

**STUDY OF BILL PRESENTMENT AND PAYMENT MODELS
USING DEPENDENCY RELATIONS**

by
DOROTHY WONG

B.A.Sc., University of British Columbia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN BUSINESS ADMINISTRATION
in
THE FACULTY OF GRADUATE STUDIES
(Management Information Systems Division,
Faculty of Commerce and Business Administration)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 2001

© Dorothy Wong, 2001

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Commerce (Business Administration)

The University of British Columbia
Vancouver, Canada

Date Oct 9th, 2001

Abstract

Electronic Bill Presentment & Payment (EBPP) involves the automation of the bill presentment and payment processes over the Internet. As EBPP becomes more widely used, different models from it have emerged. However, none of them is centered around the customer's needs.

This thesis suggests that a customer-oriented model can be implemented using Intelligent Agent (IA). This model can work as well as the models available in the market, but it has not come into existence. The question of why such a model does not exist is then investigated. There are two parts to this research: prototype implementation of the IA Model assesses its technical and economical feasibilities, and analysis based on the Dependency Network Diagram (DND) and literature research investigates the strategic and organizational feasibilities of the model.

This research paper gives the reader an understanding on the dependency relations in billing models and how these relations determine the success or failure of a model. Management of Information Systems (MIS) researchers and managers can apply the results of this thesis when they design technologies that involve dependency relations.

Table of Contents

ABSTRACT	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	iv
LIST OF FIGURES	v
ACKNOWLEDGEMENTS	vi
1.0 INTRODUCTION	1
2.0 RELATED WORKS	8
2.1 EBPP AND EXISTING MODELS	8
2.2 INTELLIGENT AGENT IN E-COMMERCE	13
3.0 THE INTELLIGENT AGENT EBPP MODEL	16
3.1 OVERVIEW OF MODEL	16
3.2 ARCHITECTURE AND IMPLEMENTATION	19
4.0 DEPENDENCY RELATIONS	42
4.1 THEORY AND BACKGROUND INFORMATION ON THE DEPENDENCY NETWORK DIAGRAM	42
4.2 CONSTRUCTING THE DEPENDENCY NETWORK DIAGRAM – STEPS AND RULES	46
4.3 DEPENDENCY NETWORK DIAGRAMS OF THE MODELS	49
4.3.1 <i>Biller Direct</i>	49
4.3.2 <i>Bank Forward</i>	52
4.3.3 <i>Consolidator</i>	55
4.3.4 <i>Intelligent Agent</i>	59
5.0 ANALYSIS ON MODELS	63
5.1 COMPARISON ON MODELS	63
5.2 SHORTCOMINGS OF THE IA MODEL	73
6.0 CONCLUSION	81
BIBLIOGRAPHY	85

List of Tables

TABLE 1. COMPUTATIONAL COMPLEXITY OF VARIOUS STEPS..... 33

TABLE 2. COMPARISON OF FEATURES 35

TABLE 3. BALANCES IN BANK ACCOUNTS 36

TABLE 4. NUMBER OF DAYS REQUIRED FOR TRANSFERS..... 37

TABLE 5. TEST RESULTS OF PROTOTYPE..... 37

List of Figures

FIGURE 1. SIMPLIFIED RELATIONSHIP DIAGRAM IN THE BILLER DIRECT MODEL	9
FIGURE 2. SIMPLIFIED RELATIONSHIP DIAGRAM IN THE BANK FORWARD MODEL	10
FIGURE 3. SIMPLIFIED RELATIONSHIP DIAGRAM ON THE CONSOLIDATOR MODEL	12
FIGURE 4. OVERVIEW OF INTELLIGENT MODEL IN SIMPLIFIED VIEW	18
FIGURE 5. OVERVIEW OF MODEL (IA BROKEN DOWN INTO SUB-COMPONENTS)	20
FIGURE 6. E-R DIAGRAM FOR TABLES IN IA'S KNOWLEDGE BASE	22
FIGURE 7. EXECUTION MODULE OF IA IN SUB-COMPONENTS	28
FIGURE 8. FLOW CHART FOR PAY/TRANSFER	30
FIGURE 9. FLOW CHART TO GENERATE BEST SOLUTION	34
FIGURE 10. AN EXAMPLE OF A DND	45
FIGURE 11. DND FOR BILLER DIRECT MODEL	49
FIGURE 12. DND OF BANK FORWARD MODEL	52
FIGURE 13. DND FOR CONSOLIDATOR MODEL.....	55
FIGURE 14. DND FOR THE INTELLIGENT AGENT MODEL.....	59

Acknowledgments

I would like to extend special thanks to my supervisors Dr. Carson Woo and Dr. John Tillquist for their considerable guidance and assistance on this research over the past few years, and to my external examiner Dr. Paul Kedrosky for his valuable suggestions.

I would also like to thank my parents and brother for their continual support, and Laurie for his understanding.

1.0 Introduction

Electronic Bill Presentment & Payment

With the increasing popularity of the Internet, on-line billing has generated a great deal of interest among merchants and banks. Since the 1980s, customers could pay their paper bills through home banking services by digital bill payment. However, it was not until recent years that electronic bill presentment came into existence [6]. Now electronic bill presentment is often provided together with electronic bill payment, and they are referred to as Electronic Bill Presentment and Payment (EBPP).

EBPP has become attractive to billers¹ and their customers. It provides billers advantages such as cost saving, reduction in billing cycles, and closer and more direct relationship with their customers. To customers, it is a convenient and time-saving method to view and pay their bills. Since EBPP was first introduced, several variations of it have appeared in the market. The most prominent models now are Biller Direct, Bank Forward, and Consolidator [1].

In the Biller Direct Model, the biller provides EBPP at its web site. Besides viewing a bill, the customer² can also pay for it electronically by credit card or bank account. The biller then forwards the customer's payment request to the appropriate financial institution (e.g., bank) to receive payment. In the Bank Forward Model, the bank is the EBPP service provider. Bills from various billers are available for viewing at the bank's web site and the customer can pay the bills with their account at this bank. In the Consolidator Model, the EBPP service is provided by the consolidator. A consolidator is a third party that presents bills from various billers to the customer and allows the customer to pay with various options. In this model, the customer can view bills from different billers as well as pay the bill with accounts from different banks, or

¹ Companies that issue bills to its customers on a regular basis (e.g., telephone and utilities companies).

² The terms "the biller", "the bank", "the customer", and "the consolidator" are used throughout this thesis to refer to the groups of billers, banks, customers, and consolidators, respectively.

with credit cards, depending on the consolidator [1]. Details of these models will be explored in later chapters.

Intelligent Agent

There are various definitions for the term “intelligent agents”, or “software agents”. In [5], they are defined as “software entities that have been given sufficient autonomy and ‘intelligence’ to let them carry out specified tasks with little or no human supervision. An agent interacts with an ever-changing environment whilst representing the interests of a particular owner.” In [3], the author defines them as “long-lived programs that perform some tasks on behalf of a user.”

In this research, we will use Wooldridge & Jennings’ definition of weak agency [9]. It defines an IA as a hardware or software-based computer system that possesses the four properties: autonomy (able to operate without human intervention), social ability (interact with other agents and humans), reactivity (respond to changes in the environment), and pro-activeness (act according to its goal).

Several years ago, IA was mostly used to automate repetitive or personal management tasks [4] because of their characteristics. In chapter 2, there will be a discussion on the numerous applications in the area of e-commerce that make use of IA. Examples include handling electronic mails, scheduling meetings, filtering electronic news, and recommending entertainment. IA in these applications saves its user time and painstaking efforts by taking care of routine operations.

Motivation

From the discussion above, it appears that the IA technology is a good candidate to handle EBPP because pulling bills from different web sites and paying these bills is a routine process that does not require making complicated decisions. Therefore, besides

handling electronic mails, scheduling meetings, and filtering electronic news, perhaps IA can also be programmed to retrieve bills and make payments for its user.

There are various ways of how an IA operated model can be structured. We propose that there will be one IA working for each user. Using the information supplied by the user, the IA could retrieve bill information from various billers and present the information to its user. It can also follow the user's instructions to pay bills by sending the necessary information to the appropriate banks. In some ways it acts like a consolidator except that there is one IA serving each user, and the service can be customized to suit each user's different needs.

Since the IA has been performing well in other areas as a personal assistant to carry out operations similar to those required for EBPP, it should also work well in a customer-oriented model to provide EBPP services to the user. However, such a model has not appeared in the EBPP business. Why are intelligent agents used widely in other areas but not in the EBPP business? **More specifically, what are the obstacles that prevent this model to exist in EBPP?** This is the question that we plan to answer in our research.

The answer to our research question may appear to be trivial as there are relatively few billers and banks compared to the vast number of individual customers. If we view the billers and banks as suppliers of bill information and payment services, and customers as consumers of these resources, there are many consumers but very few suppliers. The asymmetry between the two sides gives the suppliers (billers and banks) monopoly over the situation and they will not allow a customer-oriented model to exist. However, a deeper understanding on the situation will reveal that the circumstance is not as monopolistic as it seems. Besides the big billers and banks that are already offering EBPP, there are smaller billers and banks³ that also want to participate in the business. Therefore the degree of asymmetry between the two sides is not as big as one thinks, and

³ There are six banks in Canada, but there are many small banks in the United States

there is indeed an opportunity to bring together the customers to ask for a model that is oriented towards their need.

From the view of system analysis, we can examine the question by assessing the feasibility of implementing the project (i.e., the IA operated EBPP model). A feasibility analysis determines how likely a project will succeed and helps the management or researcher to decide if they should proceed with it. Although feasibility factors are dictated by the nature of a project, they are generally classified as follows: economic, technical, operational, schedule, legal and contractual, and political [28].

Economic feasibility examines the financial costs and benefits of implementing a project. It determines whether the project can be accomplished within the given budget and whether the benefits are worth the costs. For example, one would ask if it costs too much to build an IA application, and whether there is enough cost saving resulted from this application so that it is worth building [28].

Technical feasibility, on the other hand, evaluates the possibility to build the project with the existing technology [28]. Questions such as: “Is it too difficult to implement such an application?” and “Does a general user have the technical knowledge to use the system?” belong in this category.

The last four categories (operational, schedule, legal and contractual, and political) can be grouped into “organizational” feasibility [7]. Organizational feasibility considers issues such as the probability of solving the problem, the duration required implement the project, relevant legal issues, and the society’s acceptance. Possible faults of the IA Model in this area include: lack of trust from customers on IA, and laws that prohibit this kind of information flow.

In addition to the above, we would like to add the strategic feasibility. According to Sabherwal and King, a strategic application is defined as “one that has a profound effect on a company's success and destiny, by (a) influencing or 'shaping' the company's

strategy or (b) playing a direct role in the implementation or support of the company's strategy" [8]. In accordance with this idea, strategic feasibility would assess a project based on issues that affect the company's future such as the project's competitive advantage and its alignment with the company's strategy.

From experience, we think that it is technically easy to build such an IA system, and the financial cost should be relatively small. We will confirm our guess in the research by building a prototype of this model and estimating the time and cost involved. Due to the customer-oriented nature of the IA Model, we further think that it would work better than other available models from a customer's point of view. Therefore, we hypothesize that the main barrier lies in the strategic feasibility. More specifically, we believe that some key players involved (e.g., banks and billers) are not supportive of this model because it is not to their own benefit to do so. We will use the Dependency Network Diagrams to examine each model and analyze the results. By doing so, we hope to answer our research question by identifying obstacles to use the IA Model in EBPP, and if these obstacles can be solved.

Examples similar to our case have occurred in other businesses. There have been innovations that fail to succeed even though they work well for a specific group in the society and are able to solve problems that its alternatives fail to solve. Therefore, besides the performance of the innovation, there are obviously other factors that researchers and managers in MIS need to review in order to assess the workability of the innovation. The answer to our research question will provide this group the strategic issues that they need to pay attention to when designing new technologies, and a means to make their designs appropriate for a given situation.

The main difference between this research and the existing literature on EBPP [6, 10, 29, 32, 33] is that most of the existing ones focus on the advantages and impacts of this service on the banking industry. Comparisons are usually made between particular service providers (e.g., Royal Bank, CheckFree), and on functional issues such as charge of service, popularity, and ease of use. This research, however, will concentrate on

comparing models (e.g., Bank Forward vs Biller Direct) using strategic and organizational factors such as dependency, role, goal, and norms, and will bring a new insight on the subject.

Thesis Outline

The thesis consists of six chapters and will be divided as follows:

Chapter 2 reviews basic concepts of EBPP and gives detailed descriptions on the three existing models. A brief overview on IA applications in e-commerce is also presented.

Chapter 3 begins with an introduction to the architecture of an IA Model of the EBPP in order to give the reader a general idea of how such a model could be constructed. It then discusses the design and implementation issues of our prototype and provides basic technical information such as data flow diagrams and flow charts on significant processes and performance on test scenarios. The intention of providing this information is to prove that such an application is indeed technically feasible and works well for the user. This chapter will conclude with giving an estimate on the time and financial cost of implementing the remaining parts of the system (other than the prototype). The objective of this chapter is to verify our guess that implementing an IA EBPP model is both technically and economically feasible.

In Chapter 4, theories on Dependency Network Diagram (DND) will be presented. DND is a key tool used in this thesis to analyze the dependency relations between different roles in each model. All four models will be represented by DNDs and the dependencies among roles will be analyzed. The objective of this chapter is to illustrate how the roles vary in different models in areas of goal, dependency, governance control, and actions.

Chapter 5 analyzes the results from chapter 4 and compares all the models. The objective of this chapter is to identify the shortcomings of the IA EBPP model, investigate the difficulties to overcome these shortcomings, and explain why this model does not exist. In other words, this chapter answers the research question.

Chapter 6 concludes this thesis by reviewing its contents, contributions, and limitations.

2.0 Related Works

2.1 *EBPP and Existing Models*

As mentioned in chapter 1, Electronic Bill Presentment & Payment (EBPP) consists of two key parts: bill presentment and bill payment. We define an EBPP application as one that allows the customer to view their bills and pay for them over the internet.

Depending on how these actions are performed, EBPP can be grouped into different models. In this section we will examine the three most common models: Biller Direct, Bank Forward, and Consolidator. The objective of this section is to give the reader a basic understanding of each model to prepare them for our analysis in later sections.

Biller Direct

In the Biller Direct Model, the EBPP is provided directly by the biller to its customers. The next figure is a simplified view of how the bank, biller, and customer relate to each other, where the arrows denote direct interactions such as bill presentment and payment request. A double-ended arrow indicates that actions between the two actors go in both directions, and a one-ended arrow indicates one way actions, in the same direction as the arrow. For example, the double-ended arrow between the bank and the biller represents the biller's action to send payment authorizations to the banks, and the bank's action to send payments to the biller. The interactions among the parties are not revealed on the diagrams in this chapter because they will be described in section 4.3.

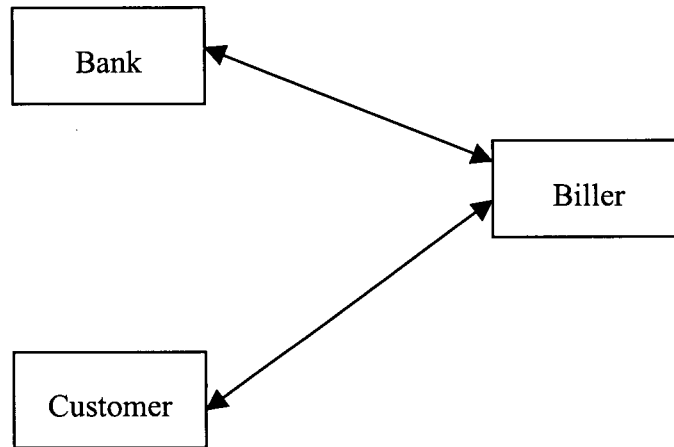


Figure 1. Simplified Relationship Diagram in the Biller Direct Model

As shown in the diagram, there is no direct interaction between the bank and the customer and all interactions go through the biller. In this model, the biller is the party that develops and provides the EBPP service. Customers must sign an agreement with the biller to enroll into the service. This agreement outlines conditions such as service charges, penalty on late payments, and payment methods [37]. After signing this agreement and providing other relevant information such as the preferred method of payment (e.g., credit card or direct debit from bank accounts) and bill account information, the customer can access their billing information any time by entering their user identification and password. Besides viewing bills, the customer can also pay for them at the web site [5]. These actions are represented by the double-ended arrow between the biller and the customer.

