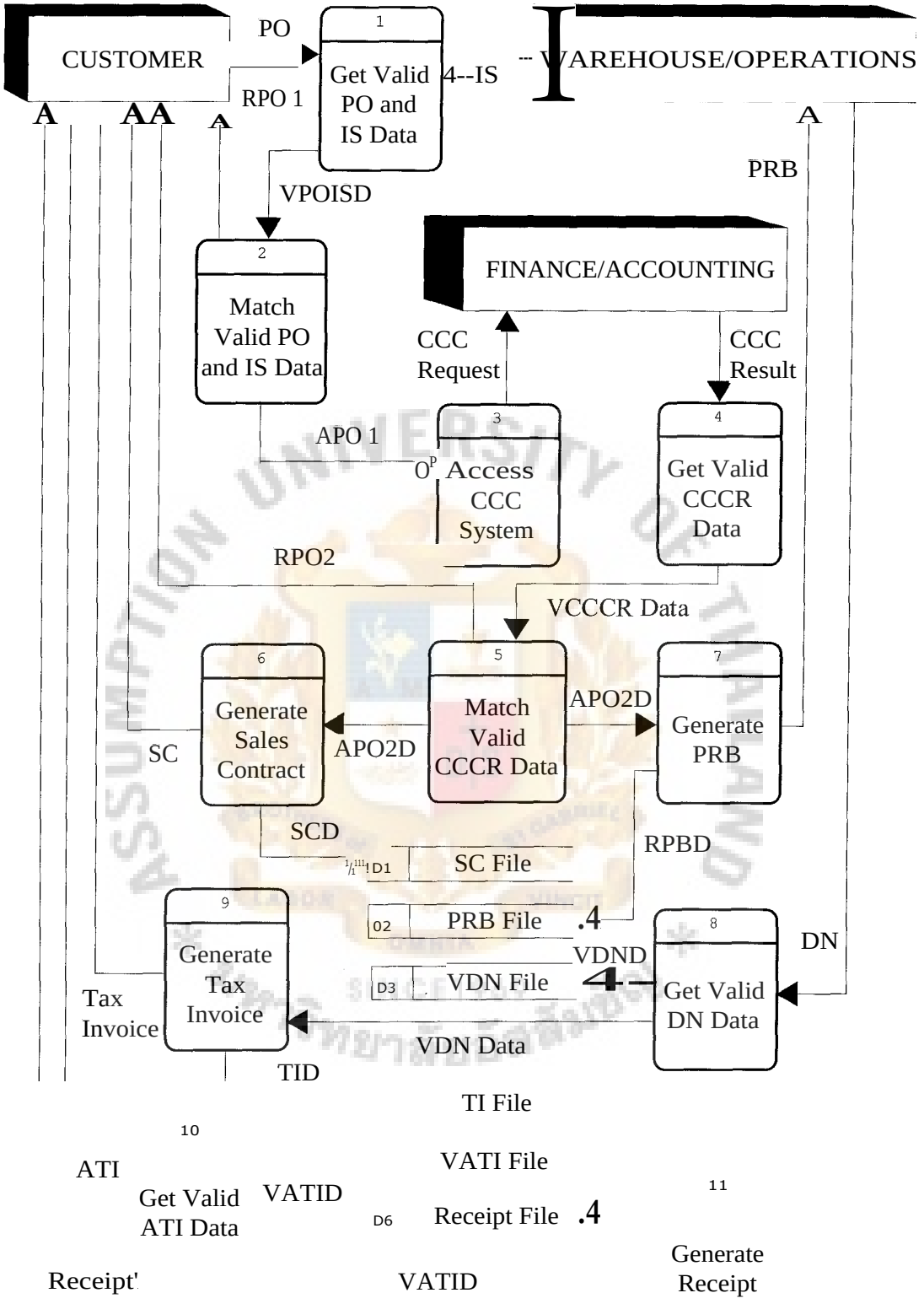


Figure 3.2. Level-0 Data Flow Diagram of Sales Ordering System.

Customer Credit Checking Result Data, (5) Match Valid Customer Credit Checking Result Data, (6) Generate Sales Contract, (7) Generate Product Requisition Bill, (8) Get Valid Delivery Order/ Delivery Note Data, (9) Generate Tax Invoice, (10) Get Valid Approval of Tax Invoice Data, and (11) Generate Receipt.

For the data flows in the level-0 data flow diagram, there are twenty-five data flows to represent the data or information transaction within and outside the system. Seven additional data flows (from the context data flow diagram) represent the data flows that flow within the system, and six additional (from the context data flow diagram) data flows represent the data flows that flow to data stores. For seven data flows (within) in this level-0 data flow diagram, there are: (1) Valid Purchase Order Data and Valid Inventory Status Data, (2) Acceptation of Purchase Order (Available Inventory), (3) Valid Customer Credit Checking Result Data, (4) Acceptation of Purchase Order (Available Credit), (5) Acceptation of Purchase Order (Available Credit), (6) Valid Delivery Order/Delivery Note Data, (7) Valid Approval of Tax Invoice Data. For six data flows (to-data stores) in this level-0 data flow diagram, there are: (1) Sales Contract Data, (2) Product Requisition Bill Data, (3) Valid Delivery Order/Delivery Note Data, (4) Tax Invoice Data, (5) Valid Approval of Tax Invoice Data, and (6) Receipt Data.

For the sources/sinks in the level-0 data flow diagram, there are three sources/sinks as same as sources/sinks represented in the context data flow diagram.

For the data stores in the level-0 data flow diagram, there are six data stores representing the place of data storing. Six data stores are: (1) Sales Contract File, (2) Production Requisition Bill File, (3) Valid Delivery Order/Delivery Note File, (4) Tax Invoice File, (5) Valid Tax Invoice File, and (6) Receipt File.

In level-0 data flow diagram, there are eleven processes to perform system functions of sales ordering. These eleven processes are decomposed from the sales ordering system in the context data flow diagram. For process 1.0, Get Valid Purchase Order Data and Valid Inventory Status Data, the purpose of this process is to get valid data from the customer's purchase order and Warehouse/Operations department's inventory status report. In getting the valid data from those document, it needs to check the conformation of data contained in the customer's purchase order and get data of the specific inventory from the inventory status report. The results or outputs of this process are Valid Purchase Order Data and Valid Inventory Status Data.

For process 2.0, Match Valid Purchase Order Data and Valid Inventory Status Data, the purpose of this process is to match the results received from the previous process 1.0 that is Valid Purchase Order Data and Valid Inventory Status Data. The method is that if the company has an inventory to satisfy the purchase order, the process is to accept that purchase order. If not, the process is to reject the purchase order. For the process outputs, the process generate the Acceptation of Purchase Order (Available Inventory) for the accept, and generate Rejection of Purchase Order (Unavailable Inventory) for the reject.

For process 3.0, Access Customer Credit Checking System, the purpose of this process is to access into the Customer Credit Checking System of Finance/Accounting Department in order to check the credit allowance of that customer. The input of this process is Acceptation of Purchase Order (Available Inventory) The output of this process is Customer Credit Checking Request.

For process 4.0, Get Valid Customer Credit Checking Result Data, the purpose of this process is to get valid customer credit checking result from the system. The input into this process is the Customer Credit Checking Result and this process needs to transform this result into usable data for the further process. The result or output of this process is the Valid Customer Credit Checking Result Data.

For process 5.0, Match Valid Customer Credit Checking Result Data, the purpose of this process is to match and decide whether to accept or reject the order. If the customer credit checking result show the outstanding credit balances of that customer, it can accept the purchase order. The input of this process is valid data of checking result which got from the process 4.0, and this process generates three outputs which are: (1) Rejection of Purchase Order (Unavailable Credit), (2) Acceptation of Purchase Order (Available Credit) to process 6.0, and (3) Acceptation of Purchase Order (available Credit) to process7.0.

For process 6.0, Generate Sales Contract, the purpose of this process is to get data from process 5.0, which is Acceptation of Purchase Order (available Credit), and generates sales contract to the customer for keep evident. Therefore, the outputs of this process have two outputs. One is

Sales Contract and another is Sales Contract Data, which is store at the Sales Contract File.

For process 7.0, Generate Product Requisition Bill, the purpose of this process is as same as the previous process (process 6.0) that is to get Acceptation of Purchase Order (available Credit) from the process 5.0, and generates Production Requisition Bill to Warehouse/Operations department. The Production Requisition Bill generated by this process will sent to Warehouse/Operations department in order to release the product to the customer. This process also generates the Product Requisition Bill Data for storing at the Production Requisition Bill File. This file data will be needed to prepare report for the manager.

For process 8.0, Get Valid Delivery Order/Delivery Note Data, the purpose of this process is to get delivery order/delivery note from the Warehouse/Operation department, and generates Valid Data of Delivery Order/ Delivery Note for the next process. The delivery order/delivery note received from the Warehouse/Operations department is document generating before the delivery of the product to the customer. The information in the delivery order/delivery note is necessary for the next process, which is the generating of tax invoice. The outputs of this process are Valid Delivery Order/Delivery Note Data. One is generated for the process 9.0, one for stored in the Valid Delivery Order/ Delivery Note File.

For process 9.0, Generate Tax Invoice, the purpose of this propose is to get the Valid Delivery Order/Delivery Note Data from the process 8.0, and generates tax invoice for the customer. Another output of this process is Tax Invoice Data. This data is stored at the Tax Invoice File. In generating

tax invoice for the customer, it needs to be careful since the data contained in the tax invoice should not be errors.

For process 10.0, Get Valid Approval of Tax Invoice Data, the purpose of this process is to get the approval of tax invoice from the customer, and get valid data of the approval of tax invoice. After get valid approval of tax invoice data, the result data are input into the process 11.0, and another result data is stored at Valid Approval of Tax Invoice File.

For process 11.0, Generate Receipt, the purpose of this process is to generate receipt for the customer and receipt data for storing in the receipt file.

(3) Level-1 Data Flow Diagram

There are eleven processes in the level-0 data flow diagram. Level-1 data flow diagram is the data flow diagram that shows the decomposition from the level-0 data flow diagram. By an iterative process of breaking the description of a system down into the finer and finer detail which creates a set of charts in which one process on a given chart is explained in greater detail on another chart, level-1 data flow diagram is the result of the decomposition of the level-0 data flow diagram.

For process, 1.0, Get Valid Purchase Order Data and Valid Inventory Status Data, this process is decomposed into two sub-processes which are: process 1.1, Get Purchase Order Data and Inventory Status Data, and process 1.2, Get Valid Purchase Order Data and Inventory Status Data. Process 1.1 gets purchase order from the customer and inventory status from Warehouse/Operations department, and gets both data. Next, process 1.2 validates those data. And the outputs of this process are Valid Purchase

Order Data and Valid Inventory Status Data. The data flow diagram showing the decomposition of process 1.0 is shown in Figure 3.3.

For process 2.0, Match Valid Purchase Order Data and Valid Inventory Status Data, this process is decomposed into three sub-processes which are: process 2.1, Match Valid Purchase Order Data and Valid Inventory Status Data, process 2.2, Reject the Purchase Order, and process 2.3, Accept the Purchase Order. Process 2.1 matches data and generates the matched result which are matched data and unmatched data. Process 2.2 gets unmatched data from process 2.1 and produces the Rejection of Purchase Order (Unavailable Inventory). Process 2.3 gets matched data from process 2.1 and produces Acceptation of Purchase Order (Available Inventory). The data flow diagram showing the decomposition of process 2.0 is show in Figure 3.4.

Process 3.0, Access Customer Credit Checking System, this process is decomposed into three processes which are: process 3.1, Calculate Purchase Amounts, process 3.2, Access Customer Credit Checking System, and process 3.3 Get Customer Credit Checking Result. Process 3.1 gets the Acceptation of Purchase Order from process2.0 and calculates the purchase amounts of the customer. Process 3.2 accesses the Customer Credit Checking System and process 3.3 gets Customer Credit Checking Result. The data flow diagram showing the decomposition of process 3.0 is show in Figure 3.5.

Process 4.0, Get Valid Customer Credit Checking Result Data, this process is no decomposition since it functions only one process. This process gets customer credit checking result from the previous process, and

generates Valid Customer Credit Checking Result for the next process. The data flow diagram showing the decomposition of process 4.0 is show in Figure 3.6.

Process 5.0, Match Valid Customer Credit Checking Result Data, this process is decomposed into three processes which are: process 5.1, Match Valid Customer Credit Checking Result Data, process 5.2, Reject the Purchase Order (Unavailable Credit), and process 5.3, Accept the Purchase Order (Available Credit). Process 5.1 gets valid customer credit checking result data from process 4.0 and produces matched data and unmatched data. Process 5.2 gets unmatched data from process 5.1 and produces Rejection of Purchase Order (Unavailable Credit). Process 5.3 gets matched data from process 5.1 and produces Acceptation of Purchase Order (Available Credit). The data flow diagram showing the decomposition of process 5.0 is show in Figure 3.7.

Process 6.0, Generate Sales Contract, this process is no decomposition. This process gets Acceptation of Purchase Order (Available Credit) from the previous process and generates Sales Contract for the customer. This process is also generate Sales Contract data to be stored in the sales contract file. The data flow diagram showing the decomposition of process 6.0 is show in Figure 3.8.

Process 7.0, Generate Product Requisition Bill, this process is no decomposition. This process also gets Acceptation of Purchase Order (Available Credit) from the previous process and produce Production Requisition Bill for Warehouse/Operations Department. This process is also produce Production Requisition Bill Data for the Product Requisition Bill

File. The data flow diagram showing the decomposition of process 7.0 is show in Figure 3.9.

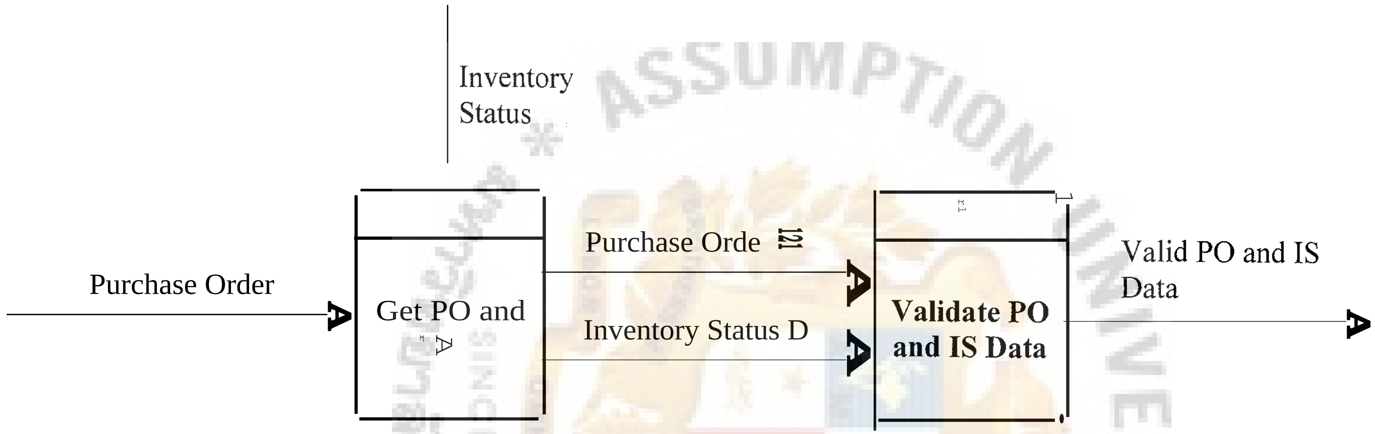
Process 8.0, Get Valid Delivery Order/Delivery Note Data, this process is no decomposition. This process gets Delivery Order/Delivery Note from Warehouse/Operations department and produce Valid Delivery Order/Delivery Note Data for process 9.0 to generate tax invoice. This process is also produce Valid Delivery Order/Delivery Note Data to store in the Valid Delivery Order/Delivery Note File. The data flow diagram showing the decomposition of process 8.0 is show in Figure 3.10.

Process 9.0, Generate Tax Invoice, this process is also no decomposition. This process gets Valid Delivery Order/Delivery Note Data from process 8.0. This process uses that information to generate Tax Invoice for the customer. This process also produces Tax Invoice Data to store in the Tax Invoice File. The data flow diagram showing the decomposition of process 9.0 is show in Figure 3.11.