If the customer decides to make a payment, the biller acts as the payment originator and forwards the payment authorization to its payment provider (e.g., bank). The biller's payment provider then requests payment from the customer's financial institution⁴. After validating the authorization forwarded by the biller, the customer's bank then deposits the specified amount into the target account (the biller's) [1]. The

⁴ The Financial Institution can be a bank or a credit card company, depending on the payment method. To keep it simple, we will only consider banks for this research, the same assumption applies to the biller's payment provider.

actions between the bank and the biller are represented in the above diagram by the double-ended arrow.

Bank Forward

In the Bank Forward Model, the bank acts as a central service provider that consolidates bills from various billers and presents them electronically to the customer [1]. Instead of viewing bills from only one biller at the web site, the customer can view bills from different billers at once. The following figure shows the relations among the bank, the biller and the customer. It can be seen that the bank now takes the intermediary position and there is no direct interaction between billers and customers.

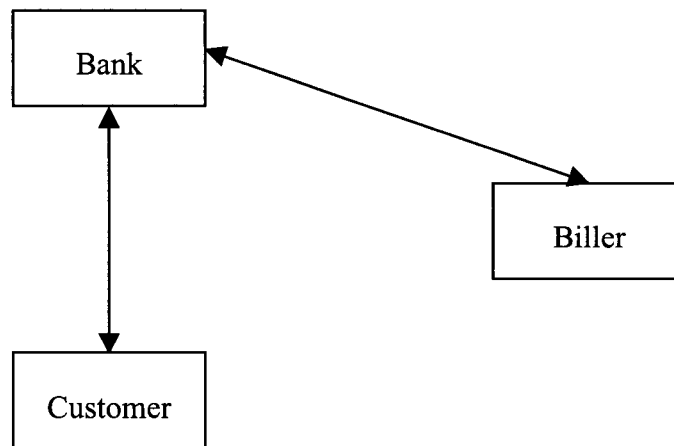


Figure 2. Simplified Relationship Diagram in the Bank Forward Model

In this model, the bank is the party that develops and provides the EBPP service. It partners with billers who want their bills to be presented electronically. In most cases, the bank dictates over the details of the EBPP process such as data format, display style, and details to display. The biller pays the bank for its services, and the amount to pay usually depends on the number of bills to be presented. In general, a biller can partner with more than one bank to display its bills [1].

To use this service, the customer must have an account with the bank. As in the Biller Direct and other models, they need to sign a service agreement with the bank and provide relevant information before using the service [36].

When a bill is issued, the biller sends it to the bank, which in turn presents it to the customer in the fashion agreed upon both parties. As in the Biller Direct Model, the customer can access the bill information by entering the required information (e.g., user ID and password). In most cases, the bank has the full details of the bill and there is no direct contact between the customer and the biller when the customer views the bill. The biller's action of providing bill information and the bank's action to pay the biller are represented by the double-ended arrow between them.

Besides viewing the bill, the customer can also execute payments by authorizing a withdrawal from their bank account. Some banks (e.g., Chase On-Line [42]) even enable the customer to pay any biller or individual directly from their bank account by supplying the recipient's bank account number. By supplying the required information such as name of payee, amount to pay, and account to pay from, the customer gives the bank authorization to withdraw funds and pay on their behalf. The double-ended arrow between the bank and the customer represents the bank's action to present bills to the customer, and the customer's actions to request services and supply information.

Consolidator

In this model, the consolidator is the party that develops and provides the EBPP service. Interested billers and banks agree to comply with terms such a format of data transfer and amount of details to present, which are controlled by the consolidator. As in the Bank Forward Model, each biller and bank involved can partner with more than one consolidator [1]. The following diagram shows the intermediary position of the consolidator.

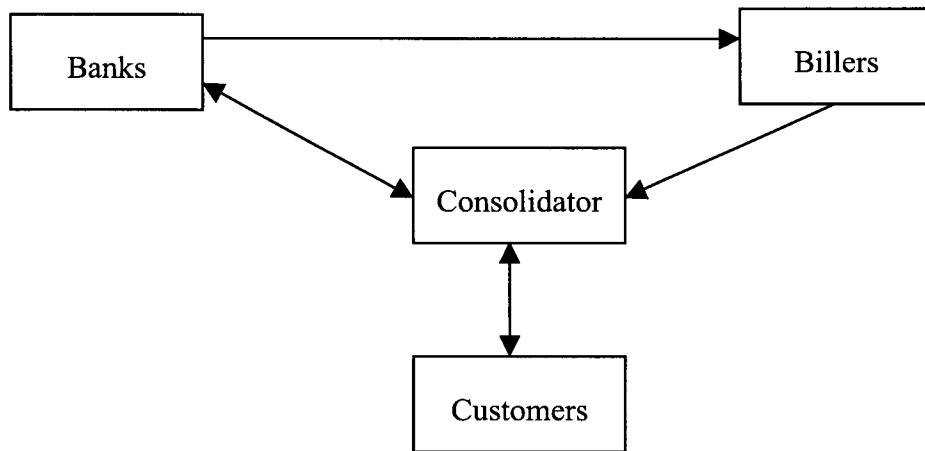


Figure 3. Simplified Relationship Diagram on the Consolidator Model

A customer enrolls in the service by signing a service agreement similar to that in other models, and providing the relevant information. After the enrollment, the customer's bill information is made available on-line for viewing. When a bill is issued, the biller sends it to the consolidator, which presents it to the customer when they log into the service. Like the Bank Forward Model, full details of the bill reside with the consolidator, there is no direct contact between the customer and the biller when the customer views the bill [1]. Because the consolidator usually does not send anything back to the biller, there is only a single-ended arrow between itself and the biller. Besides viewing the Bill, the customer can also execute payments with their accounts in different banks. Although only the bills from certain billers can be viewed, many consolidators (e.g., Checkfree) also let the customer pay anyone with this service by

providing the payment recipient's bank account number [21]. To initiate a payment, the customer provides the consolidator with an authorization, which is then forwarded to the customer's bank⁵. After verifying the request, the bank transfers the specified amount into the biller's bank account, which may be in a different bank. These actions are depicted by the double-ended arrow between the customer and the consolidator.

2.2 *Intelligent Agent in E-Commerce*

As we have defined in the introduction, an intelligent agent (IA) possesses the four characteristics: autonomous, social, reactive, and pro-active. It will be shown in the next chapter that the IA in our prototype of the EBPP application satisfies these four characteristics in the following way:

- **Autonomous:** It is able make decisions according to its rules database and make suggestions to the user. It can carry out delegated tasks without human supervision.
- **Social:** It interacts with banks and billers via some kind of communicative language to retrieve information and make requests for payments.
- **Reactive:** When the situation changes and something needs to be done (e.g., a withdrawal is made suddenly and the balance of the account is not enough to cover another payment), it responds by taking a different action according to what it "knows" in its knowledge base.
- **Pro-active:** Besides passively acting in response to the environment (e.g., due date of a bill, balances in accounts), it also takes initiative to act according to its goal and thus shows goal-directed behavior (e.g., suggesting the user to pay with another account if doing so is cheaper).

Wooldridge & Jennings identifies three key issues on intelligent agency: agent theories, agent architectures, and agent languages. In simple terms, agent theory specifies

⁵ There are some consolidators that provide accounts to customers in which they can store money and use them to pay for their bills (e.g., CheckFree, Transpoint), but this method is not as prevalent as the one we discuss in this section and in general the bank is the actor that performs the fund transfer in a Consolidator model [1, 40].

how to conceptualize and represent the properties of an agent, agent architecture defines how to implement the agent and addresses issues such as software and hardware structures, and agent languages are the programming languages used to build the agent. The topic on agent languages deals with how to effectively build and run these agent programs [9]. Because our focus is not to explore the development of IA, we will not go into details of theories and languages on intelligent agency. In the next chapter, we will discuss the architecture of our relatively simple intelligent agent system and the technologies required to build it. In the rest of this section, we will take a brief look at IA applications in e-commerce. The objective is to show that IA is indeed being widely used in the area of e-commerce to handle tasks that are in some ways similar to the EBPP process. Therefore the idea of using IA in EBPP being completely ignored is not a pure coincidence.

Intelligent agents are suitable to act as mediators in e-commerce because of its personalized, continuously running, and autonomous nature. They can automate time consuming stages of the buying process and optimize the cycle of on-line purchase [4]. There are many opportunities to use intelligent agents in e-business applications. In her article, Maes lists several applications where IAs are used to handle consumer behaviors that involve information filtering and retrieval, personalized evaluations, and time sensitive interactions. These agent systems are PersonaLogic, Firefly, Tete a Tete, Bargain Finder, Jango, Kasbah, and Auction Bot [4].

In a multi-agent e-business environment, there are mainly four types of agents. An application agent has expertise in a specific area. A large number of them each holding different expertise are formed into a network and work with each other to solve a complex problem. The “integrated value chains” phenomenon where companies with common interests and related abilities work together is an example. Personal agents work directly with users to support their work. They make recommendations by monitoring and learning the user’s behaviors. Examples of personal agents include intelligent tutoring systems and Web browsing assistants. General business activity agents perform common functionalities for commerce activities. Negotiation agents

between buyers and sellers, billing agent, and marketing agents are examples of this kind. The fourth type is the brokering agents, which are also referred to as matchmaking agents. They keep information about other business agents and help users to publish their services or find services [22].

Since IA technology is being used in various areas and there are existing applications in the market that perform similar functions as the bill retrieval and bill payment actions, applying the agent technology in the EBPP area is therefore a reasonable approach. In the next chapter, we will present the architecture and implementation on the IA EBPP prototype we built for this research and discuss how it handles EBPP tasks.

3.0 The Intelligent Agent EBPP Model

In this chapter, we will begin with a presentation on the IA Model structure and discuss how the roles (bank, biller, user, and IA) are organized. Then the architectural and implementation issues on our prototype will be examined. The objective of this chapter is to confirm our hypothesis in the introduction:

- It is technically and economically feasible to build an IA EBPP Model.
- Functionally, an IA EBPP model is not inferior to the other models in the market.

One may think that because IA has been used in many areas of e-commerce to handle operations similar to those in electronic billing, there should be no technical difficulties to implement the IA Model from the vendor's point of view. However, it is necessary to build a prototype to assess the technical feasibility of IA Billing because there are issues that cannot be ascertained before doing so. First, we have discussed that one attribute of IA Billing that makes it superior to its alternatives for a user is its decision-making ability. However, as the number of biller and bank accounts increases, IA's job to make a decision will be more complex, and technical problems may occur. Second, in order for IA Billing to be widespread, it must show satisfactory performance on a personal computer (PC) used at most households, which does not have a vast amount of memory nor a very high speed. Section 3.1 will address these concerns.

3.1 *Overview of Model*

Contrary to the models discussed in chapter 2, the IA EBPP Model has not come into existence. We are only suggesting one possible structure of the model based on our knowledge on IA and EBPP.

We assume that the model can work as follows: First, a third party vendor or service provider develops an IA EBPP application and persuades banks and billers to

support it by agreeing to provide the necessary resources. A third party is selling this application because the product may be biased towards the bank or the biller if either of them was the service provider, and the model would not be customer-oriented. No assumption is made on the identity of the vendor so that all possibilities are considered. Candidates include software development companies and individual investors. By accepting the IA EBPP model, banks and billers agree on the format and method of the data transfer and recognize requests made by IAs. After obtaining acceptance from enough banks and billers (otherwise the model cannot provide enough convenience to attract customers), the vendor sells this application in the market. The user⁶ of this application would be charged a monthly fee, which may include the connection cost and service charges from banks and billers (depending on the arrangement among the parties involved), and possibly need to pay for the software installation, depending on the marketing approach of the vendor.

Before using the service, the user is required to sign agreements with billers and banks. Although the vendor of the IA application may manage to ask for some favorable conditions, it is very likely that this agreement would be entirely laid out by the banks and billers. In order for the IA to operate properly, the user also needs to feed it with necessary information such as URLs and account numbers of banks and billers, and personal preferences so that the application can fit the particular needs of each user.

In order to present bills, the IA first visits each biller to retrieve bill information. This information is then stored in the database and displayed according to the user's preferences. When the user decides to make a payment, the IA follows the instructions and sends the request to the appropriate banks. Besides following instructions, the IA is also able to make suggestions based on its "intelligence", which depends on its design. When the bank receives a payment authorization, it verifies it and then transfers the amount from the user's account into the designated account (e.g., the biller's). Because

⁶ The term "user" is used instead of "customer" in the IA Model because it gives a more accurate description of the relationship with the IA, but it should be noted that they refer to the same entity because customer to the bank and biller is the user of the IA.

the IA is goal oriented, it remains “unstable” until the delegated task is completed. For example, if for some reasons the payment cannot be executed, the IA will keep trying (by repeating the task or by attempting other alternatives) until the payment is made or cancelled by the user. The following diagram summarizes what we have discussed and shows a simplified structure of the model.

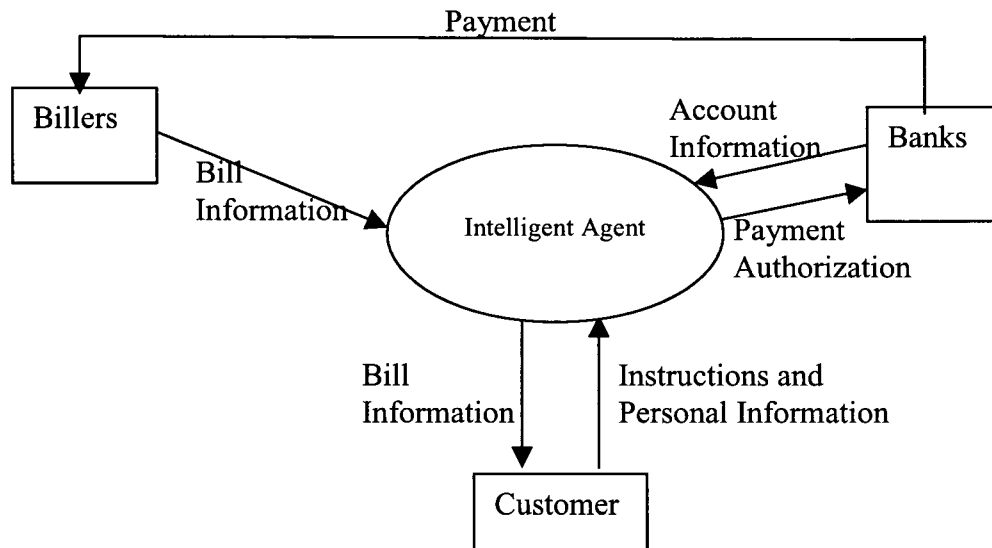


Figure 4. Overview of Intelligent Model in Simplified View

It can be observed that the IA Model is similar to the Consolidator Model in the sense that the IA takes an intermediary position to interact with individual billers and banks to retrieve bill information and forward payment requests so that the user does not have to visit multiple web sites. However, there are a number of significant differences between these two models:

- For one given IA or consolidator, each IA serves one user (one-to-one relationship), but one consolidator serves many customers (one-to-many relationship).
- The IA has the goal of working for the benefits of its user, but a consolidator tries to balance the benefits of all parties.
- The IA can be customized to suit each user’s unique needs, but the consolidator has control over most of the settings.

3.2 *Architecture and Implementation*

There are many ways to implement the IA EBPP application. In terms of the IA's location, it can either reside in the user's machine or at a remote server. In the first case, the user only needs to connect when data is being retrieved or sent, and the IA can perform the rest of the work off-line. In the second case, however, the user must be connected to the remote server (of the service provider, which is the vendor of the application) to perform any action, such as viewing previously retrieved bills. It will become apparent in later parts of this thesis that the location of the IA does not affect our analysis. We have built our prototype with the IA installed on the user's machine simply because it is easier to do so with the available resources.

The IA's behaviors are determined by the program design of the application. Our prototype is only one of the many possible designs. It is very simple because our goal is to demonstrate how an IA Model works in practice. Additional functions can be added to the application by simply changing the structure of the database or altering the design of the program. After introducing the concepts and techniques involved in implementing the prototype, we will also show the test results at the end of this section to give a clear picture of how the IA carries out its duties.

The next diagram shows the structure of our design. It is a Data Flow Diagram of figure 4 in the section 3.1, with the IA (indicated in the dotted box) broken down into its sub-components. As shown on the diagram, the IA consists of three main parts: user interface, execution module, and knowledge base. The prototype we built only includes the parts that reside on the user's machine because we only want to demonstrate the local operations of the IA (i.e., operations in the user's machine). The interactions with the bank and biller to retrieve information and request payment would be too complicated to implement for the purpose of this thesis. There are existing technologies (e.g., EDI) that enable information exchange between banks and companies and they can be used for the same purpose in our model. This issue will be visited again at the end of the section.