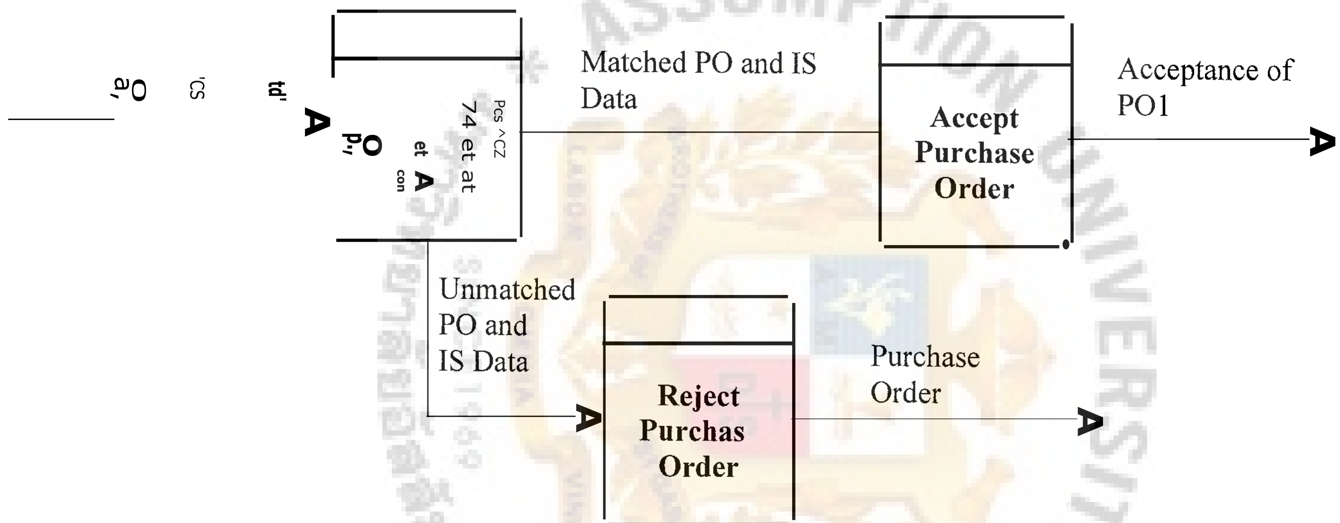
Process 10.0, Get Valid Approval of Tax Invoice Data, this process gets Approval of Tax Invoice from the customer and produces Valid Approval of Tax Invoice Data for the next process. This process also produces Valid Approval of Tax Invoice Data to store in the Valid Approval of Tax Invoice File. The data flow diagram showing the decomposition of process 10.0 is show in Figure 3.12.

Process 11.0, Generate Receipt, this process is no decomposition. This process gets Valid Approval of Tax Invoice Data from the previous process and generates Receipt for the customer. This process also produces Receipt

Data to store in the Receipt File. The data flow diagram showing the decomposition of process 11.0 is show in Figure 3.13.

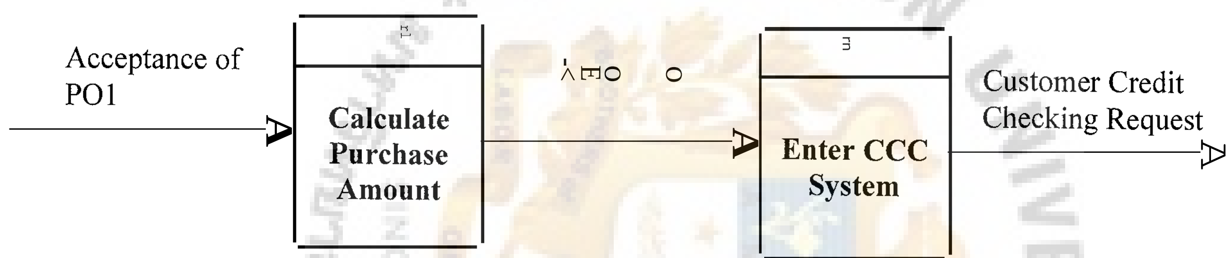


Data Flow Diagram Showing the Decomposition of Process 1.0.

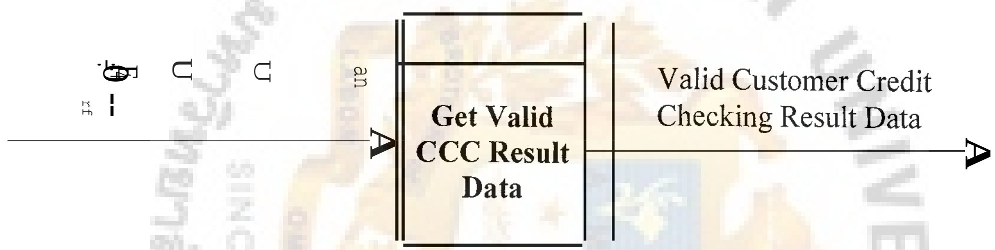


Data Flow Diagram Showing the Decomposition of Process 2.0.

ASSUMPTION UNIVERSITY OF THAILAND

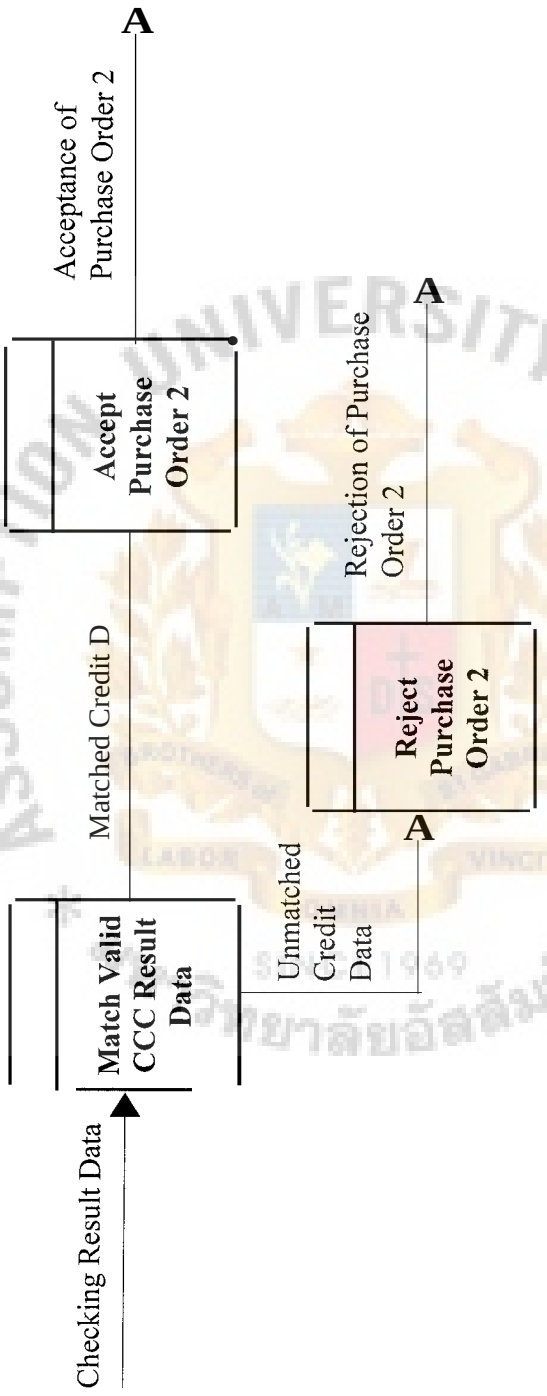


Data Flow Diagram Showing the Decomposition of Process 3.0.

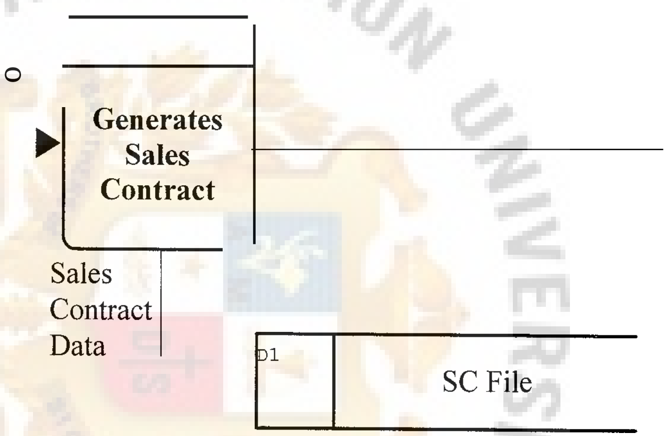


Data Flow Diagram Showing the Decomposition of Process 4.0.

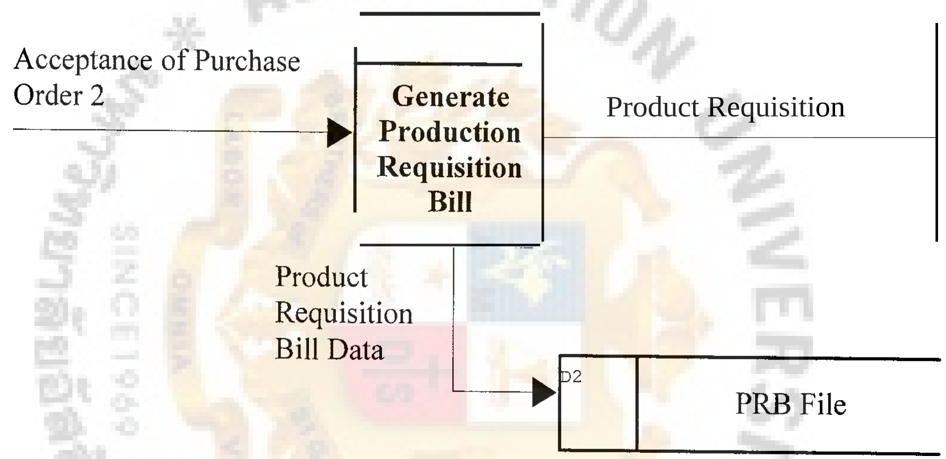
St. Gabriel's Library, **Au**

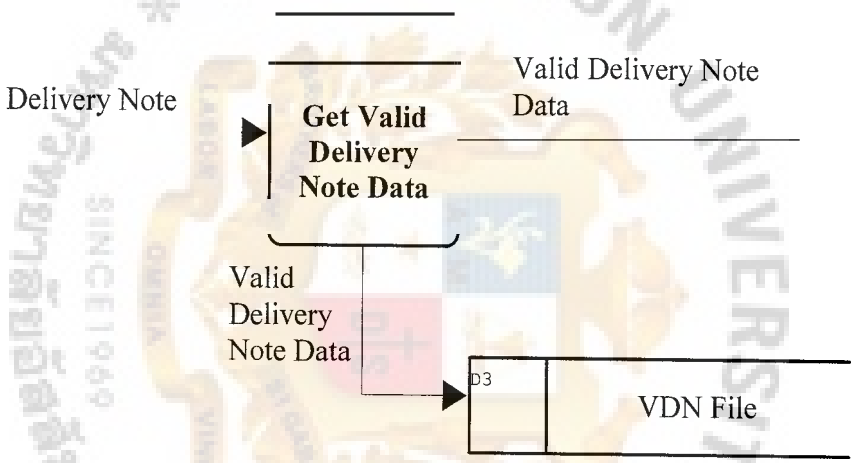


Data Flow Diagram Showing the Decomposition of Process 5.0.

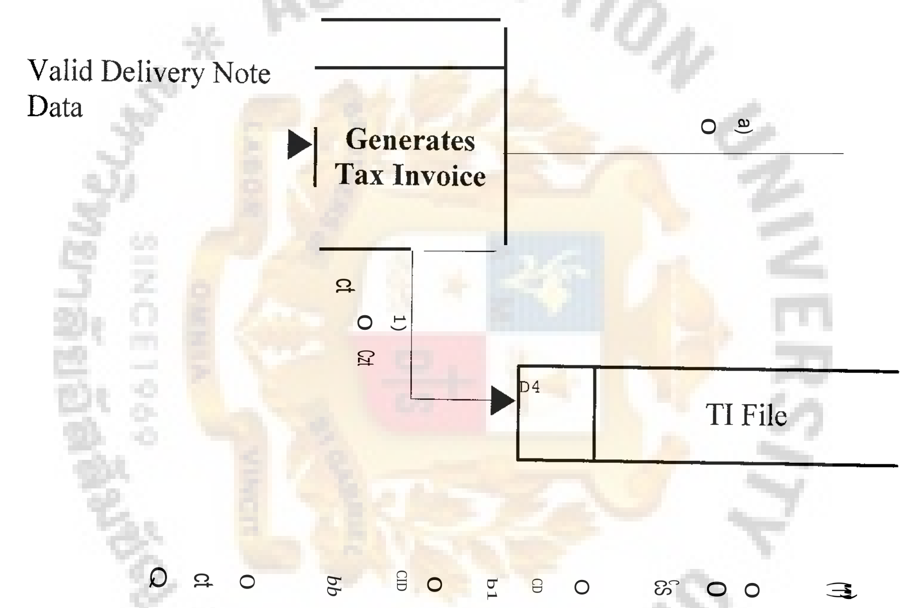


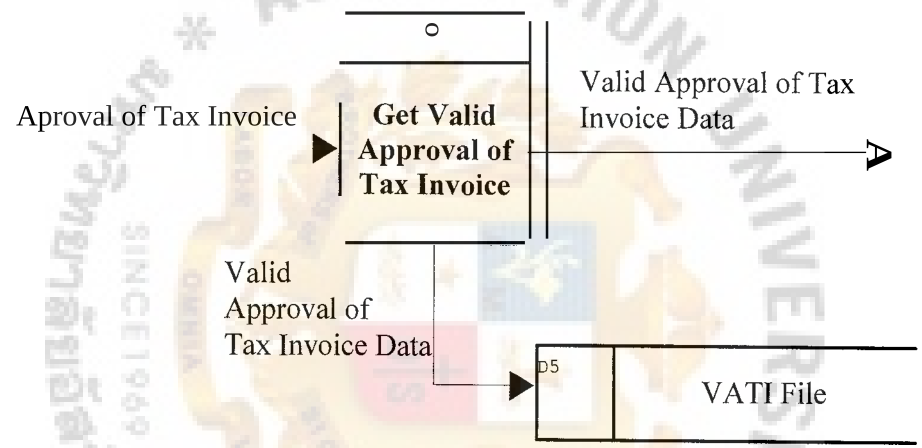
Data Flow Diagram Showing the Decomposition of Process 6.0.


Acceptance of Purchase Order 2
Generate Production Requisition Bill
Product Requisition
Product Requisition Bill Data
D2
PRB File

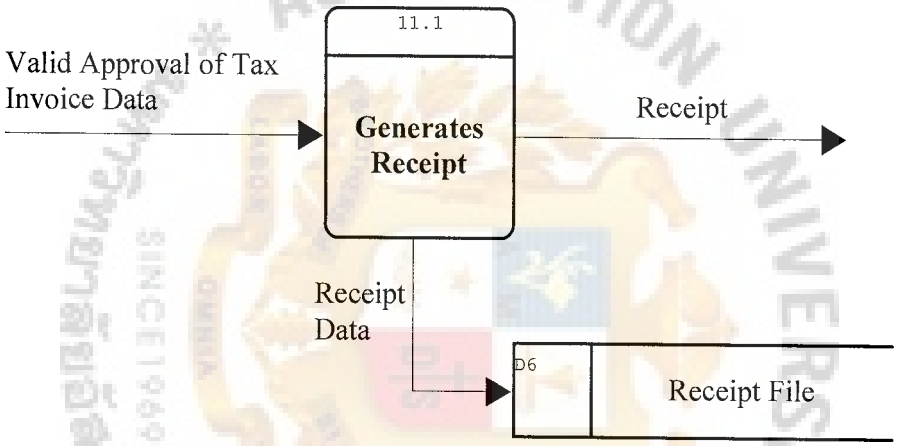


Data Flow Diagram Showing the Decomposition of Process 8.0.


Valid Delivery Note Data
Generates Tax Invoice
D4
TI File



bA O N cr; Da ta F low D iag ram Show ing t he Decomp os it ion o f Process 10. 0.



Flow Diagram Showing the Decomposition of Process 11.0.

3.2 Logic Modeling

3.2.1 Overview

Logic Modeling involves representing the internal structure functionality of the processes represented on data flow diagrams. The structure and functionality of a system's processes are a key element of any information system. Processes must be clearly described before they can be translated into a programming language. In the analysis phase, logic modeling will be complete and reasonably detailed, but it will also be generic in that it will not reflect the structure or syntax of a particular programming language. Modeling a system's logic is part of requirements structuring, just as was representing the system with data flow diagrams and the logic contained within it. Logic modeling also indicates when processes on a data flow diagram occur. Logic modeling not only represents the logic contained in a data flow diagram's processes, but also represents the contents and structure of a data flow diagram's data flow and data stores.

In structure analysis, the primary deliverables from logic modeling are structured descriptions and diagrams that outline the logic contained within each data flow diagram process as well as diagrams that show the temporal dimension of systems- when processes or events occur and how these events change the state of the system.