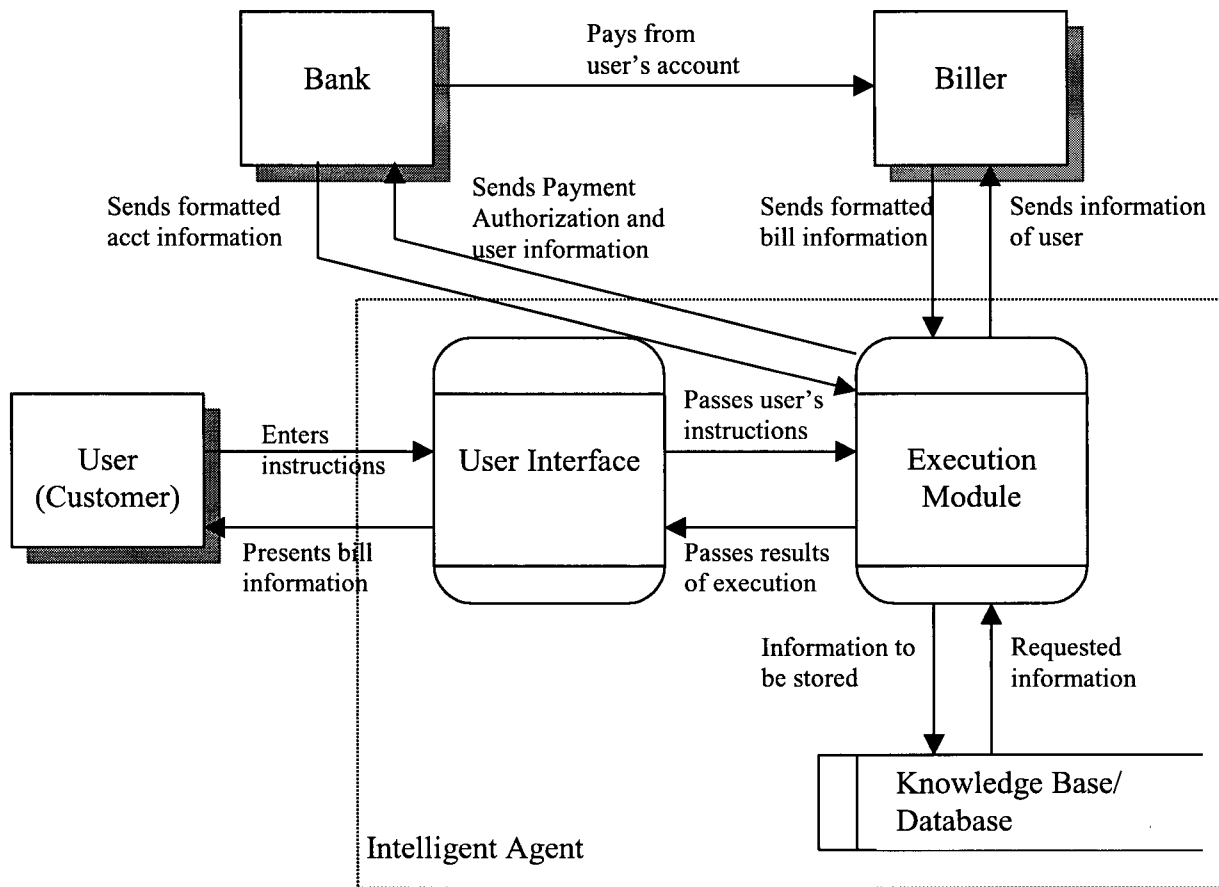


Figure 5. Overview of Model (IA broken down into sub-components)

Interaction between the user and the prototype are made via the user interface component. Various browsers can be used, depending on one's preference. We have chosen Internet Explorer 5.0 for the sake of convenience. When the user starts the EBPP service, they must first log on via the user interface. The information entered is then passed to the execution module.

The execution module is a computer program. When it receives the instructions, it checks the database and determines the appropriate actions to perform (e.g., stores the information in the database, initiates a payment). It communicates with the bank and biller to retrieve information or initiate service requests. The execution module should be designed to accommodate functions available in commercial applications and address relevant technical issues such as computational complexity. Most home PCs have neither a vast amount of memory nor a very high speed, the program therefore should not store a

large amount of information nor take a long time to execute. A number of alternatives can be chosen to program the IA. We have used Microsoft InterDev for this research because it is designed to work with multiple databases for Internet applications and therefore is suitable to handle the data centric EBPP operations.

The third component of the IA is the knowledge base. It stores the information used by the IA to perform its tasks, such as URLs of banks and billers, account numbers, bill information, payment history, and user preferences. The execution module stores information into this database and queries it when it needs to make a decision or perform an action. Microsoft Access is used in our prototype because it works well with Microsoft InterDev, but it can be replaced by other database applications. In our prototype, the knowledge base is composed of tables. Before discussing the content of each table, we will first explain the relationships among the entities in the knowledge base.

Entity Relations in Knowledge Base

An entity is an object for which data are to be collected and stored, and a relationship describes how two entities are connected [38]. In our model, it is only necessary to store data obtained from banks and billers, and these entities can be further classified into bank accounts, biller accounts, and bills. These entities are depicted in the Entity-Relationship (ER) diagram in figure 6. The user is not displayed on the diagram because each IA serves only one user, and therefore all the entities are related to the same user.

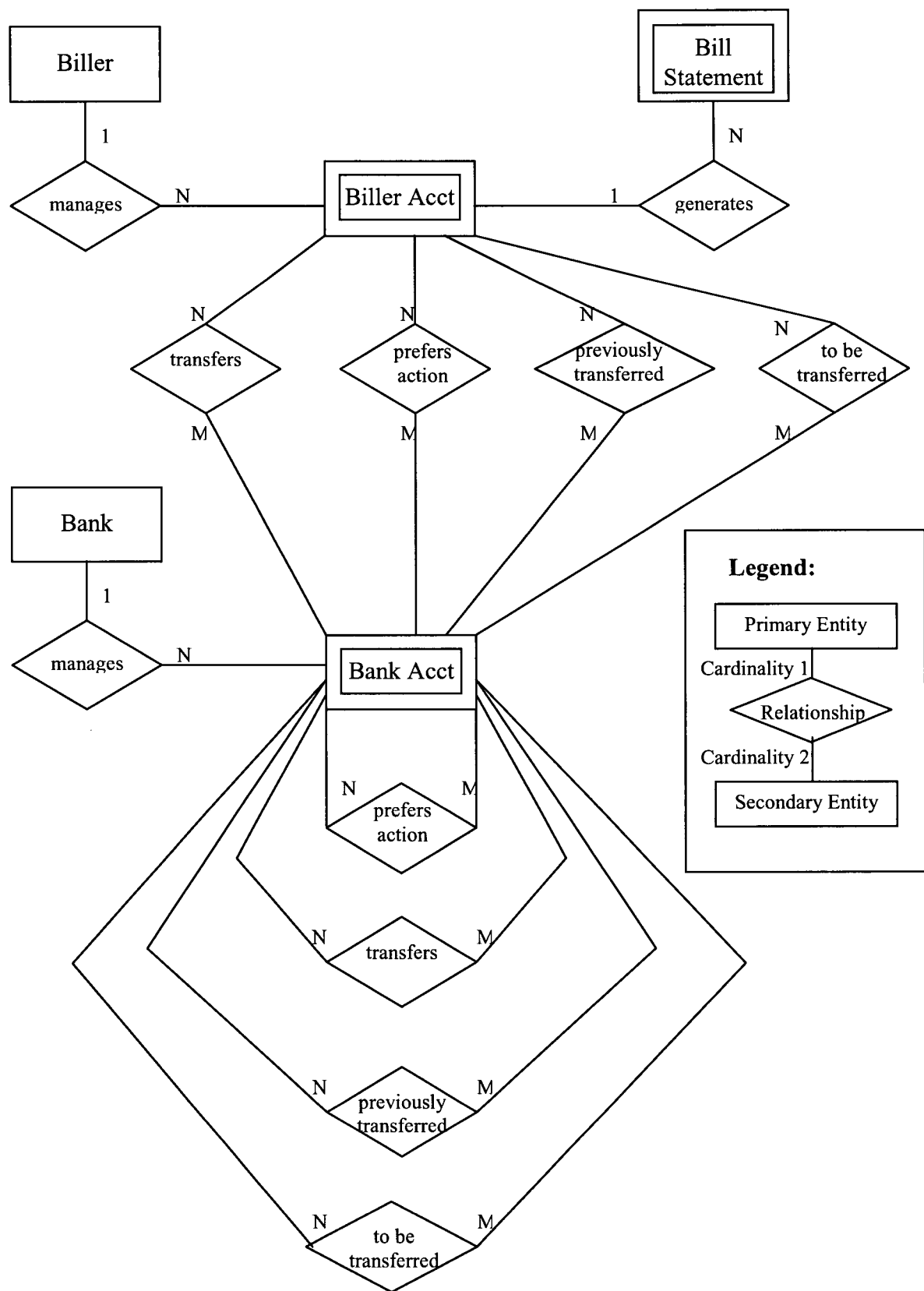


Figure 6. E-R Diagram for Tables in IA's Knowledge Base

The entity *Biller* lists all the billers that send bills to the user. The information stored for a biller are its name and its contact information (e.g., URL), which are required for the IA to retrieve bill information.

Biller (BillerName, BillerContactInfo)

Because a user can have more than one account with each biller (e.g., a household with more than one telephone number), the entity *Biller Acct* is needed to store the information for separate accounts, which includes the biller's name, the account number, and the account information (e.g., PIN) of the user.

Biller Acct (BillerName, BillerAcctNum, PIN)

The entity that stores the bill information for an account is *Bill Statement*. It stores the information of the bill account as well as other information needed to pay a bill such as the due date, the amount due, and description of the purchase.

Bill Statement (BillerName, BillerAcctNum, IssueDate, DueDate, AmtDue, MinPay, Description)

where *MinPay* is the minimum amount to pay by the due date.

The entity *Bank* lists all the banks that the user has accounts in. The information stored for a bank are its name and its contact information (e.g., URL, telephone number).

Bank (BankName, BankContactInfo)

Because a user can also have more than one account with each bank (e.g., checking, saving accounts), the entity *BankAcct* is needed. It includes the bank's name, the account number, and the account information of the user (e.g., PIN).

Bank Acct (BankName, BankAcctNum, PIN)

A bill statement (generated by a biller account) is paid by transferring the required amount from a bank account into a biller account. This is described by the *transfers* relationship. The contents of this relationship include the time and cost of the transaction from a bank account into a biller's account:

Transfers (Time, Cost).

The relationship *previously transferred* stores transfers made in the past, a record is added to this table every time a transaction is performed:

Previously Transferred (DateOfTransfer, TransferAmt)

ToBeTransferred contains transfers that have not been completed due to reasons such as insufficient fund in the source account. Every time the customer logs on, the IA checks this table to see if current conditions allow it to carry out any of these pending tasks. If a task in this table could be completed, the IA performs the task and removes it from the table.

To Be Transferred (Constraint)

As discussed, the user can enter their preferences and select a default bank account to pay a certain biller account in the case when the IA needs to make a decision (e.g., user does not select a bank account when initiating a payment instruction). The relationship *Prefers Action* describes this condition.

Prefers Action (Case, Action)

where *Case* describes the circumstance that invokes this preference (e.g., insufficient fund), and *Action* describes the action to be taken under this circumstance (e.g., wait, default account).

Because a transfer can be made between two bank accounts as well as between a bank account and a biller account, we have the same relationships *transfers*, *previously transferred*, *to be transferred*, and *prefers action* for bank accounts.

Cardinality reveals the number of entity occurrences with another related entity. The cardinality between a biller and a biller account is one to many (1:N). Similarly, more than one statement can be generated by each biller account and therefore there is a one to many (1:N) relationship between the two. Because transfers can be made from various bank accounts into a biller account and vice versa, this relationship is also many to many (N:M). The relations *previously transferred* and *to be transferred* are also many to many under the same reason.

The relationship *prefers action* between a bank and a biller account is also many to many (N:M) because a bank account can be selected as a default account to pay a number of biller accounts and there can be more than one default bank account to pay a biller account under various conditions (e.g., pay with checking account if it is the first week of the month, pay with saving account if it is the last week of the month).

Because similar conditions apply to transfers between bank accounts, the relationships between them have the same cardinalities.

Tables in Knowledge Base

The information collected are grouped and stored into nine tables in the knowledge base: *Biller*, *BillerAcct*, *BillStatement*, *Bank*, *BankAcct*, *Transfers*, *ToBeTransferred*, *PreviouslyTransferred*, and *PrefersAction*. It can be easily shown that these tables are normalized, meaning that there are no data redundancies in them. They are listed below with their attributes, and the keys are underlined.

Biller (BillerName, BillerContactInfo)

Biller Acct (BillerName, BillerAcctNum, PIN)

**Bill Statement (BillerName, BillerAcctNum, IssueDate, DueDate, AmtDue,
MinPay, Description)**

Bank (BankName, BankContactInfo)

Bank Acct (BankName, BankAcctNum, PIN)

There are two sets of the following tables, one for billers, and another one for banks.

Transfers (TransferFrom, TransferTo, Time, Cost)

Previously Transferred (TransferFrom, TransferTo, DateOfTransfer, TransferAmt)⁷

To Be Transferred (TransferFrom, TransferTo, Constraint)

Prefers Action (TransferFrom, TransferTo, Case, Action)

These tables can be classified into two groups: one group contains information required by the IA to operate (*Biller*, *BillerAcct*, *Bank*, *BankAcct*, *Transfer*, and *PrefersAction*), and another one keeps track of the details of the operations (*BillStatement*, *PreviouslyTransferred*, and *ToBeTransferred*). Contents of the first group are entered by the user and are not changed when an operation is executed. The second group, on the other hand, holds information that is being updated constantly by the IA as it works.

⁷ It is assumed that there are no more than one transfer made between two specific accounts in one day, which is true in a general case.

Execution Module

The execution module of our prototype provides four main selections as shown on the next diagram: add/remove bank or biller, set preference, pay/transfer, and view. Viewing and paying bills are basic functions that are available in any EBPP application [1], and must also be provided by the IA application. Because a major component of the IA is a knowledge base, there must be a means to input and update its contents. The selections “add/remove bank or biller” and “set preference” enable the user to do so. A real application may have more functionalities to serve the complex needs of the user, but ours only provide the basic ones.