Creating diagrams and descriptions of process logic is not and end in itself. Rather, these diagrams and descriptions are created ultimately to serve as part of an unambiguous and thorough explanation of the system's specifications. These specifications are used to explain the system requirements to developers, whether people or code generators. Users, analysts, and programmers use logic diagrams and descriptions throughout analysis to incrementally specify a shared understanding of

requirements, without regard for programming languages or development environments.

3.2.2 Logic Modeling with Decision Tables

Several Methods can be used to represent the logic contained in an information system process, nut sometimes a process' logic can become quite complex. If several different conditions are involved, and combinations of these conditions dictate which if several actions should be taken, then other several methods may not be adequate for representing the logic behind such a complicated choice.

A Decision Table is a matrix representation of the logic of a decision, which specifies the possible conditions for the decision and the resulting actions. A decision table is a diagram of process logic where the logic is reasonably complicated. All of the possible choices and the conditions the choices depend on are represented in tabular form. There are three parts to the table: the condition stubs, the action stubs, and the rules. The condition stubs contain the various conditions that apply in the situation the table is modeling. The action stubs contain all the possible courses of action that result from combining values of the condition stubs. The part of the table that links conditions to actions is the section that contains the rules.

In constructing the decision tables, there is a set of procedures to be followed as:

(1) Name the conditions and the values each condition can assume.

Determine all of the conditions that are relevant to the problem, and then determine all of the values each condition can take. For some conditions, the values will be simply "yes" or "no".

(2) Name all possible actions that can occur.

The purpose of creating decision tables is to determine the proper course of action given a particular set of conditions.

(3) List all possible rules.

In first creating of decision able, it has to create an exhaustive set of rules. Every possible combination of conditions must be represented. It may turn out that some of the resulting rules are redundant or make no sense, but these determinations should be made only after it has listed every rule so that no possibility is overlooked. To determine the number of rules, multiply the number of values for each condition by the number of values for every other condition.

(4) Define the actions for each rule.

After all possible rules have been identified, provide an action for each rule.

(5) Simplify the decision table.

Make the decision table as simple as possible by removing any rules with impossible actions. Consult users on the rules where system actions are not clear and either decide on an action or remove the rule. Look for patterns in the number of rules, especially for indifferent conditions.

The decision table of the Inventory Checking System is depicted in Table 3.1. There are two condition stubs for this table: Purchase Order Information and Inventory Availability. Purchase Order Information has two values: "CWS", which stands for Conform With Standard, and "UCWS", which stands for Unconform With Standard. Inventory Availability has two values: "A", which stands for Available Inventory, and "U", which stands for Unavailable Inventory. There also are two possible courses of action in this table: Accept Purchase Order and Reject Purchase Order.

Table 3.1. Decision Table of Inventory Checking System.

	CONDITION/ COURSE OF ACTION	RULES			
		1	2	3	4
CONDITION STUBS	Purchase Order Information	CWS	UCWS	CWS	UCWS
	Inventory Availability	A	A	U	U
ACTION STUBS	Accept Purchase Order	X			
	Reject Purchase Order		X	X	X

To read the rules, start by reading the values of the conditions as specified in the first column: Purchase Order Information is "CWS", or Conform With Standard, and Inventory Availability is "A", or Available Inventory. When both of these conditions occur, the Inventory Checking System is Accept Purchase Order. In the second column, Purchase Order Information is "UCWS" and Inventory Availability is "A". The Inventory Checking System is Reject purchase Order. In the third column, Purchase Order Information is "CWS" and Inventory Availability is "U". The Inventory Checking System is Reject Purchase Order. In the forth column, Purchase Order Information is "UCWS" and Inventory Availability is "U". The Inventory Checking System is Reject Purchase Order.

Because of the indifferent condition for rules 2 and 4, it can reduce the number of rules by considering Rules 2 and 4 into one rule as shown in Table 3.1. The indifferent condition is represented with a dash. Whereas it started with a decision table with four rules, it now has a simpler table that conveys the same information with only three rules. The reduce decision table of inventory checking system is shown in Table 3.2.

Table 3.2. Reduce Decision Table of Inventory Checking System.

	CONDITION/ COURSE OF ACTION	RULES		
		1	2	3
CONDITION STUBS	Purchase Order Information	CWS	CWS	UCWS
	Inventory Availability	A	U	-
ACTION STUBS	Accept Purchase Order	X		
	Reject Purchase Order		X	X

The decision table of Customer Credit Checking System is depicted in Table 3.3. There are two condition stubs for this table: Customer Credit Outstanding and Amount Purchased. Customer Credit Outstanding has two values: "A", which stands for Available Credit, and "U", which stands for Unavailable Credit. Amount Purchased has two values: ">OB", which stands for More Than Outstanding Balance, and "<OB", which stands for Less Than Outstanding Balance. There also are two possible courses of action in this table: Accept Purchase Order and Reject Purchase Order.

Table 3.3. Decision Table of Customer Credit Checking System.

	CONDITION/ COURSE OF ACTION	RULES			
		1	2	3	4
CONDITION STUBS	Customer Credit Outstanding	A	U	A	U
	Amount Purchase	>OB	>OB	<OB	<OB
ACTION STUBS	Accept Purchase Order			X	
	Reject Purchase Order	X	X		X

To read the rules, start by reading the values of the conditions as specified in the first column: Customer Credit Outstanding is "A", or Available Credit, and Amount Purchased is ">OB", or More Than Outstanding Balance. When both of these conditions occur, the Customer Credit Checking System is Reject Purchase Order. In the second column, Customer Credit Outstanding is "U" and Amount Purchased is ">OB". The Customer Credit Checking System is Reject Purchase Order. In the third column, Customer Credit Outstanding is "A" and Amount Purchased is "<OB". The Customer Credit Checking System is Accept Purchase Order. In the forth column, Customer Credit Outstanding is "U" and Amount Purchased is "<OB". The Customer Credit Checking System is Reject Purchase Order.

Because of the indifferent condition for rules 2 and 4, it can reduce the number of rules by considering Rules 2 and 4 into one rule as shown in Table 3.3. The indifferent condition is represented with a dash. Whereas it started with a decision table with four rules, it now has a simpler table that conveys the same information with only three rules. The reduce decision table of customer credit checking system id shown in Table 3.4.

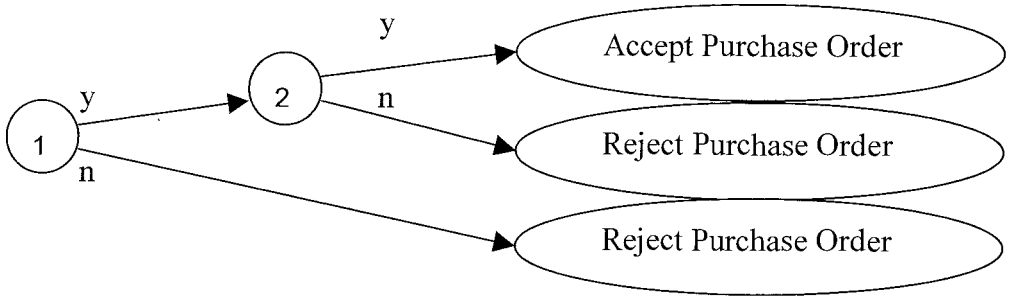
Table 3.4. Reduce Decision Table of Customer Credit Checking System.

	CONDITION/ COURSE OF ACTION	RULES		
		1	2	3
CONDITION STUBS	Purchase Order Information	A	A	U
	Inventory Availability	>OB	<OB	-
ACTION STUBS	Accept Purchase Order		X	
	Reject Purchase Order	X		X

3.2.3 Logic Modeling with Decision Trees

A decision tree is a graphical technique that depicts a decision or choice situation as a connected series of nodes and branches. Decision trees were first devised as a management science technique to simplify a choice where some of the needed information is not known for certain. By relying on the probabilities of certain events, a management scientist can use a decision tree to choose the best course of action. Therefore, the decision tree is a graphical representation of a decision situation in which decision points (nodes) are connected together by arcs (one for each alternative on a decision) and terminate in ovals (the action which is the result of all of the decisions made on the path that leads to that oval).

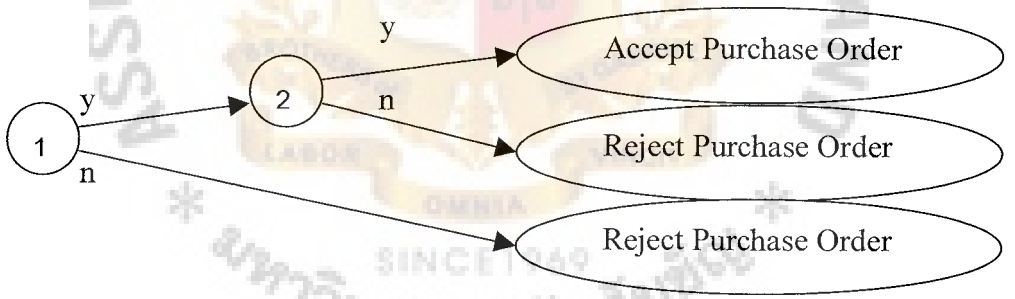
As used in requirements structuring, decision trees have two main components: decision points, which are represented by nodes, and actions, which are represented by ovals. To read decision tree, it begins at the root node on the far left. Each node is numbered, and each number corresponds to a choice; the choices are spelled out in a legend for the diagram. Each path leaving a node corresponds to one of the options for that choice. From each node, there are at least two paths that lead to the next step, which is either another decision point or an action. Finally, all possible actions are listed on the far right of the diagram in leaf nodes. Each rule is represented by tracing a series of paths from the root node, down a path to the next node, and so on, until an action oval is reached. The decision tree representation of decision logic in inventory checking system is shown in Figure 3.14. The decision tree representation of decision logic in customer credit checking system is shown in Figure 3.15.



Legend: (1) Information Conformation?

(2) Availability of Inventory?

Figure 3.14. Decision Tree Representation of Decision Logic in Inventory Checking System.



Legend: (1) Information Conformation?

(2) Availability of Inventory?

Figure 3.15. Decision Tree Representation of Decision Logic in Customer Credit Checking System.

3.3 **Conceptual Data Modeling**

3.3.1 Overview

There are many reasons to support the important of a conceptual data modeling as a part of the structuring system requirements. First, the characteristics of the data captured during data modeling are crucial in the design of databases, programs, computer screens, and printed reports. For example, facts such as these-a data element is numeric, a product can be in only one product line at a time, a line item on a customer order can never be moved to another customer order, customer region name is limited to a specified set of values-are all essential pieces of information in ensuring data integrity in an information system.

Second, data rather that processes are the most complex aspects of many modern information systems and hence require a central role in structuring system requirement. Transaction processing systems can have considerable process complexity in validating data, reconciling errors, and coordinating the movement of data to various databases. These types of system have been in place for years in most organizations; current system development focuses more on management information systems, decision support systems, and executive support systems. MIS, DSS, and ESS are more data-intensive and require extracting data from various data sources. The exact nature of processing is also more ad hoc that with transaction processing systems, so the details of processing steps cannot be anticipated. Thus, the goal is to provide a rich data resource that might support any type of information inguiry, analysis, and summarization.

Third, the characteristics about data are reasonably permanent. In contrast, the paths of data flow are quite dynamic. Who receives which data, the format of reports, and what reports are used change considerably and constantly overtime. A data model

explains the inherent nature of the organization, not its transient form. So, an information system design based on a data orientation, rather that a process or logic orientation, should have a longer useful life.

Finally, structural information about data is essential for automatic generation of programs, For example, the fact that a customer order has many line items on it instead of just one line item affects the automatic design of a computer screen for entry of customer orders. Thus, although a data model specifically documents the file and database requirements for an information system, the business meaning, or semantics, of data included in the data model have broader impact on the design and construction of a system.

The most common format used for data modeling is entity-relationship (E-R) diagramming. A similar format used with object-oriented analysis. Data modeling using the E-R notation explains the characteristics and structure of data independent of how the data may be stored in computer memories. A data modeling using the E-R notation is usually developed iteratively. During requirements structuring, an E-R model represents conceptual data requirements for a particular system. Then, after system inputs and outputs are fully described during logic design, the conceptual E-R data model is refined before it is translated into a logical format from which database definition and physical database design are done.

3.3.2 The Process of Conceptual Data Modeling

A conceptual data model is a representation of organizational data. The purpose of a conceptual data model is to show as many rules about the meaning and inter-relationships among data as are possible. The process is typically began by developing a conceptual data model for the system being replaced, if a system exists. This is essential for planning the conversion of the current files or database into the database of

the new system. A new conceptual data model will include all of the data requirements for the new system. Conceptual data modeling is one kind of data modeling and database design carried throughout the system development process. E-R diagramming is suited for this since E-R diagrams can be translated into a wide variety of technical architectures for data, such as relational, network, and hierarchical. An E-R data model evolves from project identification and selection through analysis as it becomes more specific and is validated by more detailed analysis of system needs.

The basic entity-relationship modeling notation uses three main constructs: data entities, relations, and their associated attributes. The E-R model notation has subsequently been extended to include additional constructs. The E-R model is expressed in terms of entities in the business environment, the relationships or associations among those entities, and the attributes or properties of both the entities and their relationships. An E-R model is noimally expressed as an entity-relationship diagram, which is a graphical representation of an E-R model.

3.3.3 Entity-Relationship Notation

(1) Entities

An entity is a person, place, object, event, or concept in the user environment about which the organization wishes to maintain data. An entity has its own identity which distinguishes it from each other entity. There is an important distinction between entity types and entity instances. An entity type is a collection of entities that share common properties or characteristics. Each entity type in an E-R model is given a name. Since the name represents a class or set, it is singular. Also, since an entity is an object, it use a simple noun to name an entity type. An entity instance is a single occurrence of an entity type. An entity type is described just once in a

St. Gabriel's Library, Au

data model while many instances of that entity type may be represented by data stored in the database.

(2) Attributes

Each entity type has a set of attributes associated with it. An attribute is a property or characteristic of an entity that is of interest to the organization. It uses an initial capital letter, followed by lowercase letters, and nouns in naming an attribute. In E-R diagrams, it can visually represent an attribute by placing its name in an ellipse with a line connecting it to the associated entity.

(3) Candidate Keys and Identifiers

Every entity type must have an attribute or set of attributes that distinguishes one instance from other instances of the same type. A candidate key is an attribute that uniquely identifies each instance of an entity type. A candidate key for a CUSTOMER entity type might be Customer ID.

(4) Multivalued Attributes

A multivalued attribute may take on more that one value for each entity instance. Suppose that Profile is one of the attributes of CUSTOMER. If each customer can have more that one profile, Profile is a multivalued attribute. During conceptual design, it is common to use a special symbol or notation to highlight multivalued attributes.

(5) Relationships

A relationship is the glue that hold together the various components of an entity-relationship model. A relationship is an association between the instances of one or more entity types that is of interest to the organization.

An association usually means that event has occurred or that there exists some natural linkage between entity instances. For this reason, relationships are labeled with verb phrases.

3.3.4 Business Rules

Conceptual data modeling is a step-by-step process for documenting information requirements that is concerned with both the structure of data and with rules about the integrity of that data. Business rules are specifications that preserve the integrity of the logical data model. There are four basic types of business rules:

(1) Entity Integrity

Each instance of an entity type must have a unique identifier that is not null.

(2) Referential Integrity Constraints

Rules concerning the relationships between entity types.

(3) Domians

Constraints on valid values for attributes. A Domain is the set of all data types and values that an attribute can assume.

(4) Triggering Operations

Other business rules that protect the validity of attribute. A triggering operation is an assertion or rule that governs that validity of data manipulation operations such as insert, update, and delete. The scope of triggering operations may be limited to attributes within one entity, or it may extend to attributes in two or more entities. Complex business rules may often be stated as triggering operations.

3.3.5 Conceptual Data Modeling and the Entity-Relationship Model

The goal of conceptual data modeling is to capture as much of the meaning of data as is possible. The more details about data in the model, the better the system can be designed and built. The advanced concepts needed in modeling data in the entity-relationship diagram are:

(1) Degree of a Relationship

The degree of a relationship is the number of entity types that participate in that relationship. The three most common relationships in entity-relationship model are unary, binary, and ternary. A unary relationship is a relationship between the instances of one entity type. A binary relationship is a relationship between instances of two entity types and is the most common type of relationship encountered in data modeling. A ternary relationship is a simultaneous relationship among instances of three entity types.

(2) Cardinalities in Relationships

The cardinality of a relationship is the number of instances of entity B that can be associated with each instance of entity A. The minimum cardinality of a relationship is the minimum number of instances of entity B that may be associated with each instance of entity A. The maximum cardinality is the maximum number of instances.