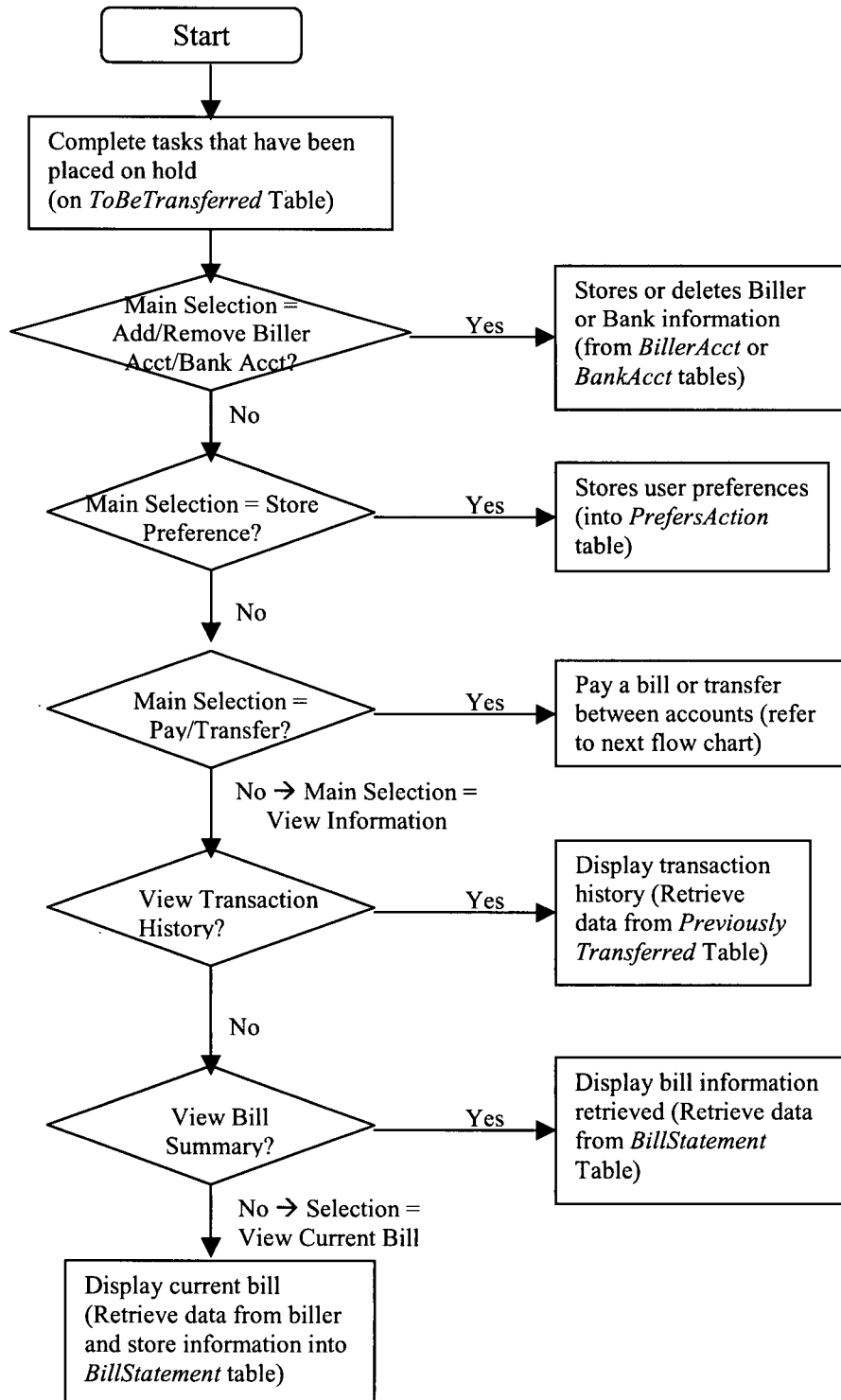


Figure 7. Execution Module of IA in Sub-Components

We mentioned earlier that an IA application could work better as a personal assistant to the user than commercially available products. The actions to add or remove a bank or a biller, set preference, and view bill information are simple operations that do not require much “intelligence” from the IA and thus will not be discussed. The pay/transfer option, on the other hand, involves more complicated decision-making abilities. For example, if the user wants to pay biller X with their account in bank Y, but bank Y cannot make a direct transfer into the account of biller X, the payment cannot be completed. In existing applications (e.g., Royal Bank On-Line, CheckFree), the bank or service provider (depending on the model) may notify the user a few days after the payment request is initiated that the payment cannot be completed because there is insufficient fund in the user’s account. In an IA EBPP application, however, the IA can immediately inform the user of such a situation and suggest alternative solutions. The IA can also advise the user of better alternatives to complete a task and saves them time and cost.

The following flow chart depicts the sequence of a pay/transfer process. It reveals the logical design of the computer program and helps the reader to determine how the IA reacts under various circumstances.

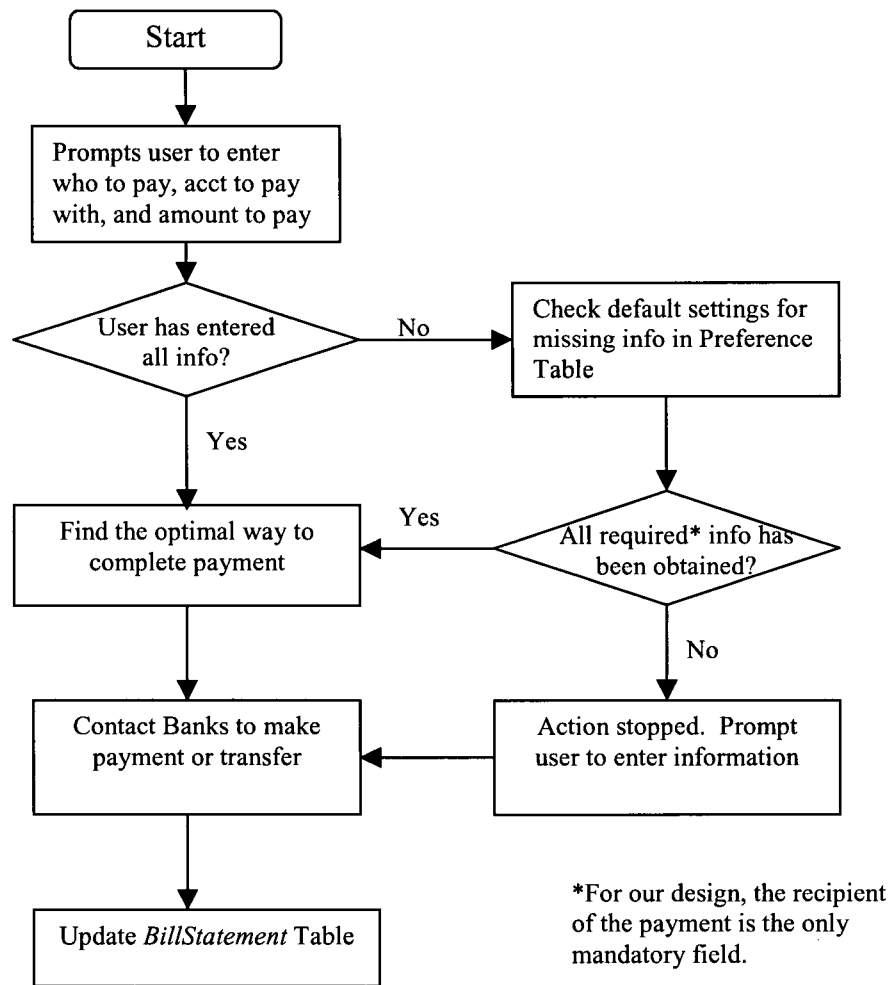


Figure 8. Flow Chart for Pay/Transfer

As indicated in the above flow chart, the IA checks the *PrefersAction* table for default settings when any information is missing (e.g., amount to pay, account to withdraw payment from), and asks the user if no default value is available. Before carrying out a task, the IA also looks into tables *Transfers*, *Bank*, *Biller*, and *BillStatement* table to answer questions such as:

- Can a direct deposit be made from this bank to this biller?
- How long does it take to make a transfer from bank A to that bank B?
- Is the balance in this account sufficient to cover the transfer?
- Does the user have a preference to pay with a certain account?

With these information the IA generates a solution that best fits the user's needs with the least costs in time and charges, which is defined as the "best solution"⁸. We have made the restriction that a payment should require no more than two transfers (i.e. involve no more than two bank accounts). Although in rare cases one may want to make a payment by transferring among a number of accounts to reduce the time required, or for confidential reasons, it is generally inefficient to do so because the fees will be expensive and it will be confusing for the user to manage their accounts. However, this restriction can be modified easily by the program designer to allow more transfers when necessary.

Computational Complexity

As mentioned in section 3.0, one objective of this chapter is to assess the technical feasibility of the IA application. To do so, the running time of the operations, which can be determined by the computational complexity of the algorithms, must be considered. As discussed earlier, "Pay/Transfer" is the most complicated main operation. In this operation, the algorithm to generate the best solution is the most complex and time consuming. Therefore we will examine the computational complexity⁹ of this algorithm to determine if there is any time constraint problem.

In our prototype, a "solution" is a list of one biller preceded by one or two banks in the order of the transfer (because of the restriction). For example, the solution "A, B, X" means transferring from bank accounts A to B, and then from B to biller X. In order for the solution to work, there must be enough balance in the source bank (i.e. bank A in this case) to cover the payment, and transfer should be allowed from A to B, and from B to X. The reason that we need at least one bank and one biller is because our goal is to pay a biller with a bank account, and transfer between bank accounts is necessary if there is not enough balance in the bank account that can directly pay the biller. From all the

⁸ The best solution in our design is not the optimal solution because an optimal solution in this case would result in the shortest time required to complete a payment and may involve more than two transfers

⁹ The computational complexity of an algorithm can be measured in terms of elementary operations (additions and comparisons) required at the worst case scenario [35].

possible solutions, the one that satisfies the user's preferences best while requiring the minimum time and cost to complete is selected as the best solution.

The algorithm to generate the best solution is represented by the next flow chart. If the user specifies all the details to carry out the payment by entering information (the biller to pay, the account to pay with, and amount to pay) on the pay/transfer instruction, or by setting preferences, the IA tries to follow these settings as close as possible. When these specifications cannot be executed due to various reasons, the IA makes suggestions and executes them upon approval of the user.

The most time consuming parts in the algorithm involve looping through one or more array(s) of selected bank accounts in the *BankAcct* table. The flow chart shows that there are nine places where this can occur (numbered from ① to ⑨), and their computational complexities are listed in the next table.

	Step	Computational Complexity
①	Find all accts that can direct pay biller and store them in array	n (number of entries in <i>Bank Acct</i> table)
②	For each acct in array, transfer time = time to direct pay biller + time to transfer from user specified acct	m (number of entries in the array generated by step 1)
③	Best Solution = solution that has the smallest transfer time	m (number of entries in the array generated by step 1)
④	Can any bank acct direct pay biller?	n (number of entries in <i>Bank Acct</i> table)
⑤	Find accts in array A that carry enough balance to cover payment, put in array B	m (number of entries in the array generated by step 4, which is the same as the one generated by step 1)
⑥	For each acct in array A, find another acct that has enough balance to cover payment AND takes least time to transfer into this acct. Find the solution that takes least time to complete	m (number of entries in the array generated by step 4) $\times n$ (number of entries in <i>Bank Acct</i> table) $\leq n^2$ (because $m \leq n$)
⑦	Select acct in array B that takes least time to pay biller, set minAcct = this acct, minTime = time for this acct to pay biller	l (number of entries in the array generated by step 5)
⑧	Find accts in array A that takes less time than minTime to pay biller, put in array C	m (number of entries in the array generated by step 4)
⑨	For each acct in array C, find another acct that has enough balance to cover payment AND takes least time to transfer into this acct. Find the solution that takes least time to complete => TempMinSoln	p (number of entries in the array generated by step 8) $\times n$ (number of entries in <i>Bank Acct</i> table) $\leq n^2$ (because $p \leq n$)

Table 1. Computational complexity of various steps

The table shows that the most complicated operations occur in steps 6 and 9 and their computational complexities are quadratic, in the order of n^2 . An algorithm with quadratic complexity is deterministic. For a 1 MHz computer, which is slower than most home PCs, the computation time required to complete this operation is 0.0025 second as n grows to 50 [35], which is still a reasonable duration. It is very unlikely that n , the number of bank accounts of a user, can reach 50, therefore we can conclude that no time constraint problems should arise.

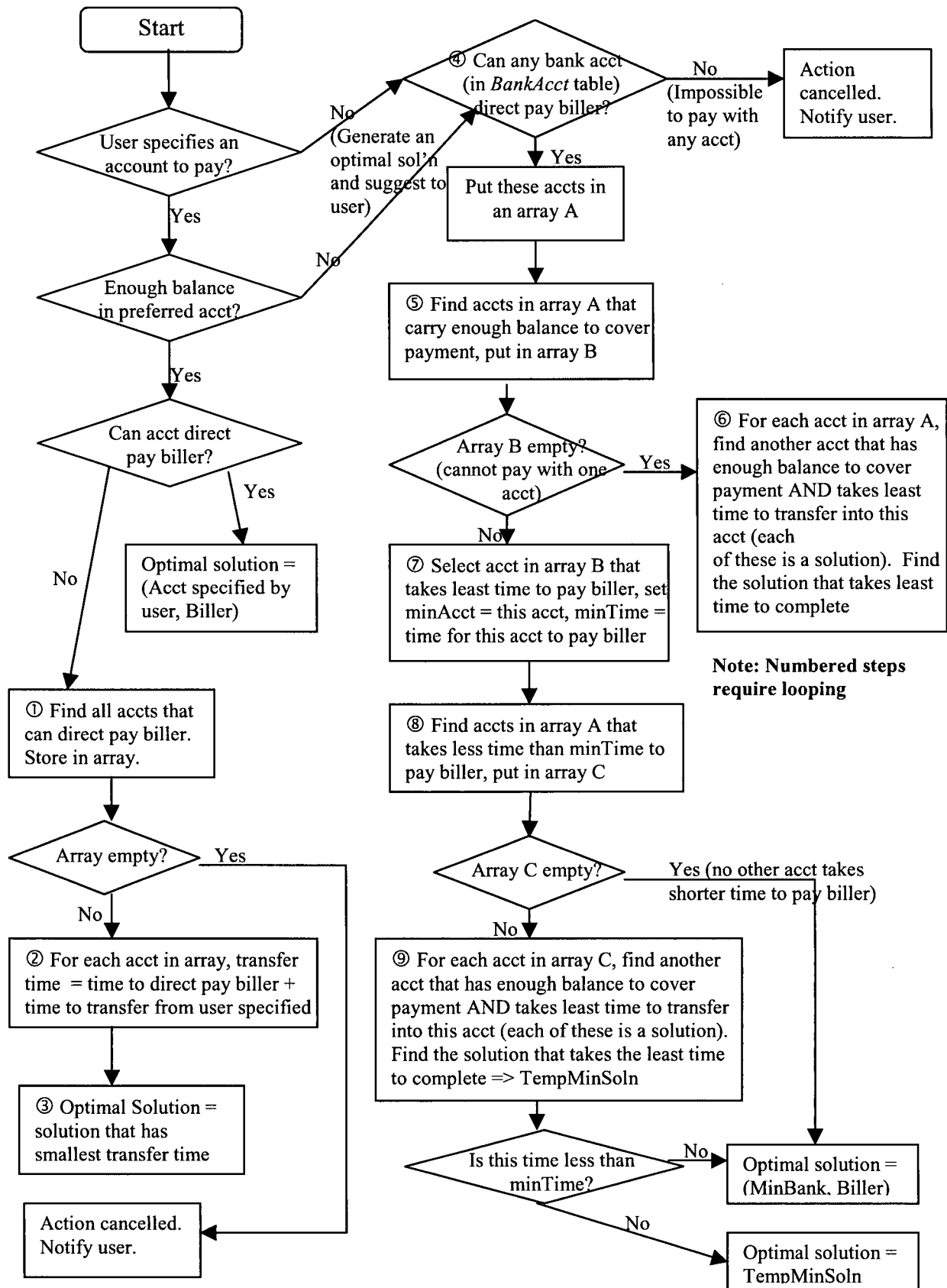


Figure 9. Flow chart to generate best solution

Functionality Comparison on Models

After discussing the operations of the prototype, we can compare the functionalities available in our IA application and the commercial products. The next table lists the functions offered by our prototype and those by Royal Bank (Bank Forward), CheckFree (Consolidator), and Telus (Biller Direct) [36, 37, 21].

Features	Royal Bank	CheckFree	Telus	IA Prototype
Making a payment from a bank account or an account with the service provider	Y	Y		Y
Making a transfer between bank accounts in the same bank	Y			Y
Making a transfer between bank accounts in different banks		Y		Y
Making a payment with a credit card		Y	Y	
Repeated transfer/payment	Y	Y		Y
View Payment History	Y	Y		Y
Add/Delete Payees and Bank Accounts	Y	Y		Y
View bills summary	Y	Y	Y	Y
View electronic version of original bill		Y	Y	Y
Export information into software (e.g., Quicken, Money)		Y		
Write an "E-check" that pays anyone		Y		
Set up pre-authorized payment			Y	

Table 2. Comparison of Features

The table shows that most functions available in commercial applications are also offered by our IA application. The only exceptions are making payment with a credit card, exporting information into a software, writing an electronic check that pays any individual, and setting up pre-authorized payment with the biller. However, these features are not crucial to EBPP [1] and thus their absence does not mean that IA Billing

is inferior to other models. Moreover, the feature to export billing information into financial software can be added easily.

The results of the comparison above eliminated the possibility that IA Billing does not exist because it is functionally inferior to existing models. We further argue that it can work better for the user than other models because of IA's flexibility and "intelligence". Our prototype has been tested under various conditions and the results will be discussed to prove our argument. The goal of presenting these test cases is two-fold:

1. To show how IA Billing handles EBPP operations
2. To demonstrate the decision making ability of the IA

Test Cases

For the following tests, there are five bank accounts A, B, C, D, and E (with the balances shown in the table below) and two billers accounts X and Y. For testing purposes, we have simplified the cases by making two assumptions:

- The balance in a bank account can be zero (although this is not true in reality, we can easily resolve the problem by offsetting the balance by the minimum deposit)
- The user wants to optimize the time, not the cost, needed to complete a payment

The following tables contain information from the knowledge base of our prototype. We are not showing the actual tables (i.e. *BillStatement*, *Biller*, *Bank* etc.) but are instead reorganizing the data into a readable form.