(3) Associative Entities

An Associative entity is a relationship that the data modeler choose to model as an entity type. The diamond symbol is included within the entity rectangle as a reminder that the entity was derived from a relationship. One situation in which a relationship must be turned into an associative entity is

when the associative entity has other relationships with entities besides the relationship which caused its creation.

The preliminary entity-relationship diagram of Sales Ordering System is depicted in the Figure 3.16. The purpose of this system is to process sales ordering transactions of the Sales and Marketing department. The transactions begin with the customer placing order. From the entity-relationship diagram, one customer can place one or more of purchase order. And also, one or more of purchase orders can be ordered by a customer. For zero or many of orders can have one or many of products, the product represents the item of product of the company. And one or many of products can be included in the zero or many of order. For each product that ordered by the customer, it needs to fulfill by one or no of inventory item, or zero or one of inventory item can fulfill each one of product in the order. Inventory item represents the inventory stock in the-stock-warehouse of the company. Next, an invoice item represents the amount of product delivered to the customer. Each one of tax invoice item must be fulfilled by each one of inventory item. That means the item sent to the customer must be fulfilled by each inventory in the stock. Invoice represents the document billing to the customer. In each one of invoice should has one or many invoice items. Finally, each customer can has one or more of invoice.

The Final entity-relationship diagram of Sales Ordering System is depicted in Figure 3.17. This entity-relationship diagram is a full entity-relationship data model with attributes.

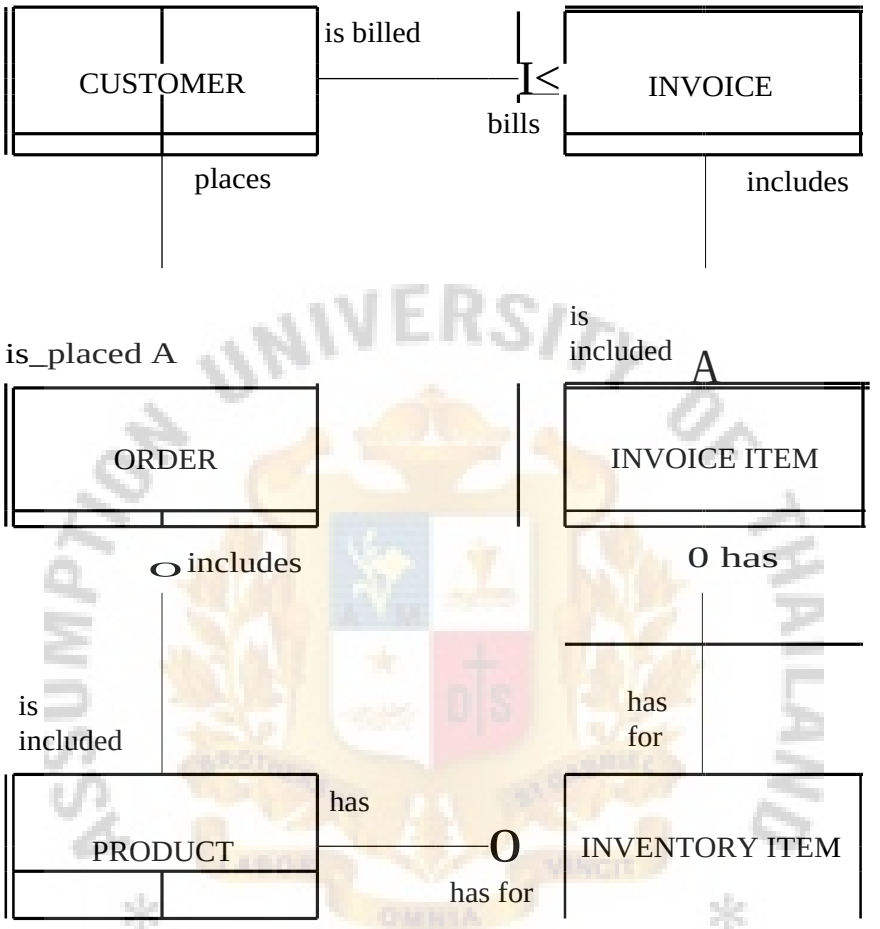
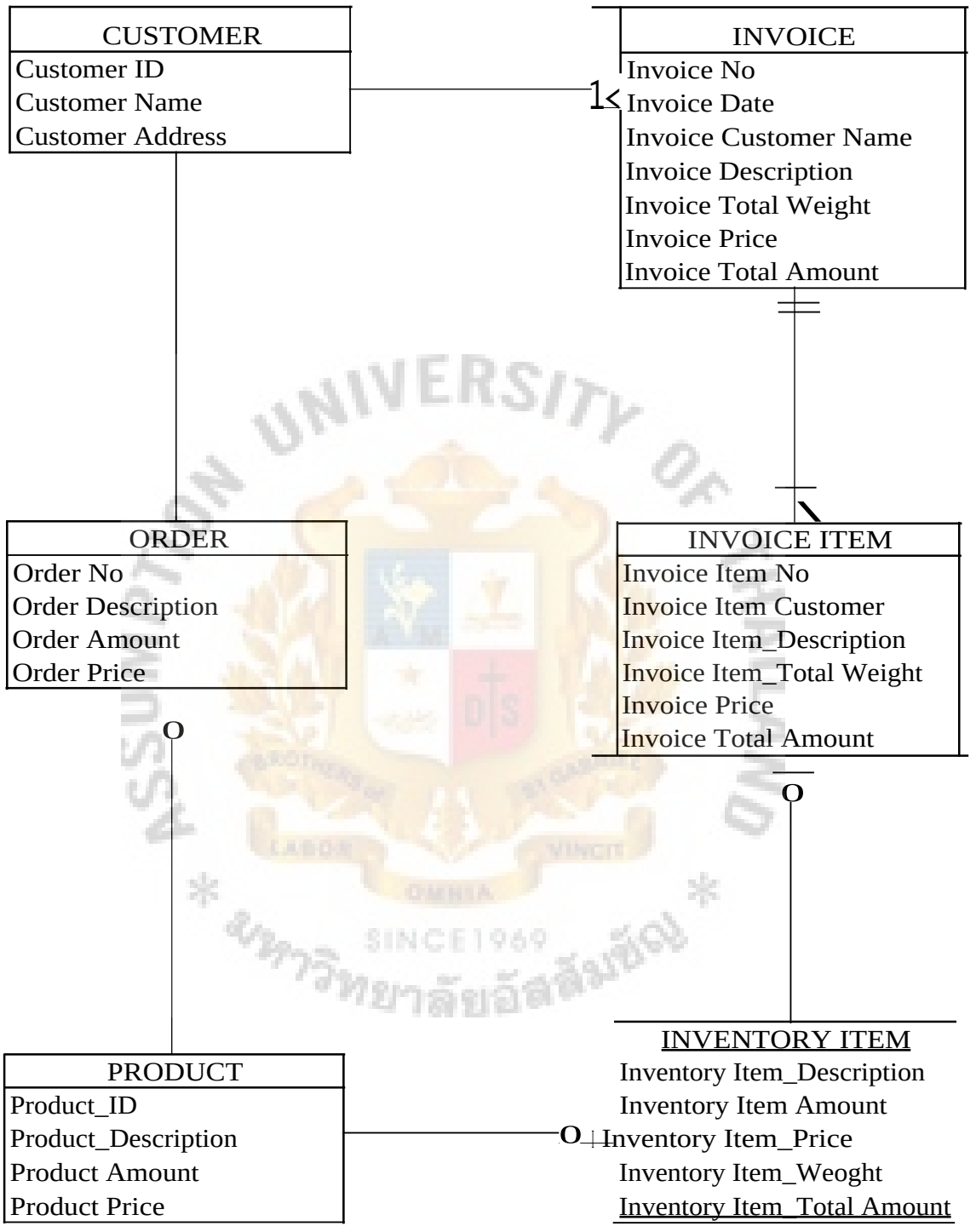


Figure 3.16. Preliminary Entity-Relationship Diagram for Sales Ordering System.



Figrue 3.17. Final Entity-Relationship for Sales Ordering System.

3.4 Forms and Reports Designing

3.4.1 Overview

System inputs and outputs-forms and reports-were identified during requirements structuring. The kinds of forms and reports the system will handle were established as part of the design strategy formed at the end of the analysis phase of the system development process. During analysis, it may not have been concerned with the precise appearance of forms and reports, only with which ones needed to exist and what their contents were. It may have distributed prototypes of forms and reports, which emerged during analysis as a way to confirm requirements with users. Forms and reports are integrally related to various diagrams developed during requirements structuring. For example, every input form will be associated with a data flow entering a process on a data flow diagram, and every output form or report will be a data flow produced by a process on a data flow diagram. This means that the contents of a form or report correspond to the data elements contained in the associated data flow. Further, the data on all forms and reports must consist of data elements in data stores and on the entity-relationship data model for the application, or must be computed from these data elements.

A form is a business document containing some predefined data and often includes some areas where additional data are to be filled in. Most forms have a stylized format and are usually not in a simple row and column foiniat. Example of business forms is product order forms, employment applications, and class registration sheets. Traditionally, forms have been displayed on a paper medium, but today video display technology allow to duplicate the layout of almost any printed form, including an organizational logo or any graphic, on a video display terminal.

A report is a business document containing only predefined data; it is a passive document used solely for reading or viewing. Examples of reports are invoices, weekly sales summaries by region and salesperson, and a picture chart of population by age categories. Often a report has rows and columns of data, but a report may consist of any format-for example, mailing labels. Frequently, the differences between a form and a report are subtle. A report is only for reading and often contains data about multiple unrelated records in a computer file. On the other hand, a form typically contains data from only one record or is, at least, based on one record, such as data about one customer, one order, or one file. The guidelines for the design of foinis and reports are very similar.

3.4.2 The Process of Designing Forms and Reports

Designing forms and reports is a user-focused activity that typically follows a prototyping approach. It must gains an understanding of the intended user and task objectives by collecting initial requirements during requirement determination. During this process, several questions must be answered. These questions attempt to answer the "who, what, when, where, and how" related to the creation of all forms and reports. Gaining an understanding of these questions is a required first step in the creation of any form or report.

After collecting initial requirements, it should structure and refine this information into an initial prototype. Structuring and refining requirements are completed independently of the users, although it may need to occasionally contact users in order to clarify some issue overlooked during analysis. Finally, the users needed to ask and review the prototype. After reviewing the prototype, users may accept the design or request that changes are made. Usually, several iteration of reviews occur during the design of a single form or report.

In designing forms and reports, design specifications are the major deliverables and are inputs to the system implementation phase. Design specifications have three sections as narrative overview, sample design, and testing and usability assessment. The first section of a design specification contains a general overview of the characteristics of the target users, tasks, system, and environmental factors in which the form or report will be used. The purpose is to explain to those who will actually develop the final form why this form exists and how it will be used so that they can make the appropriate implementation decisions.

In the second section of the specification, a sample design of form is shown. This design may be hand-drawn using a coding sheet. Using actual development tools allow the design to be more thoroughly tested and assessed. The final section of the specification provides all tests and usability assessment information.

3.4.3 Formatting Forms and Reports

Within the context of building information systems, there are two general types of information: internal information and external information. Internal information refers to information that is collected, generated, or consumed within an organization. Internal information can also consist of simple information reports summarizing daily activities, such as sales contract report that a salesperson might produce for the manager or the forms used to collect daily inventory transactions. External information refers to information that is collected from or created for customers, suppliers, or the departments. Like internal information, external information can be delivered in a variety of formats and media.

A hybrid type of information that is both internal and external is called a turnaround document. A turnaround document is produced by a system as an output that may also turn around and return to the organization as an input. Some turnaround

documents within high-volume transaction-based systems can contain computer-readable information. Different methods of producing standard forms and reports will influence the cost of producing information. The variety of media can be used to deliver system information.

On-line processing refers to the collection and delivery of the most recent available information, typically through an on-line workstation. On-line processing is preferred under the following conditions:

(1) Access to or capture of the information occurs randomly.

(2) The format and type of information are not consistent.

(3) Information is continuously changing and the most current information is needed for proper processing and decision-making.

(4) Users are in locations where there is easy access to an information system or a remotely connected terminal.

Batch processing, on the other hand, refers to the input or output of system information on a predetermined and specific time interval. This means that information collected or generated through batch processing has a shelf of a limited time period whereas on-line processing always represents the most recent available information. However, batch processing can be collected and delivered both on-line or on hard copy. For example, when viewing an on-line batch processing report, users would access a static textual or graphical report from a terminal or workstation. The display is on-line, but the data are not current. Alternatively, a hard copy batch processing report would simply be a system output produced on some type of media such as a printer. For batch output, it is important to know both the date the output is produced as well as when the data were current. Batch processing is preferred under the following conditions:

(1) Access to infoi_iation occurs regularly.

(2) The format and type of information are consistent.

(3) Information is stable for some time period and/or the users' informational needs do not require the most up-to-date information for effective decision making or task performance.

(4) Users are in locations where access to an on-line information system or remotely connected terminal is not possible or extremely difficult.

3.4.4 General Formatting Guidelines

In formatting forms and reports, the meaningful titles should; clear and specific titles describing content and use of form or report, revision date or code to distinguish a form or report from prior version, current date which identifies when the form or report were accurate. The meaningful information should; display only needed information, be provided in a manner that is usable without modification. In the balanced layout, information should be balanced on the screen or page, adequate spacing and margins should be used, all data and entry field should be clearly labeled. For the easy navigation, it should; clearly show how to move forward and backward, clearly show where the positions are, notify user when on the last page of a multipaged sequence.

3.4.5 Highlighting Information

As display technologies improve, there will be a greater variety of methods available for highlighting information. A list of the most commonly used methods for highlighting information are blinking and audible tones, color differences, intensity differences, size differences, font differences, reverse video, boxing underlining all capital letters, and offsetting the position of nonstandard information. It is more important than ever to consider how highlighting can be use to enhance an output and not prove a distraction. In general, highlighting should be used sparingly to draw the user to or away from certain information and to group together related information.

There are several situations when highlighting can be a valuable technique for conveying special information; notify users of errors in data entry or processing, providing warnings to users regarding possible problems such as unusual data values or an unavailable device, and drawing attention to keywords, commands, high priority messages, and data that have changes or gone outside normal operating ranges.

Additionally, many highlighting techniques can be used singularly or in tandem, depending upon the level of emphasis desired by the designer, and the highlighting should be used conservatively and consistently.

3.4.6 Color of Forms and Reports

Color is a powerful tool for designing the usability of a system. When applied appropriately, color provides many potential benefits to forms and reports. The use of color had positive effects on user task performance and perceptions when the user was under time constraints for completion of a task. Color was also beneficial for gaining greater understanding from a display or chart. The benefits of color seem to apply if the information is first provided to the user in the most appropriate presentation format. That is if information is most effectively displayed in a bar chart, color can be used to enhance or supplement the display. If information is displayed in an inappropriate format, color has little or no effect on improving understanding or task performance.

There are also several problems associated with using color. Most of these dangers are related more to the technical capabilities of the display and hard copy devices than misuse. For the benefits of using color, there are; soothes or strikes the eye, accents an uninteresting display, facilitates subtle discriminations in complex displays, emphasizes the logical organization of information, draws attention to warnings, and evokes more emotional reactions. For the disadvantage in using color in the designing forms and reports are color parings may wash out or cause problems for

some users, resolution may degrade with different displays, color fidelity may degrade on different displays, and printing or conversion to other media may not easily translate.

3.4.7 Designing Tables and Lists

Unlike textual information, where context and meaning are significantly derived through reading, the context and meaning of tables and lists are significantly derived form the format of the information. Consequently, the usability of information displayed in tables and alphanumeric lists is likely to be much more influenced by effective layout than most other types of information display. For the general guidelines for displaying tables and lists, the use of meaningful labels are all columns and rows should have meaningful labels, labels should be separated from other information by using highlighting, and re-display labels when the data extend beyond a single screen or page. The formatting of columns, rows, and text are sorting in a meaningful order, placing a blank line between every five rows in long columns, columns should have at least two spaces between them, allowing white space on printed reports for user to write note, using a single typefaces within and cross displays and reports, and avoiding overly fancy fonts. The formatting of numeric, textual, and alphanumeric data should right justify numeric data and align columns by decimal points or other delimiter, left-justify textual data, and break long sequences of alphanumeric data into small groups of three to four characteristics each. The product requisition bill form is shown in Figure 3.18. The sales contract form is shown in Figure 3.19. The tax invoice form is shown in Figure 3.20. The receipt form is shown in Figure 3.21. The purchase order report is shown in Figure 3.22. The product requisition bill report is shown in Figure 3.23. The sales contract report is shown in Figure 3.24. The delivery note report is shown in

Figure 3.25. The tax invoice report is shown in Figure 3.26. The receipt report is shown in Figure 3.27.

ASSUMPTION UNIVERSITY OF THAILAND
SINCE 1969

Sahaviriya Panich Company Limited

33/1 Rama 3 rd. Yannawa, Bangkok 10120

Tel: 2853101-10 Fax: 285-3197, 285-3198

Product Requisition Bill

Product Requisition Bill No: 0658/2002

Date: $0^{8}/_{1}0/02$

Sales Contract No: 0310/2002

Customer ID: 089 Payment Condition: 60 days

Customer Name: Kim Heng Seng

Customer Address: 95/265 Rama 3 Rd. Yannawa, Bangkok 10120

Delivery Place: 95/265 Rama 3 Rd. Yannawa, Bangkok 10120

Item:	ID:	Description:	Weight:	Amount:	Price(bht.):
1	C-042306	C-Channel 100x50x20x2.3mmx6m	23 kgs.	1,000	15.00
		Total:		1,000	pieces

Approval by :

Sales Section Manager Sales Dep. Manager

Figure 3.18. Product Requisition Bill Form of the System.

Sahaviriya Panich Company Limited

33/1 Rama 3 rd. Yannawa, Bangkok 10120

Tel: 2853101-10 Fax: 285-3197, 285-3198

Sales Contract

Sales Contract No: 310

Date: 08/10/02

Purchase Order No: 013

We pleased to confirm you the order. While thanking you for the confidence thet you are demonstrating to our company, we kindly ask you to return us a signed copy of this order confirmation to acknowledge your acceptance of the coditions therein and we remain.

Buyer ID: 089 Payment Condition: 60 days

Buyer Kim Heng Seng

95/265 Rama 3 Rd. Yannawa, Bangkok 10120

Delivery Place: 95/265 Rama 3 Rd. Yannawa, Bangkok 10120

Item:	ID:	Description:	Weight:	Amount:	Price(bht.):
1	C-042306	C-Channel 100x50x20x2.3mmx6m	23 kgs.	1,000	15.00
		Total:		1,000	pieces

Approval by :

The Buyer (for acceptance)

Sahaviriya Panich Compnay

Sales Dep. Manager

Figure 3.19. Sales Contract Form of the System.

Sahaviriya Panich Company Limited
33/1 Rama 3 rd. Yannawa, Bangkok 10120
Tel: 2853101-10 Fax: 285-3197, 285-3198

Tax Invoice

Buyer: Him Heng Seng Co., Ltd. 95/265 Rama 3 Rd. Yannawa, Bangkok 10120 Thailand	Commercial Invoice No: 0711/2002	Date: 10/10/02
	Country of Origin: Thailand	Country Of Destination: Thailand
Delivery Note No: 0550/2002	Sales Contract No: 0310/2002	Date: 11/10/02
Shipped per: Truck BTC (KHSS) From: Yannawa, Bangkok Thailand To: Bangkok	Term of Shipment: By cheque credit at 60 days after invoice date	

Description of goods: Light Lips Channel

Weight(kgs.)	Amount(pcs.)	Size	Prices(Bht./kg.)	Total Amount(Bht.)
23,000	1,000	100x50x20x2.3mmx6m	15.00	345,000
23,000				345,000

Total	345,000
Vat7%	24,150
Total Amount	369,150

Sahaviriya Panich Company

Sales Department

Figure 3.20. Tax Invoice Form of the System.

Sahaviriya Panich Company Limited
33/1 Rama 3 rd. Yannawa, Bangkok 10120
Tel: 2853101-10 Fax: 285-3197, 285-3198

Receipt

Receipt No: 0425/2002
Date: 12/10/02

Customer: Him Heng Seng Co., Ltd.
95/265 Rama 3 Rd.
Yannawa, Bangkok 10120
Thailand
Remark

Item:	Tax Invoice No:	Date: Amount:	Vat7%	Total Amount(Bht.):
1	0711/2002	12/010/02 345,000	24,150	369,150

Grand Total: 369,150

Approval by:

Customer Seller

Figure 3.21. Receipt Form of the System.

Sahaviriya Panich Co., Ltd.
33/1 Rama 3 Rd. Yannawa, Bangkok 10120
Tel: (02)2853101-10 Fax: (02)285-3197-8

Report: Purchase Order Report
Date of Prepared: 13/10/02
Prepared by: 001
Section: 1-3
Prepared to: Manager

Item:	PO No.	Date:	Customer	Order Amount (kgs.)	Status
1	013	08/10/02	Kim Heng Seng Co., Ltd.	23,000	shipped
2	011	08/10/03	C.H.S. Steel Co., Ltd.	15,000	shipped
3	258	08/10/04	S.J.S. Steel Co., Ltd.	15,000	shipped
4	362	08/10/05	Pisit Metal Co., Ltd.	25,000	shipped
5	998	08/10/06	Thai Metal Import Co., Ltd.	30,000	shipped
6	100	08/10/07	Lohapaisarn Wamich Co., Ltd.	28,000	shipped
7	110	08/10/08	Metalic Coil Center Co., Ltd.	25,000	shipped
8	856	08/10/09	Rocket Steel Co., Ltd.	15,000	shipped
9	256	08/10/10	Sin Siam Steel Co., Ltd.	20,000	shipped
10	778	08/10/11	Bangkok Steel Co., Ltd.	16,000	shipped
11	963	08/10/12	B.S.Metal Co., Ltd.	20,000	shipped
12	090	08/10/13	S.S.V. Steel Co., Ltd.	25,000	shipped
			Total (kgs.)	257,000	

Figure 3.22. Purchase Order Report of the System.

Sahaviriya Panich Co., Ltd.
33/1 Rama 3 Rd. Yannawa, Bangkok 10120
Tel: (02)2853101-10 Fax: (02)285-3197-8

Report: Product Requisition Bill Report
Date of Prepared: 13/10/02
Prepared by: 001
Section: 1-3
Prepared to: Manager

Item:	PRB No:	Date:	Description	Amount (kgs.)	Status
1	658	08/10/02	C-Channel 100x50x20x2.3mmx6x	23,000	shipped
2	659	08/10/02	C-Channel 100x50x20x2.3mmx6x	15,000	shipped
3	660	08/10/02	C-Channel 15 Ox5 Ox20x3.2mmx6x	15,000	shipped
4	661	08/10/02	C-Channel 75x45x15x2.3mmx6x	25,000	shipped
5	662	08/10/02	Hot Rolled Sheet 2.0mmx4'x8'	30,000	shipped
6	663	08/10/02	C-Channel 75x45x15x2.3mmx6x	28,000	shipped
7	664	08/10/02	C-Channel 75x45x15x3.2mmx6x	25,000	shipped
8	665	08/10/02	C-Channel 75x45x15x3.2mmx6x	15,000	shipped
9	666	08/10/02	Hot Rolled Sheet 2.0mmx4'x8'	20,000	shipped
10	667	08/10/02	C-Channel 125x50x20x3.2mmx6x	16,000	shipped
11	668	08/10/02	C-Channel 125x50x20x3.2mmx6x	20,000	shipped
12	669	08/10/02	C-Channel 100x50x20x2.3mmx6x	25,000	shipped
			Total (kgs.)	257,000	

Figure 3.23. Product Requisition Bill Report of the System.

Sahaviriya Panich Co., Ltd.
33/1 Rama 3 Rd. Yannawa, Bangkok 10120
Tel: (02)2853101-10 Fax: (02)285-3197-8

Report: Sales Contract Report
Date of Prepared: 13/10/02
Prepared by: 001
Section: 1-3
Prepared to: Manager

Item:	SC No:	Date:	Description	Amount (kgs.)	Amount (Bht.)
1	299	06/10/02	C-Channel 100x50x20x2.3mmx6x	25,000	453,000
2	300	06/10/02	C-Channel 100x50x20x2.3mmx6x	19,000	232,000
3	301	06/10/02	C-Channel 15 Ox50x20x3 .2mmx6x	19,500	265,000
4	302	06/10/02	C-Channe1 75x45x15x2.3mmx6x	21,000	298,000
5	303	07/10/02	C-Channel 75x45x15x2.3mmx6x	25,000	190,000
6	304	07/10/02	C-Channel 75x45x15x2.3mmx6x	28,000	185,000
7	305	07/10/02	C-Channel 75x45x15 x3 .2mmx6x	25,000	210,000
8	306	07/10/02	C-Channel 75x45x15x3.2mmx6x	29,000	213,000
9	307	08/10/02	C-Channel 150x50x20x2.3mmx6x	30,000	300,000
10	308	08/10/02	C-Channel 125x50x20x3.2mmx6x	25,000	235,000
11	309	08/10/02	C-Channel 125x50x20x3.2mmx6x	20,000	400,000
12	310	08/10/02	C-Channel 100x50x20x2.3mmx6x	23,000	345,000
			Total:	289,500	3,326,000

Figure 3.24. Sales Contract Report of the System.

Sahaviriya Panich Co., Ltd.
33/1 Rama 3 Rd. Yannawa, Bangkok 10120
Tel: (02)2853101-10 Fax: (02)285-3197-8

Report: Delivery Note Report
Date of Prepared: 13/10/02
Prepared by: 001
Section: 1-3
Prepared to: Manager

Item:	DN No:	Date:	Description	Amount (kgs.)	Amount (pcs.)
1	535	07/10/02	C-Channel 100x50x20x2.3mmx6x	25,220	1,000
2	538	07/10/02	C-Channel 100x50x20x2.3mmx6x	19,580	800
3	540	07/10/02	C-Channel 150x50x20x3.2mmx6x	19,500	850
4	542	07/10/02	C-Channel 75x45x15x2.3mmx6x	21,250	1,200
5	543	08/10/02	C-Channe175x45x15x2.3mmx6x	25,020	1,000
6	544	08/10/02	C-Channel 75x45x15x2.3mmx6x	27,590	950
7	545	08/10/02	C-Channel 75x45x15x3.2mmx6x	24,980	1,200
8	546	08/10/02	C-Channel 75x45x15x3.2mmx6x	28,560	1,100
9	547	10/10/02	C-Channel 150x50x20x2.3mmx6x	29,050	900
10	548	10/10/02	C-Channel 125x50x20x3.2mmx6x	24,500	1,000
11	549	10/10/02	C-Channel 125x50x20x3.2mmx6x	19,500	800
12	550	10/10/02	C-Channel 100x50x20x2.3mmx6x	23,000	1,000

Total: 287,750 11,800

Figure 3.25. Delivery Note Report of the System.

Sahaviriya Panich Co., Ltd.
33/1 Rama 3 Rd. Yannawa, Bangkok 10120
Tel: (02)2853101-10 Fax: (02)285-3197-8

Report: Tax Invoice Report
Date of Prepared: 13/10/02
Prepared by: 001
Section: 1-3
Prepared to: Manager

Item:	TI No:	Date:	Description	Amount (kgs.)	Amount (Bht.)
1	299	10/10/02	Kim Heng Seng Co., Ltd.	25,220	484,710
2	300	10/10/02	C.H.S. Steel Co., Ltd.	19,580	248,240
3	301	10/10/02	S.J.S. Steel Co., Ltd.	19,500	283,550
4	302	10/10/02	Pisit Metal Co., Ltd.	21,250	318,860
5	303	11/10/02	Thai Metal Import Co., Ltd.	25,020	203,300
6	304	11/10/02	Lohapaisarn Wamich Co., Ltd.	27,590	197,950
7	305	11/10/02	Metalic Coil Center Co., Ltd.	24,980	224,700
8	306	11/10/02	Rocket Steel Co., Ltd.	28,560	227,910
9	307	12/10/02	Sin Siam Steel Co., Ltd.	29,050	321,000
10	308	12/10/02	Bangkok Steel Co., Ltd.	24,500	251,450
11	309	12/10/02	B.S.Metal Co., Ltd.	19,500	428,000
12	310	12/10/02	S.S.V. Steel Co., Ltd.	23,000	369,150
			Total:	287,750	3,558,820

Figure 3.26. Tax Invoice Report of the System.

Sahaviriya Panich Co., Ltd.
33/1 Rama 3 Rd. Yannawa, Bangkok 10120
Tel: (02)2853101-10 Fax: (02)285-3197-8

Report: Receipt Report
Date of Prepared: $1^{3}/_{1}0/02$
Prepared by: 001
Section: 1-3
Prepared to: Manager

Item:	Rpt. No:	Date:	Description	TI date:	Amount (Bht.)
1	414	11/10/02	Kim Heng Seng Co., Ltd.	10/10/02	484,710
2	415	11/10/02	C.H.S. Steel Co., Ltd.	10/10/02	248,240
3	416	11/10/02	S.J.S. Steel Co., Ltd.	10/10/02	283,550
4	417	11/10/02	Pisit Metal Co., Ltd.	10/10/02	318,860
5	418	12/10/02	Thai Metal Import Co., Ltd.	11/10/02	203,300
6	419	12/10/02	Lohapaisarn Warnich Co., Ltd.	11/10/02	197,950
7	420	12/10/02	Metalic Coil Center Co., Ltd.	11/10/02	224,700
8	421	$1^{2}/_{1}0/02$	Rocket Steel Co., Ltd.	11/10/02	227,910
9	422	13/10/02	Sin Siam Steel Co., Ltd.	12/10/02	321,000
10	423	13/10/02	Bangkok Steel Co., Ltd.	12/10/02	251,450
11	424	13/10/02	B.S.Metal Co., Ltd.	12/10/02	428,000
12	425	13/10/02	S.S.V. Steel Co., Ltd.	12/10/02	369,150

Total: 3,558,820

Figure 3.27. Receipt Report of the System.

St. Gabriel's Library, Au

3.5 Interfaces and Dialogues Design

3.5.1 Overview

The process of designing interfaces and dialogues is a user-focused activity, by following a prototyping methodology of iteratively collecting information, constructing a prototype, assessing usability and making refinements. To design usable interfaces and dialogues, it needs to answer the questions who, what, when, where, and how guiding in the design of forms and reports. Thus, this process parallels that of designing forms and reports.

The deliverable and outcome from system interface and dialogue design is the creation of a design specification. This specification is also similar to the specification produced for form and report design. For the interface and dialogue design, one additional subsection is a section outlining the dialogue sequence-the way a user can move from one display to another.

Methods of Interaction

The human-computer interface defines the ways in which users interact with an information system. All human-computer interfaces must have an interaction style and use some hardware device for supporting this interaction. In designing the user interface, the most fundamental decision needed to make relates to the methods used to interact with the system.

Command language interaction is a human-computer interaction method where users enter explicit statements into a system to invoke operations. Command language interaction places a substantial burden on the user to remember names, syntax, and operations. Most newer or large-scale system no longer rely entire on a command language interface. Yet, command language are good for experienced users, for system with a limited command set, and for rapid interaction with the system.

Menu interaction is a human-computer interaction method where a list of system options is provided and a specific command is invoked by user selection of a menu option. Menus have become the widely used interface method because the user only needs to understand simple signposts and route options to effectively navigate through a system. Menus can differ significantly in their design and complexity. The variation of the design is most often related to the capability of the development environment, the skills of the developer, and the size and complexity of the system.

In small and less complex system with limited system options, it may use a single menu or a linear sequence of menus. A single menu has obvious advantages over a command language but may provide little guidance beyond invoking the command. For large and more complex systems, it can use menu hierarchies to provide navigation between menus. These hierarchies can be simple tree structures or variations wherein children menus have multiple parent menus or which allow multilevel traversal. Variations as to how menus are arranged can greatly influence the usability of a system. There are two common methods for positioning menus. In a pop-up menu, menus are displayed near the current cursor position so users do not have to move the position or their eyes to view system options. A pop-up menu had a variety of potential uses. One is to show a list of commands relevant to the current cursor position. Another is to provide a list of possible values to fill in for the current field. In a drop-down menu, menus drop down from the top line of the display. Drop-down menus have become very popular in recent years because they provide consistency in menu location and operation among applications and efficiently use display space.

There are some guidelines in designing menus. For the wording in menu design, each menu should have a meaningful title, command verbs should clearly and specifically describe operations, and menu items should be displayed in mixed upper-

and lower-case letters and have a clear, unambiguous interpretation. For the organization of menu design, a consistent organizing principle should be used that relates to the tasks the intended users perform. For the length of the menu design, the number of menu choices should not exceed the length of the screen, and submenus should be used to break up exceedingly long menus. For the selection of menu design, selection and entry methods should be consistent and reflect the size of the application and sophistication of the users. For the highlighting of the menu design, it should be minimized and used only to convey selected options or unavailable options.