Bank Account	Balance (\$)
A	100
B	200
C	75
D	500
E	700

Table 3. Balances in Bank Accounts

The number of days required for a transaction between two bank accounts and between a bank account and a biller account are listed in the next table. Transfer cannot be made between accounts marked as "N/A". For example, it takes six days to make a transfer from bank account A to biller account Y, and three days from account C to A, but a direct transfer is not available from account C into Y.

Transfer From/To	A	B	C	D	E	X	Y
A	0	2	3	7	2	N/A	6
B	2	0	3	1	2	N/A	4
C	3	3	0	5	2	1	N/A
D	7	1	5	0	5	1	N/A
E	2	2	2	5	0	N/A	N/A

Table 4. Number of Days Required for Transfers

Results of five tests showing different user inputs and circumstances (account balance, amount owed) are shown next.

Test	User's Input	User's Preference	IA's Suggestion	Days Required
1	Pay To: X Pay From: A Amount: \$100	Not specified	Transfer \$100 from A to D and pay \$100 from D to WW	1
2	Pay To: X Pay From: A Amount: \$200	Not specified	Pay \$200 from D to WW	1
3	Pay To: X Pay From: A Amount: \$200	Not to pay with account D	Transfer \$125 from E to C, then pay \$200 from C to WW	3
4	Pay To: Y Pay From: E Amount: \$400	Not specified	Transfer \$400 from E to B, then pay \$400 from B to Y	6
5	Pay To: Y Pay From: C Amount: \$600	If insufficient fund in C, then pay with A	Transfer \$400 from D to B, then pay \$600 from B to Y	5

Table 5. Test Results of Prototype

In test 1, there is sufficient balance in account A to cover the payment, but there is no direct transfer from A to X. Since the user has not set any preference, the IA searches for an account that takes the shortest time to pay X and arrives at C and D. Although it is faster to transfer from A to C, there is not enough balance in C to make the payment to X immediately. Therefore the IA suggests to transfer \$200 from account A to D, and pay biller X from account D. Because there is sufficient balance in account D, these two transactions can take place simultaneously. It will only take one day for biller X to receive the payment, but it takes seven days for account A to transfer the money back into account D¹⁰. Because we programmed our IA to follow the user's original command as close as possible, the IA suggests the extra step of depositing the amount from account A to D. Otherwise, this step is not necessary and a simple transfer from D to X will complete the payment.

In test 2, there is insufficient fund in account A to pay, and the user does not specify any preference. Therefore the IA makes the suggestion to pay with account D instead, which takes one day to complete the payment.

In test 3, there is insufficient fund in account A to pay, and the user specifies in the preference option that no payment should be made from account D. The IA then arrives at the second best solution, which is to transfer the deficit ($\$200 - \$75 = \$125$) from account E to C and then pay biller X from account C. It is necessary to transfer \$125 from E to C because there is not enough balance in account C, and direct payment cannot be made from account E into biller X. It takes three days to complete the payment because it takes two days to transfer from E to C, and one day from C to X.

In test 4, there is sufficient balance in account E to pay, but account E cannot make a direct transfer into biller account Y. It takes the shortest time to pay Y from account B (4 days) but the balance in account B is insufficient, thus a transfer from E into

¹⁰ As we have mentioned repeatedly, this is only one way of designing how the IA process the transfer. This method has the advantage of reducing the time required for the biller to receive the payment, but may introduce inconvenience to the user because it takes time to restore the original balance in account D.

B must be made before a payment from B to Y can be completed. The process takes 6 days to finish because it takes 4 days to transfer \$400 from E to B, and 2 days for B to pay biller Y. This test gives a good example to distinguish the best solution from the optimal solution. With the given conditions, it will take 5 days instead of 6 if the transfer is made as follows: $E \rightarrow D \rightarrow B \rightarrow Y$. There is enough balance in account D to cover the payment, therefore the transfers from D to B (1 day), then B to Y (4 days) can happen at the same time as the transfer from E to D (2 days). Using this solution, the required time for biller Y to receive the payment will be reduced to 5 days in total (1 + 4 days). Although this solution takes a shorter time to complete and is the optimal solution, it violates our requirement that only allows two transfers in a payment and is therefore discarded.

In test 5, there is insufficient fund in account C. The IA then tries account A because it finds the user's preference (from the *PrefersAction* table) to pay from account A if account C does not carry enough balance. However, account A does not have sufficient fund to make the payment either. Therefore IA finds for the shortest possible path, which is to transfer \$400 from D to B, then pay \$600 from B to Y. Two steps are necessary because account D cannot directly pay YY, and account B does not have enough balance to pay Y.

From the test results and comparison with existing applications, it is obvious that IA Billing can adapt to user's preference and benefits and is a better model from the user's perspectives.

Assessing Technical and Economical Feasibility

As stated at the beginning of the chapter, our prototype only includes the part of the application that resides on the user's computer (browser, execution module, and database). In addition to the components in our prototype, a real application would need to communicate with billers and banks to retrieve bill information and send payment requests. The information exchange could occur directly between the user (via the IA)

and the banks/billers, or it could be handled by a central server (provided by the vendor of the application). In the second case, data from banks and billers would be sent to the central server first, which translates the data into a suitable format and sends it to the IA. Likewise, data from the IA would be translated into formats preferred by banks and billers before being sent to these parties. In either case, agreement with banks and billers on the format of data must be made.

The selection on software applications used to implement our prototype was made only for the sake of convenience. Microsoft Access 97 and Internet Explorer 5.0 used in our prototype are common applications available on most PCs, and Microsoft InterDev 6.0 can be replaced easily with other development software to further reduce the cost. Therefore, the cost to develop such an application is not expensive at all.

Besides development cost, the operating cost of the application must also be considered when assessing the economical feasibility. It is the cost to exchange information with banks and billers to perform on-going data exchange actions. Currently, banks use Electronic Data Interchange (EDI) to exchange information among themselves or with other organizations. EDI may be too expensive for our application because it works on Value Added Networks (VAN) [12]. Besides EDI, there are cheaper alternatives that serve the same purpose. Although we have not been able to give an exact evaluation on the technical and monetary costs on data exchange for the IA/Bank and IA/Biller interfaces, we could deduce that it is not difficult to do so technically and economically knowing that there are available products which perform similar functions at relatively low rates.¹¹ For example, EC Exchange by EC Co. is a PC-based software that translates data into EDI format in their central server [16]. It provides an economic way for smaller businesses to send and receive data and payments electronically to banks and corporations that work with EDI format. EC Exchange charges companies at the rate of \$45 USD for 25 transactions [16]. This technique, or similar alternatives, can be applied in the IA Model. Instead of sending and receiving data between businesses and

¹¹ We have not been able to discover the cost of such data interchange operations for the consolidators and how they deal with this cost. This is a limitation of our research that will be addressed in the end.

banks or billers, the interchange could be between the IA (or the central server) and the banks, and between the IA and the billers.

Because the size of data transfer required by individual customers in the IA Model is much smaller, the cost of data exchange should be less than what EC Exchange charges its small business customers. Compared to the charges of other EBPP services (e.g., CheckFree charges \$12.95 USD per month for up to 35 payments [21]), it is reasonable to deduce that the running cost of the IA EBPP application would be appealing to the user.

By implementing a prototype and examining available technologies, we have confirmed in this chapter that it is both technically and economically feasible to build an IA Model.

4.0 Dependency Relations

We have shown in chapter 3 that it is technically and economically feasible to implement an IA Model. In this chapter we will investigate the its strategic feasibility. In order to assess this feasibility, we need to examine the relations among actors in terms of dependency, governance controls, and actions. This purpose can be achieved by using the Dependency Network Diagram (DND) devised by Tillquist, King, and Woo [17, 18]. Theories and background information on the DND will be introduced in 4.1, the method to construct it will be explained in 4.2, and the DNDs of the four models will be presented in 4.3. The objectives of this chapter are:

- To provide information necessary to understand a DND
- To explain the implications of a DND
- To construct DNDs for the four models

4.1 *Theory and Background Information on the Dependency Network Diagram*

The Dependency Network Diagram is a technique that describes the dependency structure among organizations [17]. It gives a standard means to represent inter-organizational relations in a diagrammatic way by showing the dependency, goal, action, and governance control among organizations. In this section we will discuss what a DND can show and the significance of the results.

Each actor has a goal and performs some actions to achieve this goal. To complete these actions, the actor needs resources, where a resource is anything valued by the actor. If the actor has all the resources it needs, it is independent [19]. Otherwise, it looks for them in the environment. When another actor in the environment is able to provide this resource, a dependency arises [19, 20].

Its magnitude, which is called *dependence*, can be defined as the product of the importance of a given resource to the actor and the scarcity of it. When an actor is being depended on by another to achieve its goal, it has influence over it. For example, influence may arise between organizations when one of them is able to supply materials, legislate regulations constraining activities, or provide access to markets [19]. In his article, Thompson noted that “an organization is dependent on some element of its task environment 1) in proportion to the organization’s need for resources of performances which that element can provide, and 2) in inverse proportion to the ability of other elements to provide the same resources or performance. [25]”. Therefore, the amount of influence is also related to the importance and scarcity of the resource that is needed. In other words, if the dependence from A to B is high, B’s influence on A is high and vice versa. Actors can use its influence to obtain favorable conditions such as access to greater markets, lower costs, or extracting higher prices in exchange for their services [17, 18].

When there is an imbalance of exchange between actors, the less dependent actor obtains a net power. This power enables the actor to influence or alter the dependent actors’ behaviors. By withholding, threatening to withhold, or attaching conditions to the continued supply of the needed resource, an actor can compel others to modify their behaviors and goals [19]. Therefore, dependency relations among social actors can shape the structuring of groups, organizations, and markets [17].

There are rules attached to dependency relations, and they are known as *governance controls*. Governance controls regulate the exchange relations between actors, how they should behave, and specify how the resources are possessed, allocated, and used [19]. Governance controls can be formal or informal rules. They can be verbal agreements between friends, or business contracts between organizations. A legal, permanent governance control usually means a more stable supply of resource. For example, if there are only verbal agreements between two parties, either of them can refuse to carry out what it has agreed to do without worrying about being penalized. On

the other hand, if a legal contract is made, the party breaking the contract will face serious consequences.

The activities performed by an actor are determined by the role it holds. By taking up a role, an actor is required to achieve the norms for this role, where norms are the expectations shared by members in the network. This actor must exhibit certain behaviors, perform the necessary actions, and pursue specific goals. Roles are also oriented around goals. In order to achieve their goals, actors are compelled to seek resources [17, 18]. An actor's role changes depending on the situation. For example, a woman can take the role of teacher when she works, and holds the role of mother when she is at home.

The ideas we have discussed in this section are all captured in a DND. The next diagram shows an example DND with three actors with their own roles, actions, and goals. Each arrow indicates a dependency accompanied by its governance control. The diagram shows that both actors 2 and 3 have dependencies on actor 1. The number of dependencies placed on a role, which is equal to the number of arrows pointing to it, is called *centrality* [17]. A high centrality on a role therefore indicates that many other roles depend on it to achieve their goals, and vice versa. Since a role has influence over others who depend on it for resources, a role's relative influence increases when there are more demands placed on it and decreases when it places demands on others [17, 18]. From the DND, these properties can be observed by the number and direction of the dependencies, which are represented by arrows.

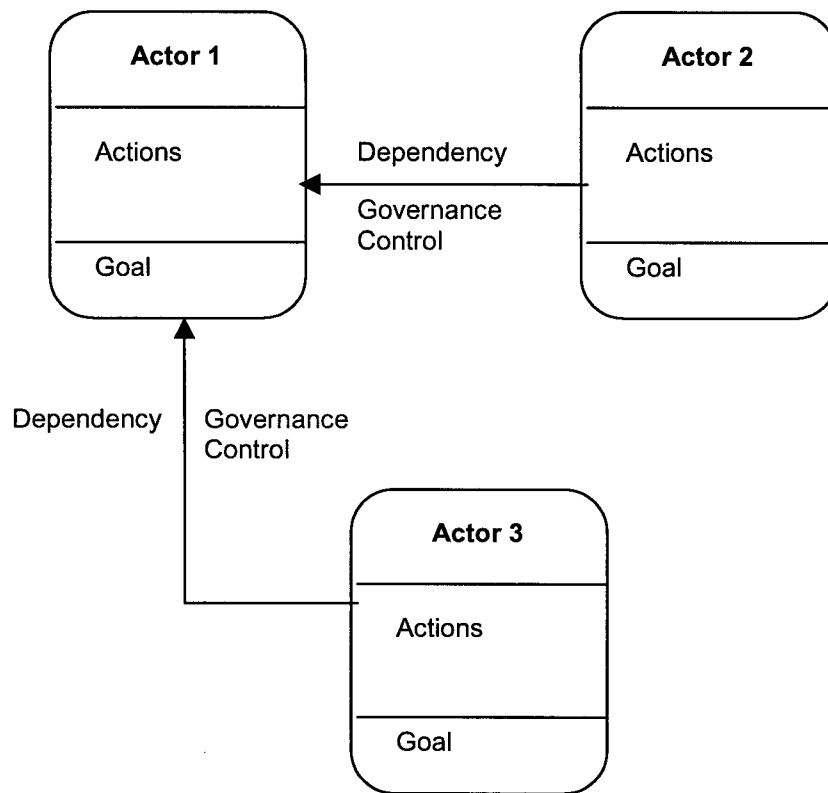


Figure 10. An Example of a DND

The centrality of a role also shows the extent to which the role is able to influence other's activities, dependencies, and goals. This is also called the *sphere of influence*, and is proportional to the centrality of the role [17]. In other words, the role's sphere of influence becomes more extensive when many dependencies from different roles point to it. A bigger sphere of influence gives a role more capability to structure the network. On the example DND, it can be seen that actor 1's sphere of influence includes actors 2 and 3.

By showing the above properties on a diagram, the DND enables one to get a visual picture of the dependency relations among actors. It reveals the structuring of the technological and organizational settings in the network and provides a convenient way to perform analysis such as feasibility study on a network of role-actors.

4.2 Constructing the Dependency Network Diagram – Steps and Rules

After introducing the theories and background of the DND, we will go over the rules and steps to construct it. By following these steps, any individual with the same information about the situation should arrive at the same diagram, which should only contain the crucial information.

The main constructs in a DND are activity, goal, role, dependency, and governance control, and they are defined in [17] as follows:

Activity: a means, procedure, or the provisioning of material or informational requirements necessary to achieve a goal

Goal: desirable or suitable objective

Role: encapsulation of a set of activities and goals

Dependency: need of one role to achieve a goal through the action of another role

Governance control: prescription for acceptable actions within the dependency

A correct DND obeys four rules: the Scope Rule, the Activities Rule, the Goals Rule, and the Dependency Rule. The Scope Rule defines the scope of the model by selecting only the activities, roles, goals, and dependencies that have direct contribution to the goal of the system to be modeled [17, 18]. This rule is useful when roles have goals that are not applicable to the purpose of the model, including these goals will only complicate the analysis. For example, in our EBPP case, a customer may have goals of sustaining a successful marriage or obtaining an academic degree, and these should obviously be excluded for our purpose.

The other three rules specify when a group of activities, goals, and dependencies should be modeled together or separately. According to the Activities Rule, two activities should be combined into one unless they are required by or performed by two separate roles. The Goals Rule states that two goals should be modeled as one unless the activities required and the resources being used or affected by these two roles are

different. Using a similar idea, the Dependency Rule requires that two separate dependencies should involve different activities and affect or use different resources. For example, if there are two dependencies D_1 and D_2 going from Roles R_1 to R_2 , with activities A_{11} from R_1 depending on A_{21} from R_2 through D_1 , and activities A_{12} from R_1 depending on A_{22} from R_2 through D_2 . Then D_1 and D_2 should be modeled separately only if $A_{11} \neq A_{12}$ AND $A_{21} \neq A_{22}$, and the resources used or affected by D_1 and D_2 are different [17, 18].