Form interaction is to allow users to fill in the blanks when working with a system. Form interaction is effective for both the input and presentation of information. An effectively design form includes a self-explanatory title and field headings, had a field organized into logical groupings with distinctive boundaries, provides default values when practical, displays data in appropriate field lengths.

Object-based interaction is the use of the most common icon method in implementing object-based interaction. Icons are graphic symbols that look like the processing option they are meant to represent. Users select operations by pointing to the appropriate icon with some types of pointing devices. The primary advantages to icon are that they take up little screen space and can be quickly understood by most users. An icon may also look like a button which, when selected or depressed, causes the system to take an action relevant to that form, such as cancel, save, edit a record, or ask for help.

Natural language interaction refers to the interaction method which use a branch of artificial intelligence research studies techniques for allowing system to accept inputs and produce outputs in a conventional language like English.

3.5.2 Layouts Design

In order to ease user training and data recording, it use standard formats for computer-based forms and reports similar to paper-based forms and reports for recording or reporting information. For the most common forms, there are header information, sequence and time related information, instruction or formatting infonnation, body or data details, total or data summary, authorization or signature, and comments. In many organizations, data is often first recorded on paper-based forms and then later recorded within application systems. When designing layouts to record or display information on paper-based form, it should try to make both as similar as possible. Additionally, data entry displays should be consistently formatted across applications to speed data entry and reduce errors.

Another concern in designing the layout of computer-based forms is the design of between-field navigation. Since it can control the sequence for users to move between fields, standard screen navigation should flow from left to right and top to bottom just as working with the paper-based forms. Data fields should be grouped into logical categories with labels describing the contents of the category. Areas of the screen not used for data entry or commands should be inaccessible to the user.

In designing the navigation procedures within the system, flexibility and consistency are primary concerns. Users must be able freely move forward and backward or to any desired data entry fields. User can also be able to navigate each form in the same way or in as similar a manner as possible. Additionally, data are not usually be permanently saved by the system until the user makes an explicit request to do so. This allows the user to abandon a data entry screen, back up, or move forward without adversely impacting the contents of the permanent data.

Consistency extends to the selection of keys and commands. Each key or command has only one function and this function was consistent throughout the entire system and across the system.

3.5.3 Data Entry Structuring

There is a set of rules to be considered in structuring data entry. First, never require data that are already on-line or that can be computed; for example, do not enter customer data on an order form if that data can be retrieved from the database, and do not enter extended prices which can be computed from quantity sold and unit prices. Second, always provide default values when appropriate; for example, assume today's date for a new sales invoice, or use the standard product price unless overridden. Third, make clear the type of data units requested for entry; for example, indicate quantity in tons, dozens, pounds, etc. Forth, use character replacement when appropriate; for example, allow the user to look up the value in a table or automatically fill in the value once the user enters enough significant characters. Fifth, Provide formatting examples when appropriate; for example, automatically show standard embedded symbols, decimal points, credit symbol, or baht sign. Finally, Automatically justify data entries; numbers should be right-justified and aligned on decimal points, and text should be leftjustified.

When entering data, the users need not to be specified the dimensional units of a particular value. For example, a user no needs to specify that an amount is in bahts or that a weight is in tons. Field formatting and the data entry prompt make clear the type of data being requested. In other words, a caption describing the data to be entered also adjacent to each data field. Within this caption, it is clear to the user what type of data is being requested. As with the display of information, all data enter onto a form automatically justify in a standard format.

3.5.4 Data Input Control

One objective of interface design is to reduce data entry errors. As data are entered into an information system, steps must be taken to ensure that the input is valid. In designing system's interface, the types of errors users may make are anticipated, and features of system's interface are designed to avoid, detect, and correct data entry mistakes. There are some types of data input's errors that are most usually occurred. Appending is adding additional characters to a field. Truncating is losing characters from a field. Transcripting is entering invalid data into s field. And transposing is reversing the sequence of one or more characters in a field. In essence, data errors can occur from appending extra data onto a field, truncating characters off a field, transcripting the wrong characters into a field, or transposing one or more characters within a field.

Practical experience has also found that correcting erroneous data is much easier to accomplish before it is permanently stored in a system. On-line systems can notify a user of input problems as data are being entered. When data are processed on-line as events occur, it is much less likely that data validity errors will occur and not be caught. In an on-line system, most problems can be identified and resolved before permanently saving data to a storage device using many of the techniques. However, in systems where inputs are stored and entered in batch, the identification and notification of errors is more difficult. Batch processing systems can reject invalid inputs and store them in a log file for later resolution.

In addition to validating the data values entered into a system, controls was be established to verify that add input records are correctly entered and that they are only processed once. A common method used to enhance the validity of entering batches of

data record is to create an audit trail of the entire sequence of data entry, processing, and storage.

3.5.5 Feedback Providing

In designing system interfaces, providing appropriate feedback is an easy method for making user's interaction more smoothing. If not providing feedback, it is to frustrate and confuse the users. There are three types of system feedback.

Providing status information is a simple technique for keeping users informed of what is going on within a system. For example, relevant status information such as displaying the current customer name or time, placing appropriate titles on a menu or screen, or identifying the number of screens following the current one all provide needed feedback to the user. Providing status information during processing operations is especially important when the operations takes longer than a second or two. For example, when opening a system interface, it will display "please wait while the file is opening", or when processing the data, it will show "loading".

A second feedback method is to display prompting cues. When prompting the user for information or action, it is useful to be specific in the request.

Errors and warning messages is the third method available for providing system feedback. The access system's interface is shown in Figure 3.28. The select function's interface is shown in Figure 3.29. The record purchase order's interface is shown in Figure 3.30. The generate product requisition bill's interface is shown in Figure 3.31. The generate sales contract's interface is shown in Figure 3.32. The record delivery note's interface is shown in Figure 3.33. The generate tax invoice's interface is show in Figure 3.34. The generate receipt's interface is shown in Figure 3.35. The generate report's interface is shown in Figure 3.36.

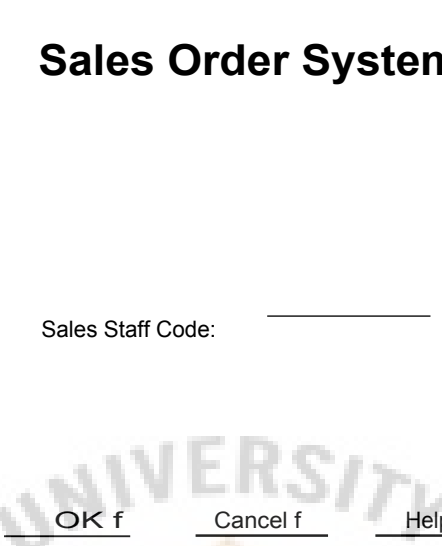


Figure 3.28. Access System's Interface of the System.

Figure 3.29. Select Function's Interface of the System.

Form

- Islam

Record Purchase Order

Staff Code: 00i
Record No: 310
Record Date 8/10/45

Sales Section:

Purchase Order No: 013
Customer ID: 089
Customer Name: Kin Heng Seng
Customer Address: 95/265 Rama 3Rd. Pan ,k 1)1
Remark:

Purchase Order Date 8/10/45

Item:	Product ID:	Description:	We	Amount:	Pnce/kg:
1	C-042306	C-Channel 100x50x20x2 3mmx6m	12 kg	1,00 pi,,,, Pieces.	15.0 baht.
				Pieces.	baht.
				Pieces.	baht.
				Pieces.	baht.

OK Cancel

Figure 3.30. Record Purchase Order's Interface of the System.

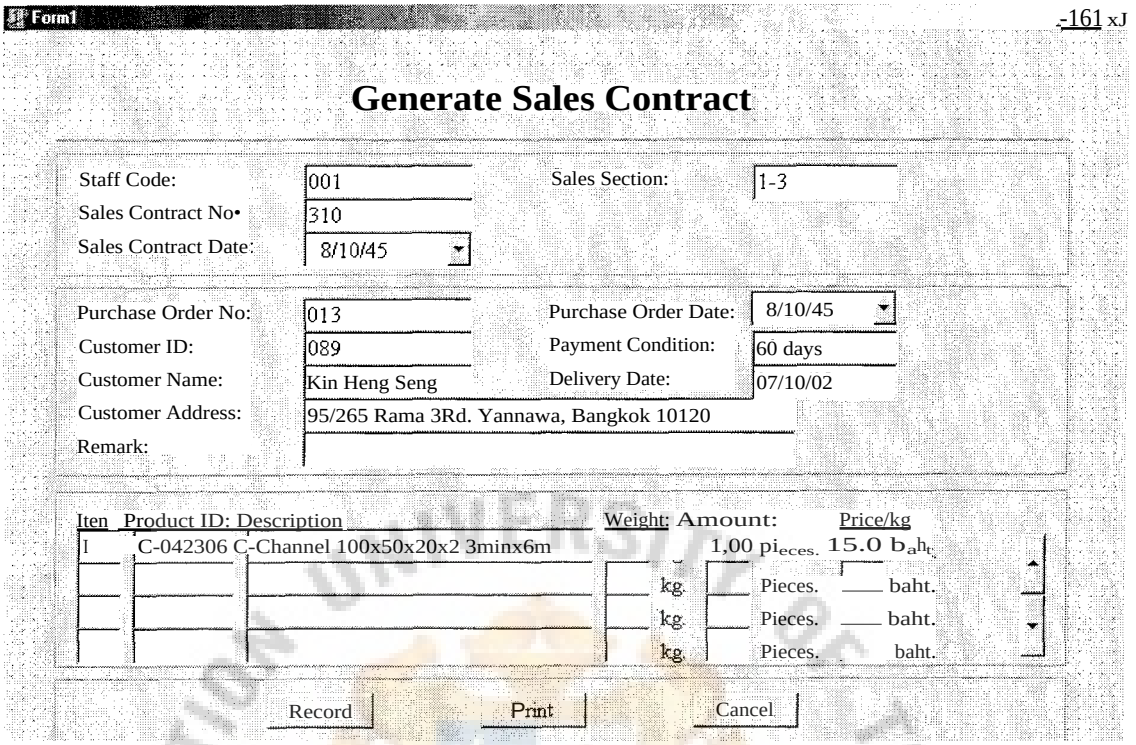


Figure 3.31. Generate Product Requisition Bill's Interface of the System.

—161? ci

Generate Product Requisition Bill

Staff Code: 001 'Sales Section:

PRB No: 310

PRB Date 8/10/45

Sales Contract No 015 ales Contracr Date: 8i1 *ul'4~*

Customer ID: 089

Customer Name: Kin Heng Seng

Customer Address: 95/265 Rama 3 Rd. Yaltriav,7a., 11-11

Delivery Place: 95/265 Rama 3 Rd. Yannawa, Bangkok 1 i) 121a

Item:	Product ID:	Description:		Amount	Price
	C-042306	C-Channel 10 Ox50x2 Ox2. 3minx6m	2	1,00 Pieces.	15.0 baht.
				Pieces.	baht.
				Pieces.	baht.
				Pieces.	baht.

Record P Cancel

Figure 3.32. Generate Sales Contract's Interface of the System.

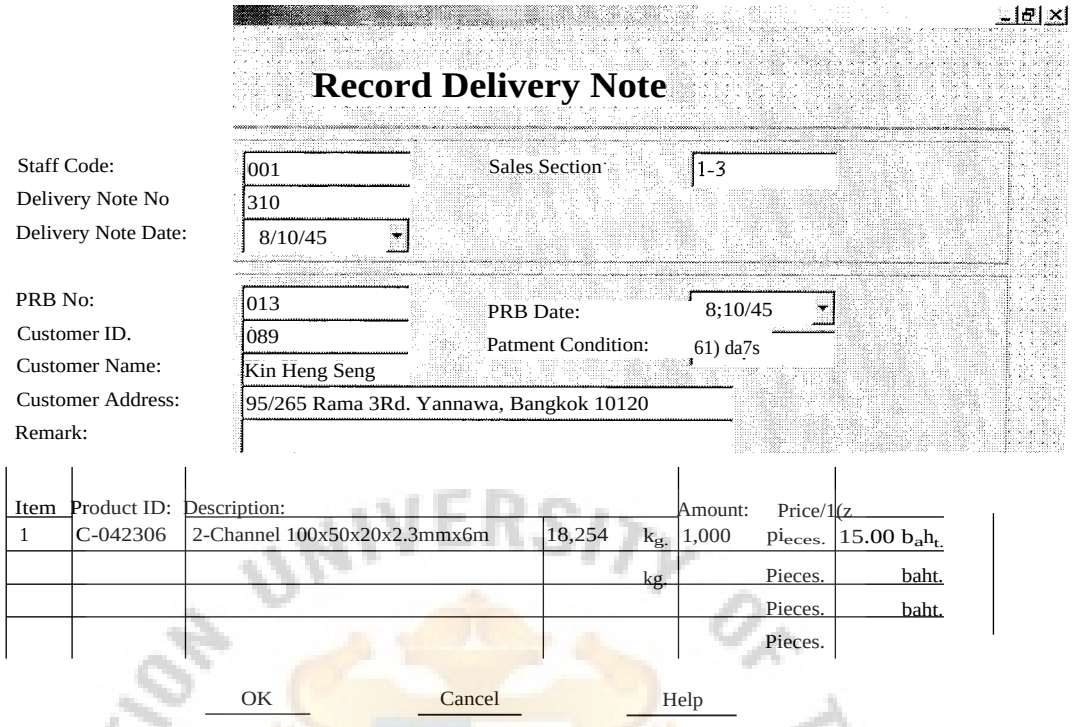


Figure 3.33. Record Delivery Note's Interface of the System.

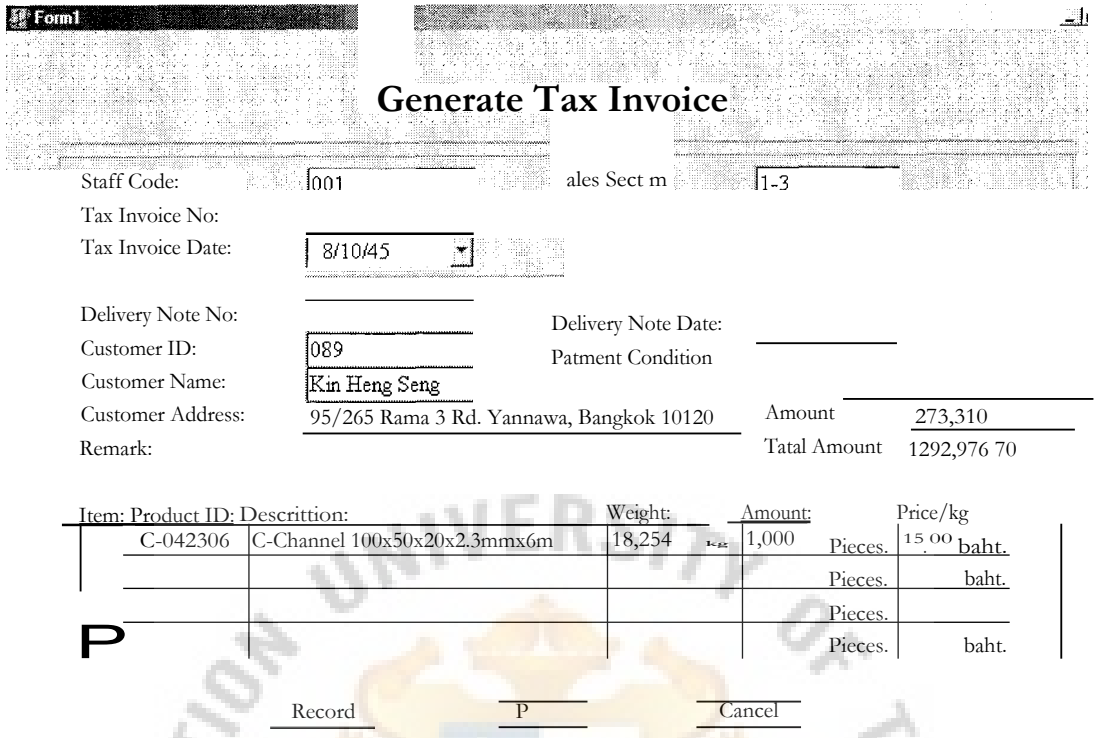


Figure 3.34. Generate Tax Invoice's Interface of the System.

Generate Receipt

Staff Code: 001 Sales Section.
Receipt No: 310
Receipt Date: 8/10145

Tax Invoice No: 013
Customer ID: 089 Tax Invoice Date: „. 1 LIMIS
Customer Name: Kin Heng Sting Patment Condition: 16. days
Customer Address. 95/265 Rama 3 Rd. Yannawa, Bangkok 10120 Amount: 273,810
Remark: Tatal Amount 292,976.