The above rules form the basis of the construction algorithm of a DND. The algorithm is repeated as follows:

1. *Identify the initial event that triggers something to happen inside the system*
2. *Identify the role r_0 that wanted to accomplish a goal g_0 due to this initial event*
3. *Trace g_0*

Where Trace g_i will do the following

Identify activities a_1, a_2, \dots, a_n needed to accomplish goal g_i using the activities rule

For each a_i ($i = 1, \dots, n$) if there is at least one

If a_i requires another role r_s to accomplish its work then

Construct a dependency from the role containing a_i to role r_s using the dependency rule

Identify the action a_{si} in role r_s that can accomplish the dependent work

Identify the goal g_{sx} in role r_s that is constraining the activity a_{si}

Trace g_{sx}

End if

If a_i triggers another goal and the goal satisfies the goals rule then

Identify the goal, call it g_p

Trace g_p

End if

End for each [17: p. 22]

We will also adapt the pseudo-English syntax used by Tillquist, King, and Woo to represent the dependencies between actors in the network. First, an initial event will start off the analysis, and is represented as follows (the terms in bold are to be replaced with the real terms):

A **problem/opportunity** creates the **resource's** goal to **goal**. The **initiator** must [conditionally] **activity** {, [conditionally] **activity** ...} [to fulfill the **initiator's** goal] to **goal** {, **goal**, ...}. [17: p. 23]

This initial event then triggers other dependencies:

The **dependent** depends upon the **resource** for **dependency** [to fulfill the **dependent's** goal] to **goal**. The **resource** must [conditionally] **activity** {, [conditionally] **activity**, ...} [to fulfill the **resource's** goal] to **goal** {, **goal**, ...}. The applicable criterion for **dependency** is the **governance control**. [17: p. 23]

This cascading dependency in turn invokes other dependencies until no more dependencies can be identified. When this happens, the DND is complete. This algorithm will be used in the next section to build DNDs for the models we have discussed.

4.3 Dependency Network Diagrams of the Models

In this section, we will apply the definitions, rules, and algorithm introduced in 4.2 to construct DNDs for Biller Direct, Bank Forward, Consolidator, and IA models. Analysis on the results will be presented in the next chapter.

4.3.1 Biller Direct

The following is the DND for the Biller Direct Model. The three roles of interest are: Biller, Bank, and Customer.

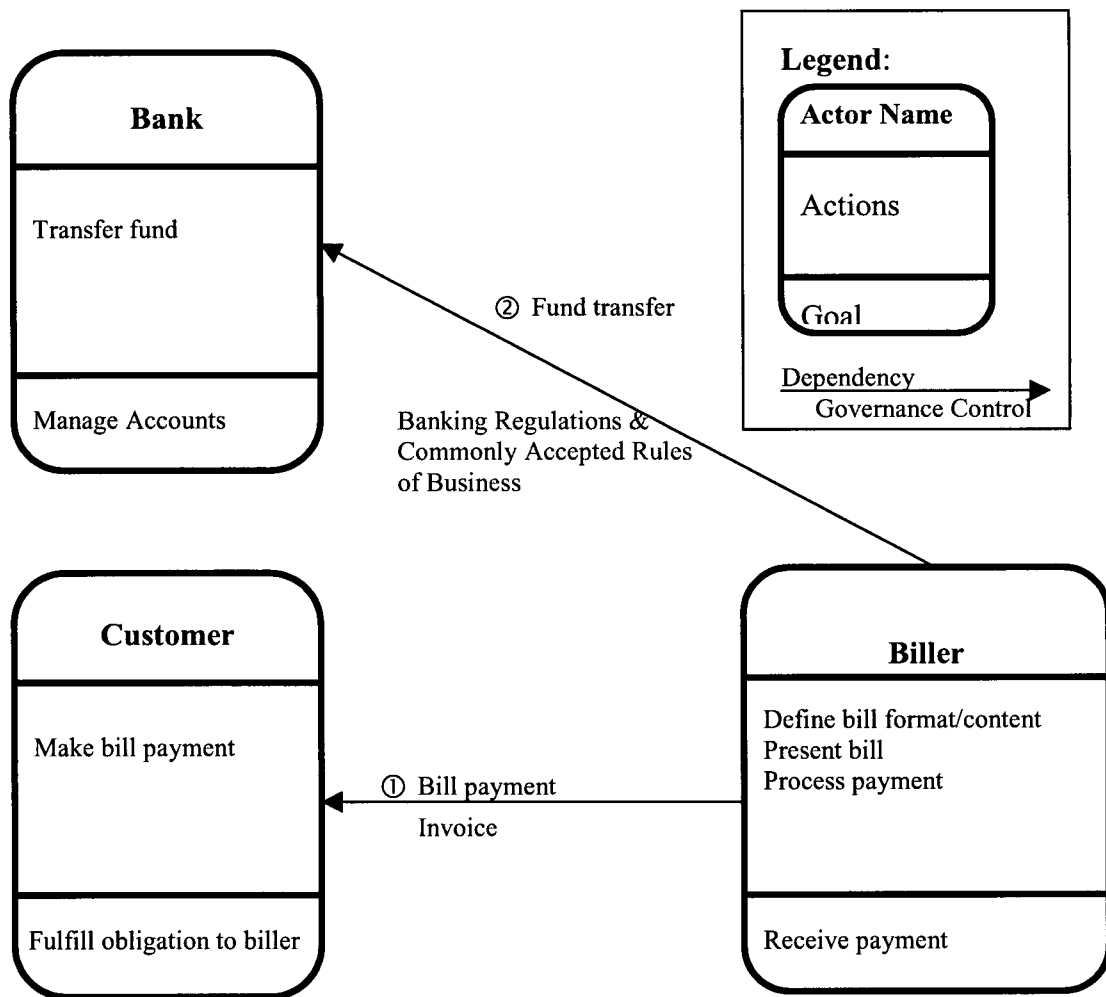


Figure 11. DND for Biller Direct Model

At the end of a billing period, the biller wants payments from its customers and its goal to **receive payment** is invoked.

Initial event:

A **purchase of service/product** by a customer from a biller creates the **biller's goal to receive payment**. The biller must **define bill format & content**, **present bill**, and **process payment authorization to receive payment**.

Before the customer can make a payment, they need to see the bill, and the biller is responsible to present it. By doing so, the biller also defines the format and content of the bill. The action *process payment* refers to the activities performed by the biller on the payment authorization received from the customer, such as forwarding it to the bank, and entering the amount in its database. The biller does not have all the resources to complete this action, one of them is a payment authorization from the customer. This need creates the *bill payment* dependency (dependency ①) on the customer. This dependency is governed by the *invoice*, which is an agreement that contains information such as amount to pay, due date of the payment, and penalty of late payment. The *make bill payment* action of the customer consists of *viewing a bill* and *authorizing payment*. They have been combined into one action according to the activity rule. In order to *make bill payment*, the customer needs to view the bill and authorize a payment at the biller's web site. Payment authorization can be made with credit card or bank account. (As discussed in chapter 2, we will only be interested in paying with bank account for this study.) In this model, both actions are performed at the biller's web site. There is no dependency from the customer on the biller because the biller is providing these resources to the customer in order to obtain the resource it needs: bill payment. It is the biller who needs the payment in the first place. The invoice is a legitimate governance control that must be followed by the parties involved. The resource *bill payment* is very critical to the biller because a business cannot operate without income. It is a resource that can only be provided by the customer in all models.

① Bill Payment (Biller → Customer)

The **biller** depends upon the **customer** for **bill payment** to **receive payment**. The **customer** must **make bill payment** to **fulfill obligations to biller**. The applicable governance criterion for **bill payment** is the **invoice** between the biller and the customer.

After obtaining authorization from the customer, the biller forwards it to the bank (this is part of the *process payment* action). Now the biller depends on the bank to transfer the fund from the customer's to its account and the *fund transfer* dependency (dependency ②) is created. The governance control of this dependency is the *banking regulations & commonly accepted rules of business*.

② Fund Transfer (Biller → Bank)

The **biller** depends upon the **bank** for **fund transfer** to **receive payment**. The **bank** must **transfer fund** to **manage accounts**. The applicable governance criterion for **fund transfer** is the **Banking Regulations and Commonly Accepted Rules of Business**.

4.3.2 Bank Forward

The following is the DND for the Bank Forward Model. Like the Biller Direct Model, the three roles of interest are: Bank, Biller, and Customer.

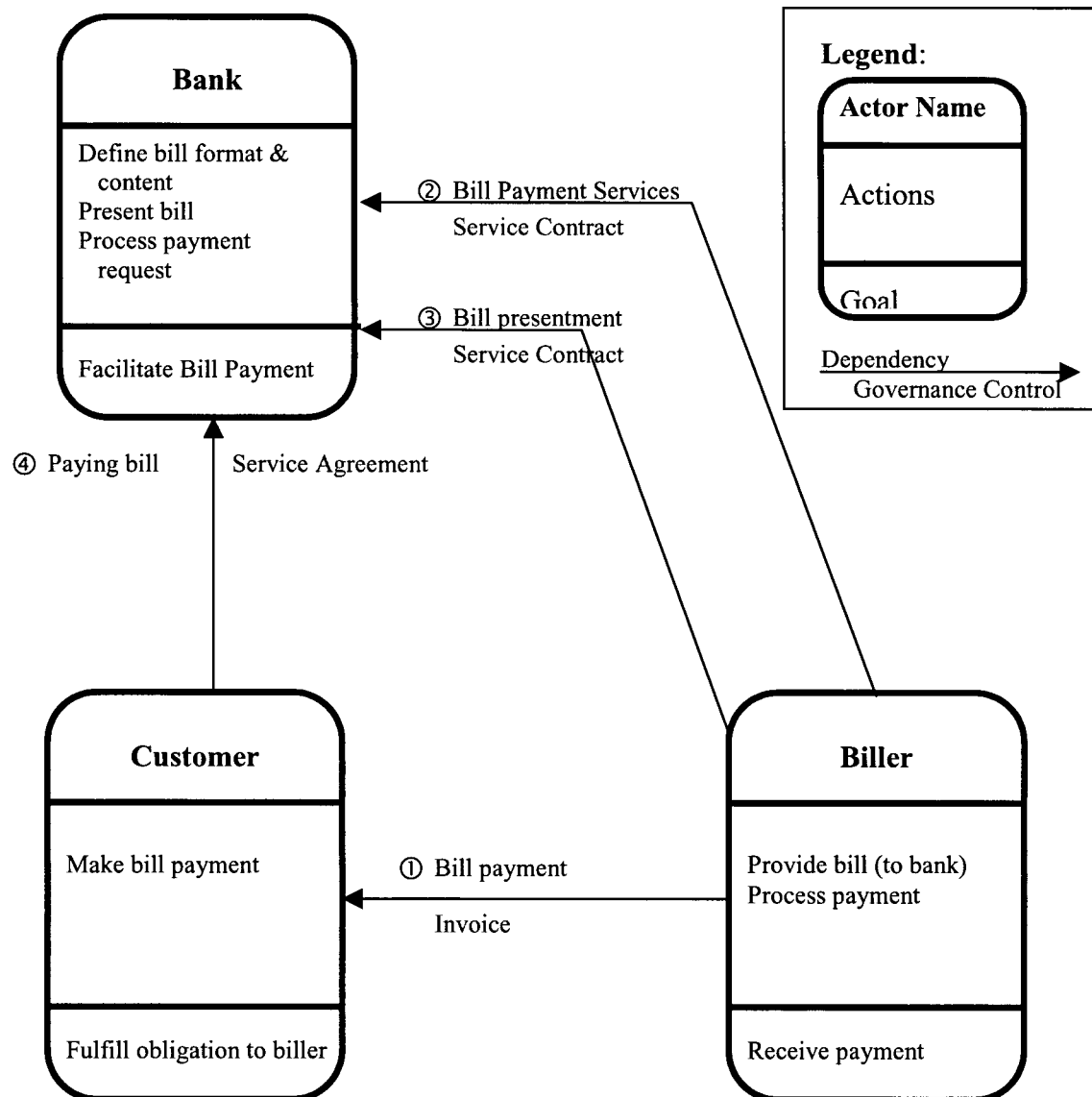


Figure 12. DND of Bank Forward Model

At the end of the billing period, the biller wants payments from its customers and its goal to **receive payment** is invoked.

Initial event:

A **purchase of service/product** by a customer from a biller creates the **biller's** goal to **receive payment**. The **biller** must **provide bill** (to the bank) and **process payment** to **receive payment**.

As in the Biller Direct Model, *bill payment* is a resource needed by the biller from the customer to perform the action *process payment*. This relation is represented by dependency ①.

① Bill Payment (Biller → Customer)

The **biller** depends upon the **customer** for **bill payment** to **receive payment**. The **customer** must **make payment** to **fulfill obligations to biller**. The applicable governance criterion for **bill payment** is the **invoice** between the biller and the customer.

Before *making bill payment*, the customer must receive a bill. Since the biller does not have the resource to present the electronic bill to customer, it relies on the bank for *bill presentment*, which creates dependency ③. In order for the bank to present the bill, the biller must format the bill information and send it to the bank, these two are included in the action *provide bill*. Because the bank defines the format and content of bill presentment and makes fund transfers between accounts, the biller depends on it to supply the bill format and content definition and to transfer payment into its account, which are represented by dependency ② (*bill payment services*). Dependencies ② and ③ are governed by the *service contract* between the bank and the biller. The parties involved in this agreement generally follow the terms, but the agreement can be terminated by either party without serious consequences. The resources *bill presentment* and definition of bill format & content (a component of *bill payment services*) are closely

related. They are represented separately here because we will see later that in certain situations they are split up and performed by different roles.

② Bill Payment Services (Biller → Bank)

The **biller** depends upon the **bank** for **Bill Payment Services** to **receive payment**. The **bank** must **define bill content/format** and **process payment requests** to **facilitate bill payment**. The applicable governance criterion for **Bill Payment Services** is the **service contract** between the biller and the bank.

③ Bill Presentment (Biller → Bank)

The **biller** depends upon the **bank** for **bill presentment** to **receive payment**. The **bank** must **present bill** to **facilitate bill payment**. The applicable governance criterion for **bill presentment** is the **service agreement** between the biller and the bank.

As mentioned before, the customer's action *make bill payment* includes viewing the bill and authorizing payment. The two sub-actions depend on the bank's actions *present bill* and *process payment request*, respectively. Dependency ④ *paying bill* describes this relation. In this model, *process payment request* refers to the bank's action to withdraw fund from the customer's account and deposits it into the biller's. The *service agreement* between the bank and the customer governs this dependency.

④ Paying Bill (Customer → Bank)

The **customer** depends upon the **bank** for **paying bill** to **fulfill obligation to biller**. The **bank** must **present bill** and **process payment request** to **facilitate bill payment**. The applicable governance criterion for **paying bill** is the **service agreement** between customer and the bank.

4.3.3 Consolidator

The following is the DND for the Consolidator Model, the roles of interest are: Consolidator, Bank, Biller, and Customer.

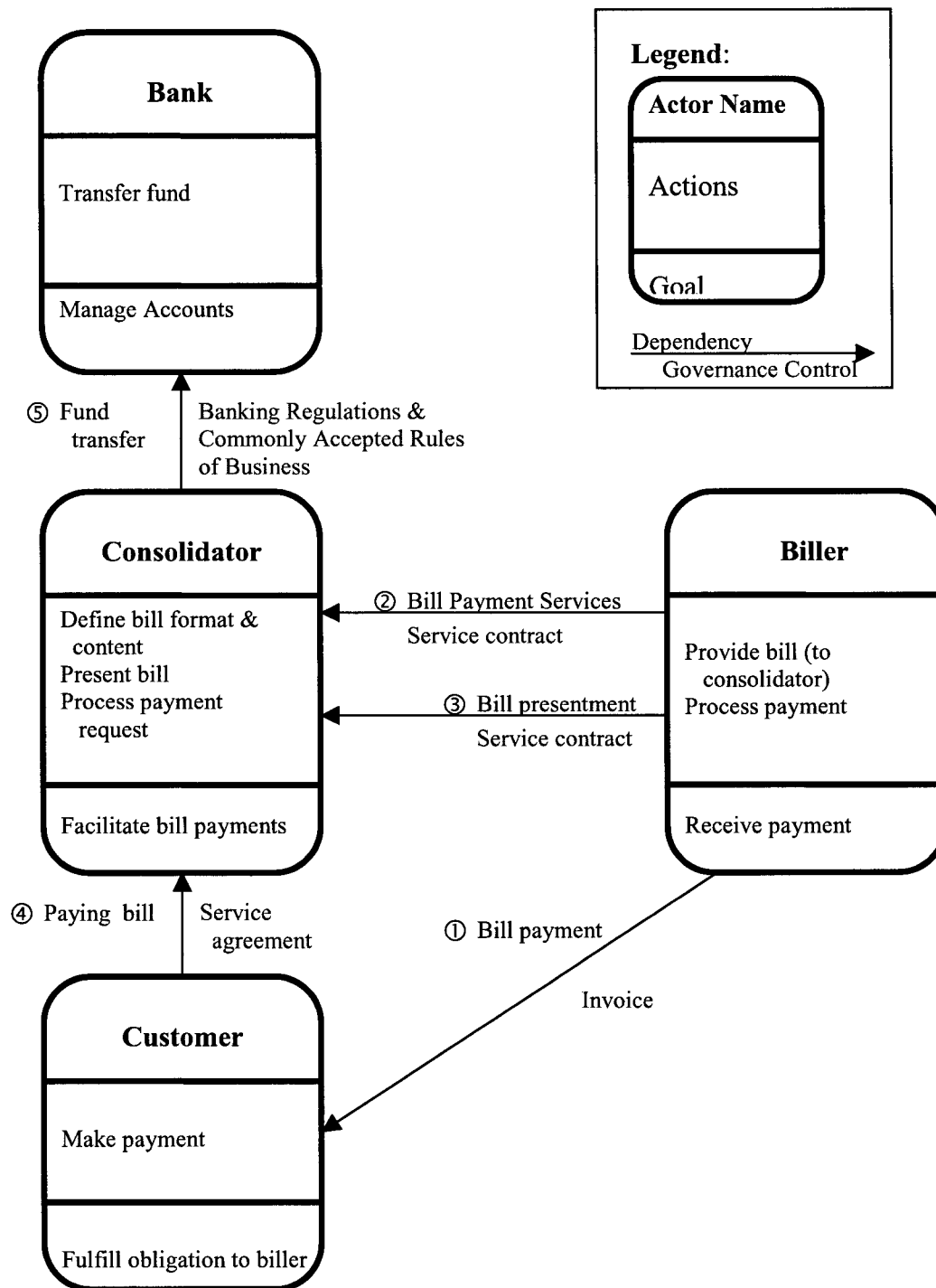


Figure 13. DND for Consolidator Model

At the end of the billing period, the biller wants payments from its customers and its goal to **receive payment** is invoked.