Item:	Tax Invoice No.	Amount	Vat:	Total Amount:		Remark:
1	T-075	273,810	19,166.7	292,976.70	baht	
					baht	
					baht.	
					baht.	

Record Print 7_1anCel

Figure 3.35. Generate Receipt's Interface of the System.

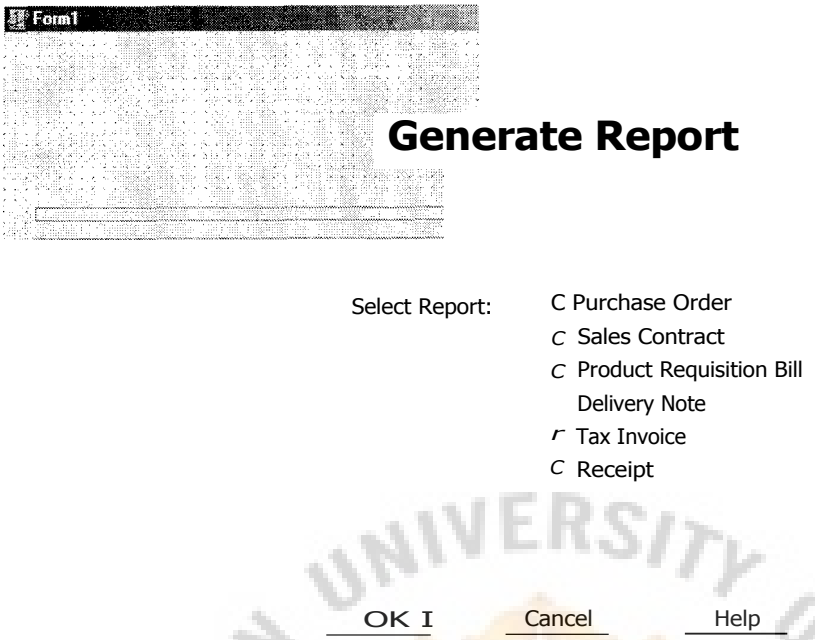


Figure 3.36. Generate Report's Interface of the System.

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969

3.5.6 Dialogues Designing

The process of designing the overall sequences that users follow to interact with an information system is called dialogue design. A dialogue is the sequence in which information is displayed to and obtained from the users. The point is to select the most appropriate interaction methods and devices and to define the conditions under which information is displayed and obtained from users.

There are three major steps in dialogue designing. First, designing dialogue sequences is to gain an understanding of how users might interact with the system. This means that it has to understand the user, task, technological, and environmental characteristics in designing dialogues. For the Sales Ordering System, the typical dialogues for the users and the system to obtain the order processing's transactions are:

(1) Request to enter the system

(2) Select the system functions

(3) Enter into the system function

(4) Input data

A formal method for designing and representing dialogues is dialogue diagramming. Dialogue diagrams have only one symbol, a box with three sections; each box represents one display within a dialogue. The three sections of the box are used as; top contains a unique display reference number used by other displays for referencing it, middle contains the name or description of the display, bottom contains display reference numbers that can be accessed from the current display. All lines connecting the boxes within dialogue diagrams are assumed to be bi-directional and thus do not need arrowheads to indicate direction. This means that users are allowed to always move forward and backward between adjacent displays.

Building dialogue prototypes and assessing usability are often optional activities. Some system may be very simple and straightforward. Others may be more complex but are extensions to existing systems where dialogue and display standards have already been established. Building prototype displays is done by using the graphical development environments as Microsoft's Excel.

The dialogue diagram illustrating sequence, selection, and iteration of Sales Ordering System is shown in Figure 3.37. In this diagram, the analyst placed the operation of sales ordering. The user must first gain access to the system through access system (item 0). Once the user select the functions (item 1) to perform. There are totally seven functions for the user to select. Seven functions are represented the identical work to perform, which are record purchase order (item 2), generate product requisition bill (item 3), generate sales contract (item 4), record delivery note (item 5), generate tax invoice (item 6), generate receipt (item 7), and generate report (item 8). Other sub-displays are product requisition bill faun (item 3.1), sales contract form (item 4.1), tax invoice form (item 6.1), and receipt form (item 7.1). For the report generation, there are purchase order (item 8.1), production requisition bill (item 8.2), sales contract (item 8.3), delivery note (item 8.4), tax invoice (item 8.5), and receipt report (item 8.6).

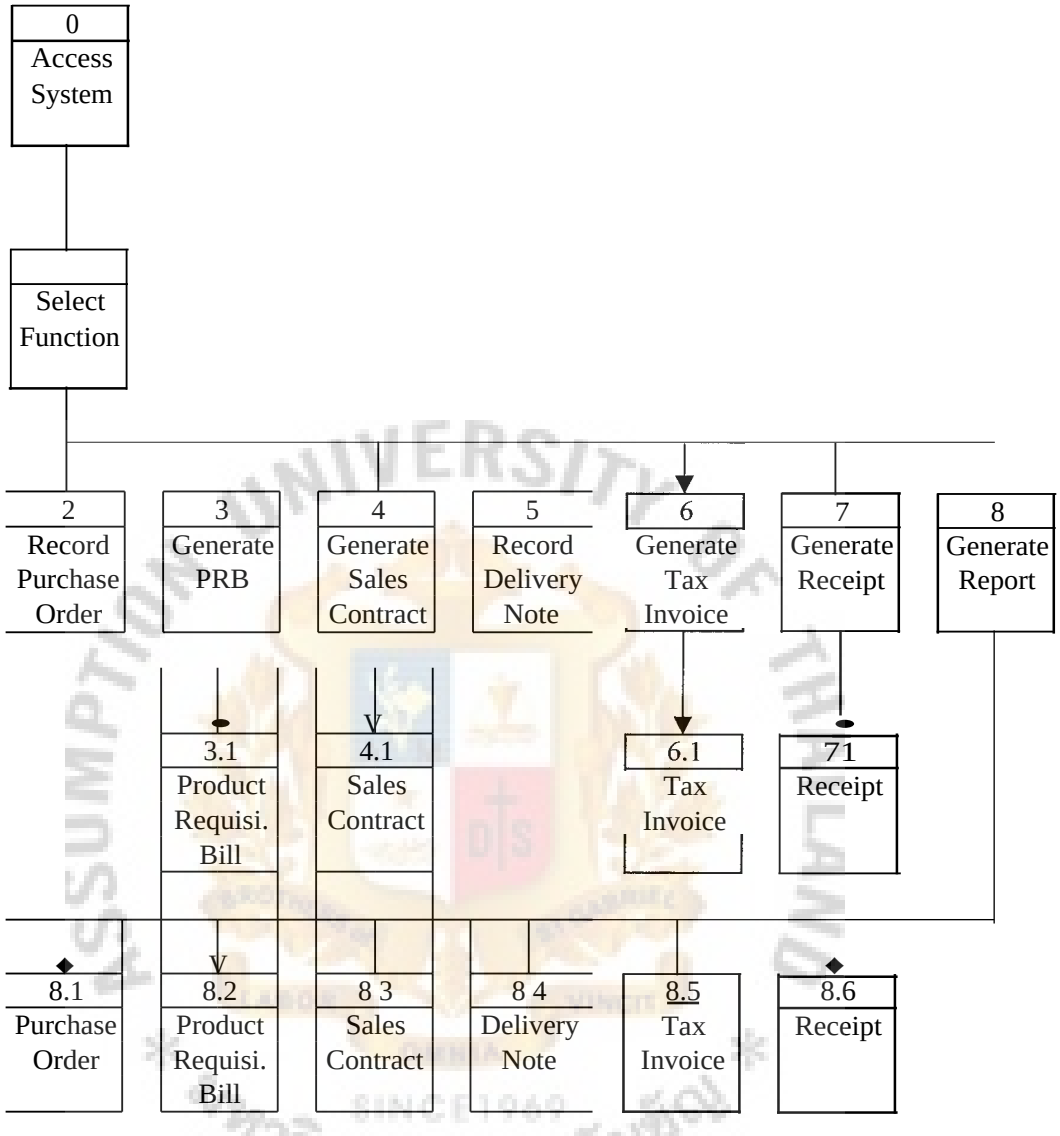


Figure 3.37. Dialogue Diagram Illustrating Sequence, Selection, and Iteration of Sales Ordering System.

St. Gabriel's Library, Au

3.6 Program and Process Design

Processes take incoming data and transform them or route them to other locations in the information system. In designing programs and processes, it will not immediately begin to write computer code. Coding does not begin until the implementation phase. The focus in designing the internals is to turn the previously defined relationships among data, processing, and output into a detailed blueprint programmers can use as a guide to writing programs code once implementation begins.

There are two primary outcomes of the physical design process for the system's internals: a set of structure charts and a set of physical design specifications for each separate part of the system. In addition to providing the functional descriptions for each part of the system, it must also provide information about input received and output generated for each program and its component parts. The specifications for input to and output from the database will come from the physical database description, in the form of pseudocode or a special pseudocode called action diagrams. All of the data that appear in the structure charts will have already been defined in the database or in inputs form designs. Specifications for modules that capture system inputs or produce system outputs may be in narrative form since the processing logic of these modules is usually quite simple. If input modules also include procedures to validate data, handle errors in the data, or capture different data depending on the data that are input, then pseudocode can also be used to explain these system modules. These specifications will be supplemented by the actual form, report, and dialogue designs.

The most common architecture for representing the physical structure of a system is hierarchical. A hierarchy looks like an inverted tree or organization chart with one root or main routine at the top and multiple levels of other modules nested underneath. A structure chart shows how as information system is organized in a hierarchy of

components, called modules. The purpose of a structure chart is to show graphically how the parts of a system or program are related to each other, in terms of passing data and in terms of the basic components of structured programming: sequence, selection, and repetition. A structure chart redefined the flow and processing of data from data flow diagrams into a structure of system components that follow certain principles of good program design. Structure charts are used to show the breakdown of a system into programs and the internal structure of programs written in newer object-oriented or event-driven programming languages is usually depicted by state-transition diagrams and Structure-English. A computer program is typically made up of several modules. Modules may also represent separately complied programs, subprograms, or identifiable internal procedures. In structure chart, there is a single coordinating module at the root and on the next level are modules that the coordinating module calls. Each of the modules that report to the coordinating module may call additional modules in the next row down. Modules at the lowest levels do not call any other modules, instead they only perform specific tasks. Middle-level modules act as coordinating modules for those lower-level modules they control but may perform some processing as well. In a structure chart, each module is represented as a rectangle containing a descriptive name of its function. Each module name is concisely and accurately reflect what that module does. When naming modules, it should avoid names that include conjunctions as conjunctions indicate that the module performs more than a single function. Modules are called in order from left to right.

Modules in a structure chart communicate with each other through passing parameters. These parameters take the form of data, represented as data couples and flags. A data couple shows data being exchanged between two modules. A flag shows control data, or a message, being passed between modules. A data couple is usually a

single data element, although it can also be a data structure or even an entire record. A flag represents information the system needs for processing; a flag itself is never processed.

Information system are typically either transaction-centered or transform-centered. In a transaction-centered system, the system's primary function is to send data to their proper destinations within a more general system. Data come into the center module of the system, the transaction center, and they are dispatched to their proper locations based on their data type. A transform-centered system has as its center function the derivation of new data values from existing data values. The modules that perform the task of bringing data into the system are called afferent modules. Afferent modules are arranged in groups referred to as afferent branches. The modules that perform tasks associated with the output of the transformed data are called efferent modules, which are arranged in efferent branches.

The goal of transform analysis is to convert a transform-centered data flow diagram into a structure chart remains faithful to the system description contained in the data flow diagram. The transform analysis process has two main parts: top-level design and detail design. The goal of top-level design is to create a structure chart that capture the essence of the afferent center transform and efferent processes in a data flow diagram. The next step is to find a boss or coordinating module. Generally, coordinating modules are not evident in a data flow diagram. Data flow diagrams map the flow of information through a system, and while such a mapping is very useful in identifying key data and processes in a system, there is no one-to-one correspondence between the processes in a data flow diagram and the modules in a structure chart.

In order to refine the structure chart, it needs to further develop the afferent and efferent branches and, if necessary, the center transform. Any addition refinements

structure chart will be in the form of design improvement. The complete refined structure chart of the Sales Ordering System is depicted in Figure 3.38. In the Structure chart diagram, purchase order data (POD) is received from the get purchase order data module. This data passes into a system. In addition, inventory data and credit data are received from the Warehouse/Operations department and Finance/Accounting department respectively. All of these data are flowed into accept purchase order module. This module verifies, compares, and evaluates all of three data and produces accept purchase order. Accept purchase order inputs into generate sales contract module and generate product requisition bill module. The system also receives delivery note data from the get delivery note data module. The data is sent into generate tax invoice module. Tax invoice data is received from get tax invoice data module and sent into generate receipt module.

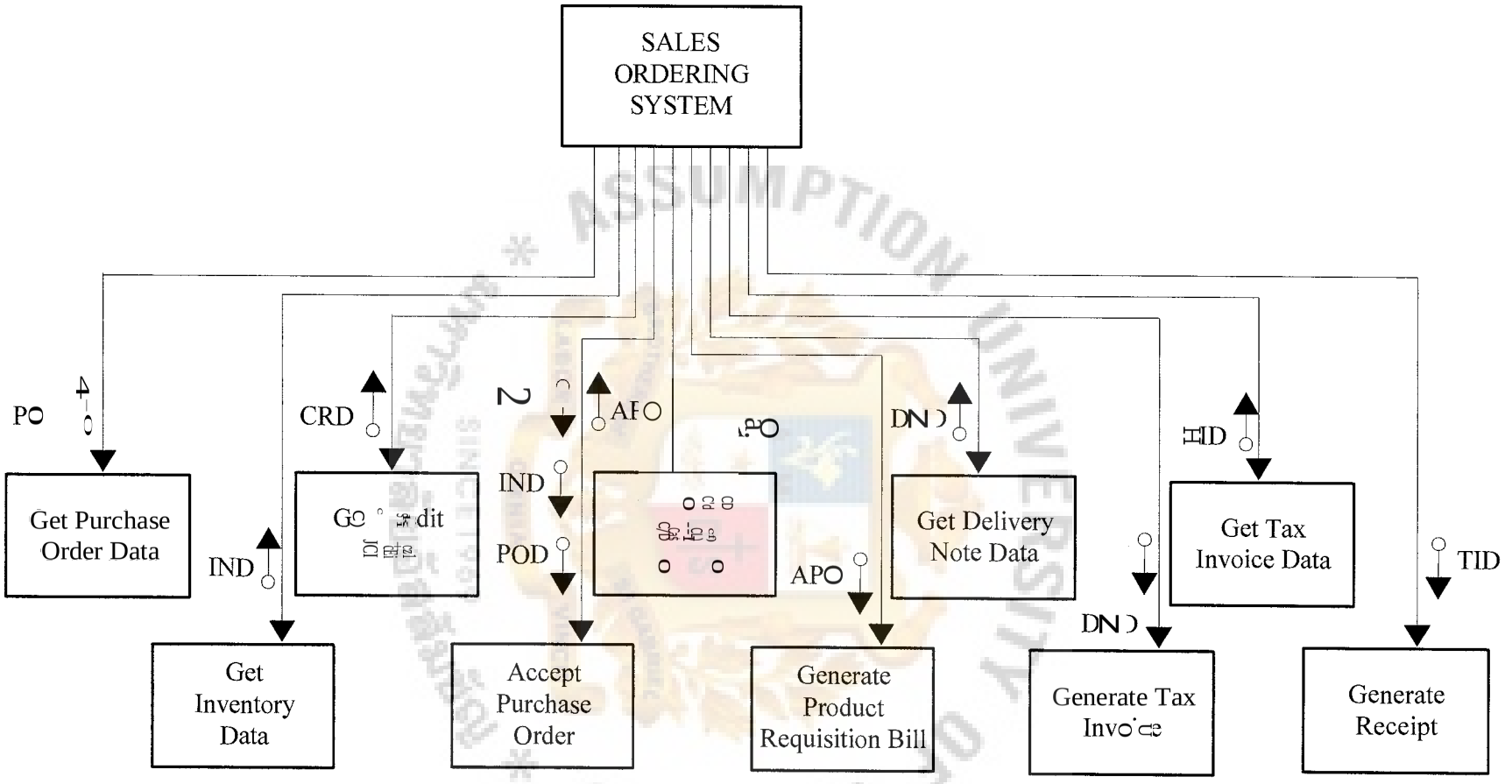


Figure 3.38. Complete Refined Structure Chart for the Sales Ordering System.