Initial event:

A **purchase of service/product** by a customer from a biller creates the **biller's** goal to **receive payment**. The biller must **provide bill** (to the consolidator) and **process bill** to **receive payment**.

The dependency from the biller on the customer for bill payment (dependency ①) is the same as in models previously discussed.

① Bill Payment (Biller → Customer)

The **biller** depends upon the **customer** for **bill payment to receive payment**. The **customer** must **make payment to fulfill obligations to biller**. The applicable governance criterion for **bill payment** is the **invoice** between the biller and the customer.

The biller lacks the resource to present bill information to customers and relies on the consolidator to provide *bill presentment*, which is represented by dependency ③. Before the consolidator presents the bill, the biller must format the bill information and send it to the consolidator. These two sub-actions are included in the action *provide bill*. The consolidator is the role that defines the format and content of the bill presentment and request the bank to make the transfer into the biller's account, the biller thus depends on it and dependency ② is created.

② Bill Payment Services (Biller → Consolidator)

The **biller** depends upon the **consolidator** for **bill payment services to receive payment**. The **consolidator** must **define bill format/content** and **process payment requests to facilitate bill payment**. The applicable governance criterion for **bill payment services** is the **service contract** between the biller and the consolidator.

③ Bill Presentment (Biller → Consolidator)

The **biller** depends upon the **consolidator** for **bill presentment to receive payment**. The **consolidator** must **present bill to facilitate bill payment**. The applicable governance criterion for **bill presentment** is the **service agreement** between the biller and the customer.

As discussed in the previous models, the customer needs to view the bill and authorize payment to pay the biller. Both activities are included in the *make payment* action. These two activities depend on the consolidator's *present bill* and *process payment request* activities, respectively. Dependency ④ *paying bill* describes this relation. In this model, the consolidator performs *process payment request* by forwarding the payment authorization to the bank [21]. The service agreement between the consolidator and the customer governs this dependency.

④ Paying Bill (Customer → Consolidator)

The **customer** depends upon the **consolidator** for **paying bill to fulfill obligation to biller**. The **consolidator** must **present bill and process payment request to facilitate bill payment**. The applicable governance criterion for **paying bill** is the **service agreement** between the customer and the consolidator.

Contrary to the bank, the consolidator does not have the resource to make the fund transfer from the customer's account into the biller's. It must therefore rely on the bank to transfer fund when it forwards the authorization, as illustrated by dependency ⑤. This is a very critical resource to the consolidator. Without this resource, the consolidator cannot perform the electronic payment component of EBPP and the model cannot function. This dependency is governed by the *banking regulations and commonly accepted rules of business*, which is strictly followed by banks. There are serious consequences for breaking this control.

⑤ Fund Transfer (Consolidator → Bank)

The **consolidator** depends upon the **bank** for **fund transfer** to **facilitate bill payment**. The **bank** must **transfer fund** to **manage accounts**. The applicable governance criterion for **fund transfer** is the **banking regulations & commonly accepted rules of business**.

4.3.4 Intelligent Agent

The following is the DND for the IA Model. The roles of interest are: IA, Bank, Biller, and User (Customer).

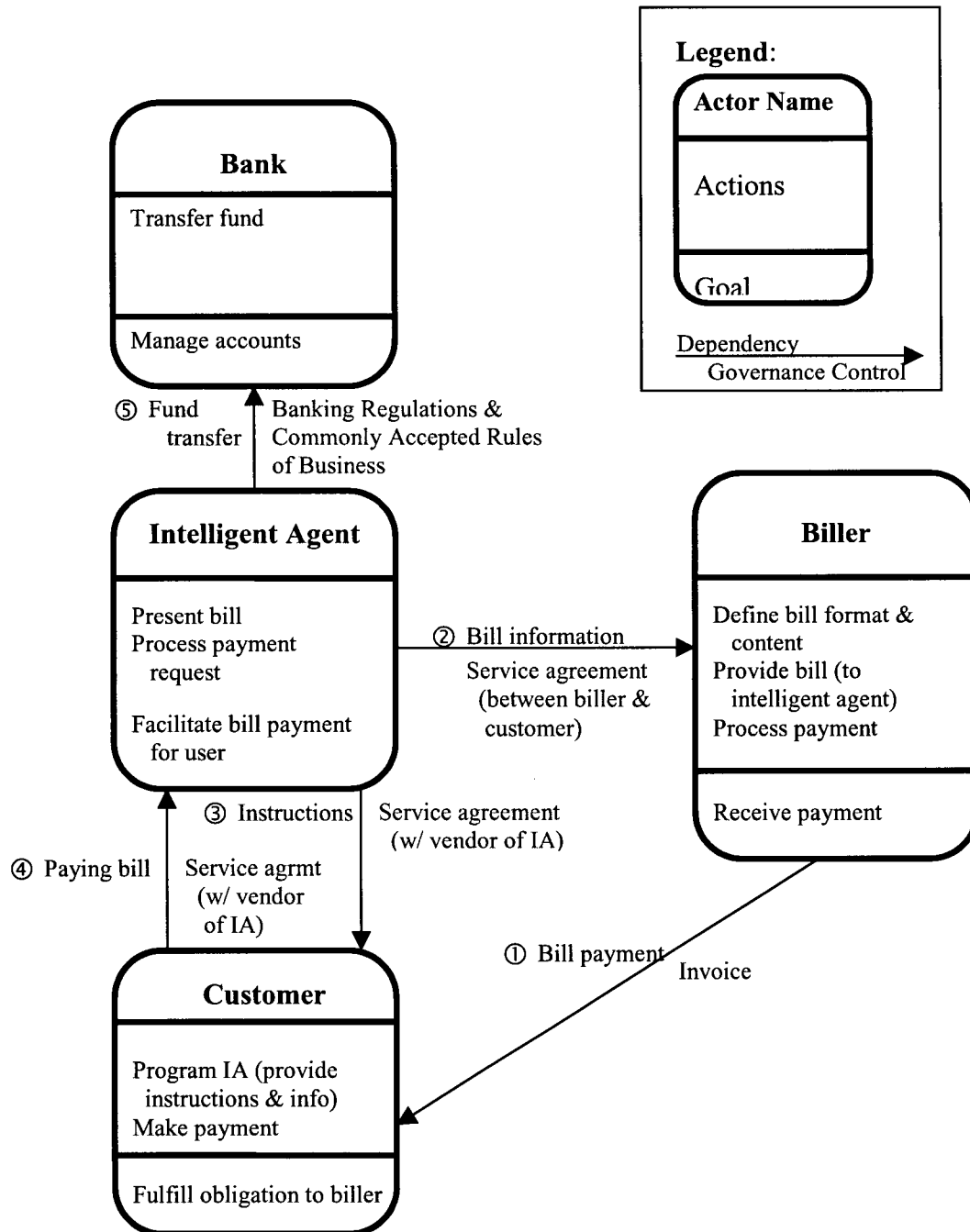


Figure 14. DND for the Intelligent Agent Model

We are using the terms “user” and “customer” interchangeably in this section but they refer to the same entity. The user of IA Billing is the customer to the bank and biller. The vendor of the IA application is not included in our analysis because there are too many possibilities of its identity. It can be a well- established international software company, a telephone company, or an internet service provider. The background of the vendor will significantly affect the outcome of the analysis and thus including it will complicate our study.

At the end of the billing period, the biller wants payments from its customers and its goal to **receive payment** is invoked.

Initial event:

A **purchase of service/product** by a customer from a biller creates the **biller’s goal to receive payment**. The biller must **define bill format & content**, **provide bill** (to intelligent agent), and **process payment to receive payment**.

Again, the dependency from the biller on the customer for *bill payment* is the same as that discussed in other models and will not be repeated here.

① Bill Payment (Biller → Customer)

The **biller** depends upon the **customer** for **bill payment to receive payment**. The **customer** must **make payment to fulfill obligations to biller**. The applicable governance criterion for **bill payment** is the **invoice** between the biller and the customer.

One of the IA’s key jobs is to *present bill*, but it does not have the bill information and relies on the biller to provide it, which creates dependency ②. Because in this model the biller is the role that controls the format and content of its bill information, an internal action *define bill format & content* is invoked before it makes the bill available for the IA. The *service agreement* between the biller and the vendor of the IA application governs this dependency.

② Bill Information (IA → Biller)

The **intelligent agent** depends upon the **biller** for **bill information** to **facilitate bill payment for its user**. The **biller** must **define bill format & content** and **provide bill to receive payment**. The applicable governance criterion for **bill information** is the **service agreement** between the biller and the customer.

Since the IA operates according to the user's specifications and preferences, it depends on the user to perform the action *program IA* and creates dependency ③. The action *program IA* means supplying the IA with instructions (e.g., location and time to fetch the bills) and the user's preferences (e.g., appearance of bill presentment). The IA does not depend on the user when it performs *process payment request* because it is the user who initiates the payment request and by doing so they should have supplied the necessary information. It is critical for the IA to obtain instructions and preference settings from the user because it cannot perform any work otherwise. However, IA is not responsible to do anything not specified by the user, and the user would want to provide this resource so that the application will work fine. The *service agreement* between the vendor of the IA and the user governs this dependency. If the IA does not function the way guaranteed by the vendor, the user can make a complaint and obtain compensation.

③ Instructions (IA → User)

The **intelligent agent** depends upon the **user** for **instructions** to **facilitate bill payment for user**. The **user** must **program the IA** to **fulfill obligation to biller**. The applicable governance criterion for **instructions** is the **service agreement** between the user and the vendor of IA.

The user needs to view the bill and authorize payment to pay the biller, both are included in action *make payment*. These two sub-actions depend on the IA to *present bill* and *process payment request*, respectively. Dependency ④ *paying bill* describes this relation, which is governed by the *service agreement* between the vendor of the IA and the customer. This agreement protects both the vendor and user and guarantees the performance of the IA, but may hold limited legal validity. This resource is not very

critical to the customer because one can always use other methods to pay the biller if the IA Model fails to provide the presentment and payment service. This resource should be as handy to the customer as possible. Otherwise no one will use the model.

④ Paying Bills (User → IA)

The **user** depends upon the **IA** for **paying bills** to **fulfill obligation to biller**. The **IA** must **present bill** and **process payment request** to **facilitate bill payment for user**. The applicable governance criterion for **paying bill** is the **service agreement** between the user and the vendor of the IA.

After the user initiates a payment, the IA will *process payment request* by forwarding the authorization to the bank and depends on the bank to *transfer fund*. Dependency ⑤ describes this relation. The *banking regulations and commonly accepted rules of business* controls this dependency. Although it is a governance control that banks follow rigidly, it may not cover interactions with a non-human agent and therefore banks may refuse to provide this resource to the IA, or impose harsh conditions on the supply.

⑤ Fund Transfer (IA → Bank)

The **IA** depends upon the **bank** for **fund transfer** to **facilitate bill payment for user**. The **bank** must **transfer fund** to **manage accounts**. The applicable criterion for **fund transfer** is the **banking regulations and commonly accepted rules of business**.

5.0 Analysis on Models

In this chapter, we will compare the DNDs of the four models presented in the last chapter and answer our research question.

The objectives of this chapter are:

- Compare the models based on their DNDs
- Identify the weaknesses of the IA Model
- Investigate how these weaknesses can be overcome

Following the procedure we have made, MIS researchers and managers could analyze the workability of their innovations before putting them in the market.

5.1 *Comparison on Models*

Biller Direct

In the Biller Direct Model, the biller is the most important actor. It lies in a central position in terms of information flow because interactions between other actors must go through it. The whole model would fall apart if it were absent because the customers would not be able to view the electronic bills and the banks would not receive the authorization required to make any payment. In this model, the biller has taken up the roles to present bills and originate payment requests, in addition to its traditional role of providing services to its customers. Instead of relying on another party to present its bills and passively waiting to get paid as in other models, it actively presents the bill to the customer and requests payment from the bank.

The extra roles bring the biller many advantages. It controls the billing information, the appearance and content of the presentment, the payment method, and the

delivery time of its bills. The direct contact and close relationship with the customer also gives it an additional means to promote its products and the ability to mine customer information. However, the extra responsibilities also lead to higher expenses. Because of the fierce competitions in the area of EBPP today, a “me-too” service will not be attractive to the public. As a result, the biller needs to invest tremendously on the development and maintenance of the system, in addition to maintaining its old bill processing system. Due to these expensive costs, providing such an EBPP service may only be practicle for large billers.

The bank only plays its traditional role of a payment clearinghouse in this model. When the bank receives a payment instruction and authorization, it verifies it and transfers the money from the customer’s account into the biller’s. To the bank, there is little difference whether it is an electronic bill payment or just a regular payment initiated at a service branch.

Similarly, the customer takes the traditional role of a billee in this model. Although this service is usually free [51], the need to visit multiple sites to retrieve different bills may still be inconvenient to the customer and reduces the attractiveness of this model.

The DND on the Biller Direct Model in figure 11 shows that the biller depends on both the bank and the customer for resources but none of them depends on it in return. Because influence results from the dependence of others on a role’s contributions, activities, and capabilities [19], the biller does not have influence over other actors in the network because it does not own any resource demanded by them. It does not have influence over the bank, and none on the customer after the transaction is completed (the biller has control over certain conditions on the invoice and thus has a degree of influence over the customer *before* the payment). Therefore, it is unlikely that it can obtain favorable terms when it deals with the bank that always has a bigger bargaining power.

Because the biller holds the bill information and defines its format and content, it is able to fulfill one of the two key activities, *present bill*, by itself. However, it still requires the customer to authorize the payment and the bank to make the fund transfer. Both of these are critical resources. As mentioned in the 4.3.1, the supplies of these resources are stable because the governance controls on them (invoice and banking regulations) cannot be easily terminated. For large billers, if they do not give up the bill information to other parties (e.g., bank, consolidator), they always have the ability to keep this service in operation. However, due to the shortcomings of this model, it is expected that the Biller Direct Model will be used less often as other competing models become more prevalent.

Bank Forward

In the Bank Forward Model, the bank also plays the role of a billing agent in addition to its traditional role of a payment clearinghouse. It now sits in a middle position in terms of information flow and manages the transfer of billing information from the biller to the customer as well as payment from the customer to the biller. In addition to settling payments and managing customers' accounts, the bank also has the goal to facilitate bill payments. By acquiring extra roles and responsibilities, the bank has become less replaceable. In the Biller Direct Model, the customer can pay with any bank and the bank cannot actively choose to take part in the process. In this model, however, the bank that provides the EBPP service is the one that customer must pay with.