IV. SYSTEM FEASIBILITY

4.1 Economic Feasibility

The system feasibility is a required activity and is potentially a large undertaking in evaluating a wide range of factors. There are some important feasibility factors as economic, technical, and operational.

The purpose of assessing economic feasibility is to identify the financial benefits and costs associated with the development system. Economic feasibility is often referred to as cost-benefits analysis. During the system initiation and planning, it will be impossible for the analyst to precisely define all benefits and costs related to a particular system. It is feasible to justify the benefits and costs after the real-time justified. In determining the benefits of the system, Sales Ordering System can provide many benefits to an organization. For the operations, the system can automate monotonous jobs, reduce errors, provide innovative service to customers and suppliers, and improve organizational efficiency and effectiveness.

The system benefits can be divided into two types, which are tangible benefits and intangible benefits. Tangible benefits refer to items that can be measured in currency and with certainty. Most tangible benefits are cost reduction and avoidance, error reduction, increase flexibility, increase speed of activity, improvement of management planning and control, and opening new markets and increasing sales opportunities.

The tangible benefits for Sales Ordering System is depicted in Figure 4.1. The values in the figure are estimated after collecting information from users of the current system. With the proposed system, the cost reduction and avoidance are estimated. In the proposed system, many types of resource are reduced such as office equipments.

The man-hour needed to process and issue the documents is reduced, so the staff can use that hours to process other works. Error reduction is another tangible benefit from the proposed system. In the proposed system, some errors are reduced and the staffs are used less time to correct errors. In the existing system, the procession of sales ordering incurred the errors, and the staffs usually their 10% of work-hour to correct errors. Therefore, thirteen of administrative staff needed to spend their 10% of work-time to correct the errors.

Tangible Benefits Worksheet Sales Ordering System		Year 1 through 3
A.	Error Reduction and Avoidance (15,000 x 13 x 10% x 12months)	Bht. 234,000
B.	Increased Flexibility and Speed of Activity (15,000 x 13 x 20% x 12months)	Bht. 468,000
	Total Tangible Benefits	Bht. 936,000

Figure 4.1. Tangible Benefits for Sales Ordering System.

Increased in flexibility is another tangible benefit in the proposed system. By utilizing the proposed system, the sales and administrative staffs will expect to have more time to process other works. The other works are servicing the customer, following the delivery, or prepare useful reports. Increased speed of activity also includes in tangible benefits of the proposed system. By using the new proposed system, the staffs can process more works than before. The additional works are estimated to be 20% of the existing works.

It is necessary to identify several intangible benefits of the proposed system. Intangible Benefits refer to item that cannot be easily measured in currency or with certainty. Intangible benefits may have direct organizational benefits such as the improvement of employee morale or they may have broader societal implications such as the reduction of waste creation or resource consumption. Potential intangible benefits may have to be the employee development in the computer-specific programming and the understanding in the scope of the work of the organization. Other intangible benefits are competitiveness necessity, more timely information, improved organization planning, increased organizational flexibility, promotion of organizational learning and understanding, availability of new, better, or more information, ability to investigate more alternatives, faster decision making, information processing efficiency, improved asset utilization, improved resource control, increased accuracy in administrative operations, improved work process that can improve employee moral, and positive impacts on society.

In determining cost, it can also divide into two types. There are tangible cost and intangible cost. Tangible costs refer to items that can easily measure in currency or with certainty. Tangible costs are hardware cost, labor cost, and operation cost. Tangible costs can further divide into two types, which are one-time cost and recurring cost. One-time costs refer to those costs resulting from the ongoing evolution and use of the system. For example, application software maintenance, incremental data storage expense, incremental communications, new software and hardware leases, and supplies and other expenses are the recurring costs. One-time costs refer to those associated with the system initiation and development and the start-up of the system. These costs typically encompass activities such as system development, new hardware and software purchases, user training, site preparation, and data or system conversion.

By developing the proposed Sales Ordering System, the one-time costs Include system development cost and user training cost. System development cost is estimated to be fifty thousand for the proposed system. This cost includes the analysis of the existing operations of the analyst and the design of the proposed system. User training cost is estimated to be Twenty thousand. In training users, it needed to spend at least one week of the operations of the information system staff. And the sales administrative staff will spend an equal time in training. Since the company invested in hardware and communication equipment initially, therefore, one-time cost in developing the proposed of Sales Ordering System is not as high as it could be. The one-time costs for Sales Ordering System are shown in Figure 4.2.

One-Time Costs Worksheet Sales Ordering System	Year 0
A. System Development Cost (50,000 x 1)	Bht. 50,000
B. User Training Cost (20,000 x 1)	Bht. 20,000
Total One-Time Cost	Bht. 70,000

Figure 4.2. One-Time Costs for Sales Ordering System.

For the recurring cost, there is incremental data storage, which estimated to be one thousand for a month. Incremental communication cost is also expected to be one thousand a month. Supplies and other expenses are expected to be one thousand for each month. This expense is included the software maintenance. In developing the

proposed system, two additional supporting end-user staff will be hired. The salary cost is expected at fifteen thousand for each staff per month. The recurring costs for Sales Ordering System are shown in Figure 4.3. The summary spreadsheet reflecting the present value calculations of all benefits and costs for the Sales Ordering System is depicted in Figure 4.4. In this Figure, the discount rate used in the calculation of both present values of benefits and costs is minimum loan rate (MLR) at 8%. Break-even analysis for the Sales Ordering System is shown in Figure 4.5.

Recurring Costs Worksheet Sales Ordering System	Year 1 through 3
A. Incremental Data Storage Cost (1,000 x 12months)	Bht. 12,000
B. Incremental Communication Cost (1,000 x 12months)	Bht. 12,000
C. Supplies and Other Costs	Bht. 12,000
D. Labor Cost (15,000 x 2 x 12months)	Bht. 360,000
Total Recurring Cost	Bht. 396,000

Figure 4.3. Recurring Costs for Sales Ordering System.

Sahaviriya Panich Company
Economic Feasibility Analysis
Sales Ordering System

Year of Project	Year 0	Year 1	Year 2	Year 3	Totals (Bht.)
Net Economic Benefit	0	702,000	702,000	702,000	
Discount Rate (8%)	1	0.9260	0.8570	0.7940	
Present Value Of Benefits	0	650,052	601,614	557,388	
Net Present Value of All Benefits	0	650,052	1,251,666	1,809,054	1,809,054
One-Time Costs	(70,000)				
Recurring Costs	0	(396,000)	(396,000)	(396,000)	
Discount Rate (8%)	1	0.926	0.857	0.794	
Present Value of Recurring Costs	0	(366,696)	(339,372)	(314,424)	
Net Present Value of All Costs	(70,000)	(436,696)	(776,068)	(1,090,492)	(1,090,492)
Overall Net Present Value					718,562
Overall Return on Investment					0.66
Break-Even Analysis					
Yearly Net Present Value Cash Flow	(70,000)	283,356	262,242	242,964	
Overall Net Present Value Cash Flow	(70,000)	213,356	475,598	718,562	

System break-even occurs between year 0 and year 1
Use first year of positive cash flow to calculate break-even fraction
((283,356 - 213,356) / 283,356) = 0.25
Actual break-even occurred at month 3 of the first year.

Figure 4.4. Summary Spreadsheet Reflecting the Present Value Calculations of All Benefits and Costs.

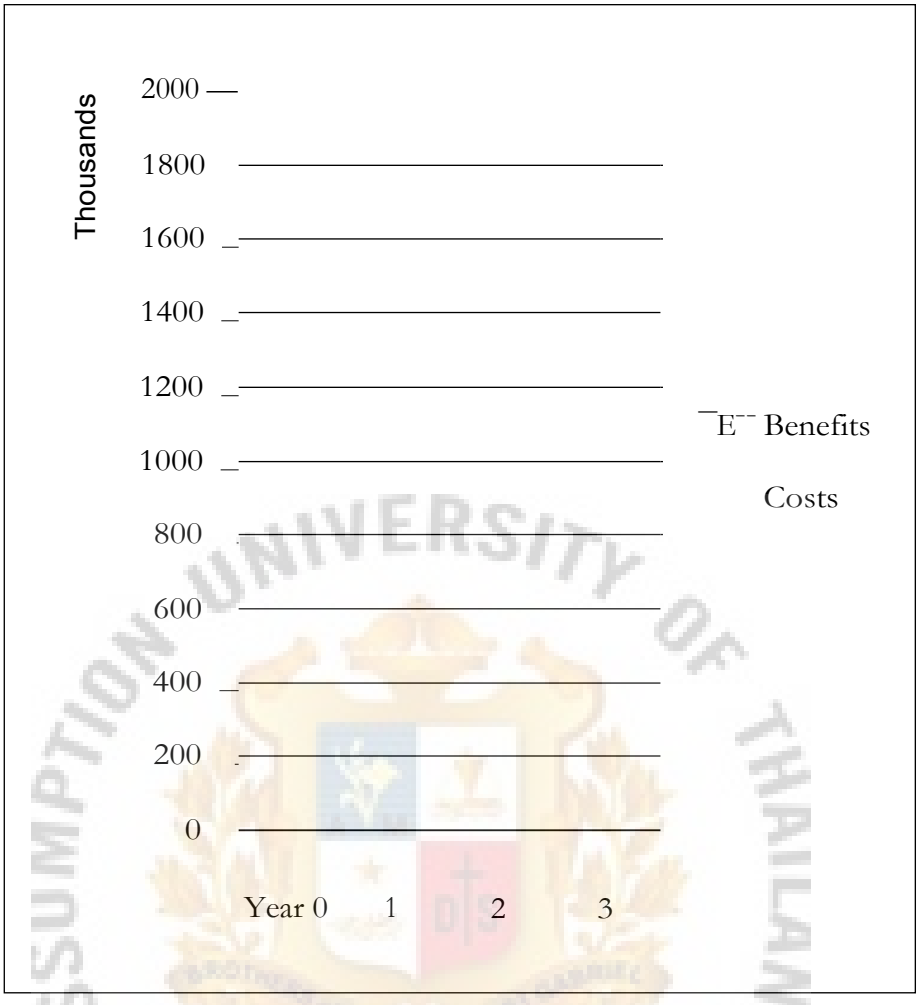


Figure 4.5. Break-Even Analysis for Sales Ordering System.

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

4.2 Verification and Validation

In verification, the Modules, units, and integrated system are tested in order to verify whether the system processes according to the system development's objectives. Four steps of testing are carried out, which are inspection, unit testing, integration testing, and system testing.

In inspection step, participants manually examine code for occurrences of well-known errors. Also, inspection technique can test by examining the code in order to search for predictable language-specific errors. By using inspection technique, it found that not only program codes but also system specifications prepared by the analyst are correct. Inspection testing is done in each module of the functions, which are recording purchase order, generating sales contract, generating production requisition bill, record delivery note, generating tax invoice, and generating receipt.

In unit testing technique, each module is tested alone in an attempt to discover any errors in its code. Like inspection testing, the modules are categorized in order to test. The module's code is tested and errors are detected before a next step of testing.

In integration testing, modules are bringing together and integrated tested. The process of bringing together all of modules that a program comprises for testing purposes. Modules are typically integrated in a top-down, incremental fashion. In the integration testing, the coordinating module (recording purchase order) is gradually tested, inventory checking and credit checking modules are added. The remain of the same level-module are added, which are accepting purchase order, generating sales contract and product requisition bill, recording delivery note, generating tax invoice and receipt. Once the program has been tested with the coordinating module and all of its immediately subordinate modules, the procedures are continued until the entire program has been tested as a unit.

In system testing, the bringing together of all the programs that a system comprises for testing purposes. Programs are typically integrated in a top-down, incremental fashion. The same process and the interfaces between the programs are tested.

For the validation, the system of sales ordering is determined and evaluated against the system objectives. There are three areas to prove for the validation of the system

First, the effectiveness and efficiency of the sales ordering operations was improved after adopting the new computerized system of Sales Ordering System. In the new computerized system, the errors of manual works are reduced and quality is improved. The fewer times that sales staffs used in processing the order let them an additional time to perform other work and gave them the flexibility in their job.

Second, since the company invested over three millions in the computer hardware and software, it was feasible for the company in the development of the computerized system that can increase the effectiveness and efficiency of the operations. In additional, the development of the new computerized system would require the hardware and software that existed in the existing technology of the company. That means the new system would optimally utilize the company-technology resource.

Third, the economic feasibility of the new computerized system shown the net benefit in the first year of implementation. After determining benefits and costs, the net present values of benefits and costs are calculated. The overall of net present value shown the benefits of implementing the new computerized system.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Sahaviriya Panich Company operates in steel product trading industry for more than forty years. Since the company dominated over sixty percents of the market share, the company is the leader in almost every kinds of steel product. The company not only concentrated in the domestic market but also in the export market. In 2002, the export orders are quantified larger than the domestic order.

As the company growth rate and market share increased, the company operations were ineffectively and inefficiently to support the current conditions. The sales order processing is currently manual. Long time processing and more errors are the obstacle of the company operations. The development of the computerized system would be helped in increasing the effectiveness and efficiency of the company sales order processing. Many activities must be carried out for the development of the new system.

In developing the new computerized system, steps begin with the collection of necessary information from the users of the system. Information such as the company's existing information system, application portfolio, and operation system are analyzed. The design and development of the new computerized system begin with the process modeling. In process modeling, the context, level-0, and level-1 data flow diagrams are created to depict the process of new computerized system. In logic modeling, Decision tables and trees are designed to describe the logic of the modules. In conceptual data modeling, the entity-relationship diagrams are created to describe the relationships between the data.

Forms and reports are designed to contain predefined data of the sales order processing. Interfaces and dialogues are also created to interact between users and the

information system. Finally, structure chart diagram is created to show how an information system is organized.

System Feasibility includes economic feasibility and verification and validation. An economic feasibility shows the reflecting of the present value calculations of all benefits and costs for the new computerized system. In verification, the internal program codes are inspected for the errors. In validation, many reasons are described to prove for the development of the new computerized system.

5.2 Recommendations

It should be clear that the computer processing ability is the helping hand for the user of the system in speeding up the works, therefore, all of the users, operating-level and top-level management should not rely too much on computer ability by overlooking the important of people. In the other hand, all of parties must understand how the computerized-integration system increases the effectiveness and efficiency of the company operations.

Since the structuring information was gathered from the sales staffs, who currently process order transaction by manual, the existing system was analyzed according to that information, and the analyst developed the proposed system from the structuring requirements. Therefore, the proposed system functions might not completed when implemented for a while. Anyway the analyst can analyze the additional functions and develop into the system.

From the research, it expected that the benefits from the system will occur during first year and continued over third year. If the company sales and growth rate still continue in that time, the operations are expected to expand. System specification's requirement needed to be assessed for the system expansion. After assessing of the

specification's requirement, the required software and hardware will be added into the system.

In backing up the information of the system, Information System department needs to maintain the spare of memory in order to support an extra of infoiniation. Considering the location for the main operations of the system hardware are another important thing, since the security of these equipments may be harmed from the unauthorized parties.

Although the new computerized system will increase the effectiveness and efficiency of the company operations, the system is separated in the operation of sales order processing. For a definitely effectiveness and efficiency, the new computerized system should be integrated with the other two systems, which are the customer credit checking and inventory management system. This will help the users of all three systems in their operations.

By integrating all three systems, the users can once check inventory status which prepared by Warehouse and Operations department, access and get customer credit information which prepared on daily basic by Finance and Accounting department, and finally, process order transactions for the customer.

For the integrated system, the users of the system do not include only the sales staff of Sales and Marketing department, but also the staff of Warehouse/Operations department and Finance/Accounting department. Therefore, it should be clear for all users of the system that how to access and use the integrated system. The effective way to guarantee the using of the integrated system by all users is to train them periodically, and make them understanding clearly the objectives of the system.

BIBLIOGRAPHY

1. Blank, L. T. and Tarquin, A. J. Engineering Economy, Mcgraw-Hill Companies, Inc., 1998.

2. Hoffer, J. A., George, J. F., and Valacich. J. S. Modern System Analysis & Design, Addison Wesley, 1999.

3. Laudon, K. C., and J. P. Laudon. Management Information System. New Jersey: Prentice-Hall International, Inc., 2000.

4. Martin E. W., D. W. Dehayes, J. A. Hoffer and W.C. Perkins. Managing Information Technology. New York: Macmillan Publishing Company, 1994.

5. Pressman, R. S. Software Engineering. New York: Mcgraw-Hill, 1992.

ASSUMPTION UNIVERSITY OF THAILAND
LABOR OMNIA VINCIT
SINCE 1969

St. Gabriel's Library, Au