There are many reasons why banks are in favor of this model. First, it allows them to build up a direct relationship with merchant billers and customers and better understand customers' spending patterns as well as their relationship with other service providers. Second, they receive considerable amount of revenue from billers and some from customers for the services. Third, because money and information travel together, the bank will be positioned to mine consumer data and consequently cross- and up-sell clients to more profitable commercial banking products and services. Fourth, it secures the bank's position and participation in the payment cycle by preventing third parties to

take this opportunity and exclude the bank from this process. Although performing bill presentment and payment electronically requires the bank to give up their old systems and lose revenue, it is an inevitable trend. As EBPP takes hold, the bank will find themselves pushed out of the payments loop, left to handle settlements with low margins and ill-equipped to offer newer and potentially more profitable services. With the advantages they already have, they can earn a larger market share and establish their position in the business. By bringing consumers and billers together at one site, the bank can leverage the trust from clients and act as the intermediary to ensure billers get paid and customers get goods and services. One group attracts another. The more billers partners with the bank, the more customers will use that site. Instead of losing customers, aggressive banks could acquire new ones. Because of the bank's broad influence in the society, they can enforce the acceptance of this model from biller and customers along the "supply push" dimensions [10].

Although the new role as a billing agent brings significant potential benefits, it also introduces extra responsibilities that require the bank to invest tremendously in information systems. Due to the large investment required and the gap between legacy systems and electronic delivery channels, many banks are taking a wait-and-see posture and let others test the profitability of this innovation [10].

In this model, the biller only takes the simple role to issue bills for the services and products it has provided. Its goal is to receive payments. Instead of sending the bill information to the customer, the biller now sends it to the bank, which in turn presents it to the customer. This model is beneficial to the biller because it eliminates the cost of printing and mailing paper bills but does not require an investment in information systems on the biller's side. However, it also causes the biller to lose its direct contact and close relationship with its customers. Large billers often include advertisements when they send bills to customers. Therefore losing this direct contact also means losing a valuable marketing channel which may lead to a reduction in revenue. Because the success of this model (and any other model) would threaten the survival of the Biller Direct Model, resistance from billers who provide EBPP is expected. On the other hand,

small billers who lack the resource to provide EBPP, and do not have much interaction with their customers would probably welcome this model because it gives them an alternative to send bills to their customers.

As in the Biller Direct Model, the customer plays the role of a biller. Although both the Biller Direct and Bank Forward models provide the same functionalities, it is much more convenient for the customer to use the latter because it reduces the number of web sites to visit. However, there is usually a charge for a Bank Forward EBPP service, but a Biller Direct service is always free [36, 37].

The DND of the Bank Forward Model in figure 12 shows that the bank has the resources to complete all its activities. It defines the format and content of the bill presentment for the billers to follow, it is able to present bills in its web site with the information sent by the biller, and it can process customer's payment requests by transferring funds between accounts. It is therefore an independent role. Since both the biller and the customer depend on the bank for resources but the bank does not depend on them, there is an asymmetry in the exchange relation. Net power will accrue to the bank from this asymmetry.

As explained in section 4.3.2, the biller depends on the bank to present bills, define bill format and content, and transfer the payment into its account, and the customer depends on it to pay bills. These dependencies are important to the biller and the customer, and they are governed by the service agreements between these actors and the bank. Although the service agreements are permanent, it is not very difficult to terminate them [24], which implies that the supplies of the resources are not very stable. However, if the bank refuses to present bills for a biller or if the format is not suitable, the biller can always use other means to convey the bill to its customers (e.g., by post, via consolidator) and get comparable results, likewise for the customer. It was discussed in 4.1 that the amount of influence of the resource provider is related to the importance and scarcity of the resource, the bank's influence is thus reduced because the resources needed by its dependents can be obtained elsewhere. It is therefore reasonable to assume that the bank

would make EBPP easily accessible in order to attract more billers and customers. In the future, if a bank manages to gain a large market share, it could provide advantages such as convenience and efficiency that are not available in competitor models. If this happens, the bank's contribution would be less replaceable and its relative influence over billers and customers would intensify. It could use this influence to control the goal and actions of other actors in the network and set conditions favorable to itself.

Consolidator

Unlike the bank and the biller, the consolidator is only present in the Consolidator Model and hence no comparison can be made on its role model-wise. The consolidator works as a billing agent by processing bill information from different billers and presenting the electronic version to the customer. It also works as a payment request originator by sending the payment instruction and authorization from the customer to the bank to request a deposit into the biller's account [1]. There are mainly four benefits that the consolidator enjoys from this model, which are similar to those enjoyed by the bank in the Bank Forward Model: direct relationships with banks, merchant billers and customers, considerable amount of revenue from billers and customers for the EBPP service, ability to mine consumer data and sell them other products and services, and a strong position in the payment cycle which gives it the ability to exclude other parties from participating in the process.

As in the Biller Direct Model, the bank only plays the traditional role of a payment clearinghouse. It does not make the choice to take part in the EBPP process, and there is no difference functionally whether the bill payment is electronic or a traditional one initiated by the customer at the branch.

Similarly, the biller only plays its traditional role. Instead of mailing bills to the customer directly, it sends the billing information to the consolidator in the specified format. It does not actively seek payment as in the Biller Direct Model, but waits passively to get paid. To the biller, there is little difference whether it is the bank or the

consolidator who provides the EBPP service and thus the impacts on the biller would be similar to those in the Bank Forward Model.

As in other models, the customer again takes the role of a Billee. The number of web sites to access bill information and making payments has been further reduced. The customer now has the convenience of having all their bills stored in one location and the choice of paying from accounts at different banks. Moreover, because neither the bank nor the biller has the complete record of the customer's billing and payment information, the customer has a higher degree of privacy. When comparing the costs, however, this model may be less attractive. A Biller Direct service is usually free of charge (e.g., Telus), and banks usually include EBPP with other services in their packages (e.g., Royal Bank), but a consolidator service has a higher charge than the other two.

The DND of the Consolidator Model in figure 13 shows that the consolidator is able to complete most of its activities. It defines the format and content of the bill presentment and it is able to electronically present bill information received from the biller. Both the biller and customer depend on the consolidator for resources and these dependencies are governed by service agreements. A similar pattern of dependency among the bank, biller, and customer in figure 12 can also be observed among the consolidator, biller, and customer in figure 13. The consolidator has a sphere of influence that includes the biller and the customer, and it should have an influence on these two actors similar to that of the bank's. Under the same reasons discussed for the Bank Forward Model, the consolidator would be willing to accommodate needs from billers and customers, especially when it is striving to acquire a large market share.

Contrary to the bank in Bank Forward, the consolidator is not an independent role because it is unable to complete the action of *processing payment request* by itself. To carry out this responsibility, it relies on the bank for *fund transfer*. Although the consolidator can make payments for the customer by other means such as by check or by credit card, it would be inconvenient for both the customer and the biller to do so, and would considerably reduce the attractiveness of the service. Therefore, the *fund transfer*

dependency is very important to the operation of the Consolidator Model. This dependency is governed by the banking regulations and commonly accepted rules of business, which is strictly followed by banks, and cannot be changed or terminated easily. Although the governance control guarantees a stable supply of resource, the consolidator is still highly dependent on the bank due to the scarcity and importance of *fund transfer*. This dependence gives the bank influence over the consolidator. As a result, the consolidator has less freedom when making decisions, setting goals, and performing actions compared to the bank in the Bank Forward Model.

It was mentioned in section 2.1 that some consolidators provides accounts to customers in which they can store money and pay bills from. By doing so, the consolidator has taken up the role of the bank. In this case, the bank will not be included in the DND and the dependency from the consolidator to the bank for fund transfer will be eliminated. The consolidator has become an independent role and has clearly increased its relative influence on other members in the network. It is obvious that the bank dislikes this method of EBPP because it excludes the bank's participation in the process. However, because this kind of Consolidator Model is still not as prevalent as the one to pay with bank accounts, customers may find it inconvenient to use, and thus no further investigation will be made.

Intelligent Agent

As in the Biller Direct and Consolidator models, the bank only plays its traditional role of a payment clearinghouse in this model. When it receives a payment instruction and authorization from the IA, it verifies the authorization and follows the instruction to make the transfer.

Similarly, the biller plays its traditional role to provide a service and issue a bill for the service. Instead of mailing the bill to the customer directly, it sends the billing information to the IA upon request. Unlike in other models, the biller is not responsible

for sending the bill information until it is asked by the IA¹². This model reduces the biller's workload and saves it the cost to invest in EBPP system or send paper bills by mail, but takes away its direct contact with the customer.

The user's role is again a billee. Similar to the Consolidator Model, bills from various billers are accessible to the user at one location, and paying from accounts in different banks is allowed. However, because of IA's unique abilities and its goal to work primarily for its user, the user can enjoy many benefits that are not available in other models, as we have mentioned before:

- The application can be customized to suit each user's particular needs.
- The user can avoid being marketed on products and seeing unwanted advertisements.
- The IA can make suggestions to the user.

On the other hand, users of the IA application may be liable to more responsibilities. In other models, the service providers will cover the customer's loss if the customer is not at fault (e.g., late payment due to late transfer from the bank) [21, 24]. In IA Billing, since most of the operations are performed by IA under the user's preferences, the chance of making a mistake on the user's side is higher. It would be more difficult for the user to prove that they are not at fault. As a result, there may be psychological barriers to consumers to use this application. Finally, economic cost is another minor drawback because IA Billing would be more expensive to use than Biller Direct or Bank Forward, but it should not cost more than a Consolidator application.

Figure 14 shows that IA does not have the resources to complete any of its activities in DND context. It depends on the biller and user to present a bill, and on the bank to process a payment request. Because IA does not own resources needed by other roles, it does not have influence on them. Instead, it is under the influence of the bank and biller who own the resources it needs. Furthermore, since it is not difficult for banks

¹² We have made this assumption because the increase of flexibility in the IA Model and its active role would logically increase the customer's responsibilities.

and billers to terminate the governance controls on these dependencies, a stable supply of the resources is not likely. Compared to the EBPP providers in other models, IA's dependence on other roles is higher. There are many projects and technologies in MIS that are similar to IA Billing, which are technically feasible to implement but still result in impossible deployment. If the researcher or manager examined the dependency relations among the parties involved, they could detect the weaknesses at an early stage and modify their designs accordingly. In the following section, the DND of the IA Model will be examined in further details.

The reader should be careful not to confuse the terms "independent" and "autonomous". In this thesis, we have used "autonomous" to refer to IA's ability to complete jobs delegated to it without human supervision. The term "independent", however, has been used to describe a role that has the resources it needs to complete all its activities. Therefore, a dependency is created when IA relies on another role to complete an activity that involves interaction with an external role. Let us use IA's dependency on the bank for fund transfer as an example. Although IA is autonomous and can devise a solution to make a payment and send the payment authorization to the appropriate banks (internal tasks), it cannot transfer the fund between accounts to make a payment (external task) and must rely on the bank to do so.

5.2 *Shortcomings of the IA Model*

After constructing DNDs for the models in chapter 4 and comparing the results in the previous section, we will discuss the weaknesses of the IA Model in the following six areas and investigate if any of them can be solved:

- Change in roles
- Distribution of resources
- Differences in goals
- Change in governance control
- Redistribution of activities
- Change in pattern of dependencies (which leads to influence)

Change in Roles

An actor's role defines its behaviors, actions, and goals. In the traditional way as well as in the Biller Direct Model, the customer receives their paper or electronic bills from the biller, who handles and processes the bill information by itself. In the Bank Forward, Consolidator, and IA models, the bills are presented to the customer by the bank, the consolidator, and the IA, respectively. By performing actions that used to be completely carried out by the biller, pursuing the goal to facilitate bill payment, and behaving as bill presenters, these actors have taken up the part of the biller's role as bill dispatcher.

Similarly, customers deal with the bank directly to access their bank account information or initiate a transfer in the traditional way and in the Bank Forward Model. The bank is the actor that presents account information to the customer and handles the transfer requests. These actions are shared by the biller, the consolidator, and the IA in the corresponding models. By performing these actions, exhibiting behaviors of a bank, and pursuing the goal to facilitate bill payment, they have taken up part of the bank's role.

Changes in roles for the biller and the bank lead to changes in patterns of dependencies (will be discussed shortly) and will decrease the relative influence of the biller and bank over others and reduce their scope of influence. It is therefore reasonable that they do not welcome these changes. Since IA represents and works solely for its user (customer), and customers generally have little bargaining power compared to large companies [23], the bargaining power of the IA is also weak.

In order for IA Billing to operate, it is necessary for IA to act like a bank and biller in certain ways to process bill information and payments. As a consequence, a change in roles in the network is inevitable and this weakness cannot be amended.

Distribution of resources

In this thesis, we define the key resources in EBPP to be bill information, bill payment, and fund transfer because they are the essential elements in the process. They belong to the biller, the customer, and the bank. Bill payment and fund transfer are always provided directly by the customer and the bank. Bill information, however, can be viewed as “transferable”. Even though bill information is owned by the biller, it is not sent by the biller directly to the customer except in the Biller Direct Model. Instead, the biller supplies it to the role that presents the bills (bank or consolidator), which in turn provides this resource to the customer.

Although the consolidator does not own any key resources like the bank and biller, it plays a significant part in the Consolidator Model by providing the EBPP service. The Consolidator Model exists because the consolidator makes a contribution that is valued by other key actors. As discussed in section 2.1, a consolidator usually has a long history in financial management and can provide state of the art technologies on EBPP that may be too costly to develop for banks and billers (especially small ones). With the Bank Forward Model favoring the bank, Biller Direct Model favoring the biller, the Consolidator Model takes a relatively neutral position by balancing the benefits among all parties.

Although neither the IA nor the consolidator owns any key resource, IA does not have a strong bargaining power and does not provide significant benefits to the other parties as the consolidator does. When a role depends on another for resources but does not contribute anything in return, it is unlikely that the resources can be obtained. Because bill information and bill payment are always owned by billers and banks, the IA always needs to obtain these resources from them and there is no way to change the distribution.

Differences in goals

From the DNDs of the models, it can be observed that both the biller and customer have consistent goals. The biller's goal is to "receive payment", and the customer's is to "fulfill obligations to the biller" (i.e. pay all the bills to avoid penalty). The bank, however, changes its goal from "managing accounts" to "facilitating bill payments" in the Bank Forward Model when it takes up the extra responsibilities to present bill and process bill payment request. As an EBPP service provider, the bank's goal coincides with that of the consolidator.

It is safe to assume that the biller and the bank also have the goal to maximize their own benefits in the Biller Direct and Bank Forward models, respectively. Although the consolidator also has the same goal, it evens out the benefits among the bank, biller, and customer. The IA, on the other hand, has the goal of facilitating bill payment for its user and work solely for their needs. It does not aim to work for other actors in the network like the bank and consolidator. Because the IA does not consider the benefits of the bank nor the biller, it would make decisions that are beneficial to its user that are not advantageous to other parties. Examples include IA's suggestions to the user to pay with other bank accounts that cost less, or delay a payment until the due date. It is therefore logical that banks and billers would not favor such a model.

In order for IA Billing to remain as a customer-oriented model, its goal to represent its user and prioritize their needs should not be altered. If this goal were

changed to please the bank and biller so that they do not reject IA Billing, the model would not be customer-oriented anymore. Therefore this obstacle cannot be eliminated.

Redistribution of activities

Except in the IA Model, the roles that present bills also define the content and format of the bill presentation (biller in Biller Direct, bank in Bank Forward, and consolidator). Although neither the bank nor the consolidator owns the bill information in the Bank Forward and the Consolidator models, they can still define the format and content of the bill presentment. In the IA Model, however, the IA presents bills but does not have the same ability. The biller is the actor who chooses the format and content of the information to send to the IA. In other words, there is a *redistribution of activities*: the activities *present bill* and *define content and format of bill presentment* that are performed by the same role and treated as one activity in other models are now split up and carried out by two different roles. Because the IA's goal (to facilitate bill payment for user) always requires the activity *present bill*, which in turn requires a definition of bill content & format, a dependency on the biller (dependency #2) cannot be avoided.

As the number of billers increases, standardization of bill content and format would become more crucial to the operation of IA Billing. If each biller sends the bill information with its own choice of format and content, the information received would be in numerous formats having various contents. As a result, the job to present bills would be very difficult for the IA because it will need mechanisms customized to read information from each biller, which will be both costly and inefficient. Furthermore, if the billers have the freedom to choose the formats and contents of the bill information, they may also change these definitions periodically. As a result, the bill presentment activity will be complicated even further because the IA will be required to keep track of all the changes and differences of the information it receives.

Change in pattern of dependencies

Pattern of dependencies refers to the direction of the dependencies in the network. A change in the pattern of dependencies can increase an actor's centrality or distance it relative to the setting and lead to changes in the actor's relative influence and autonomy, in the network [17].

As discussed in the last section, the one-way dependencies from the customer and biller to the bank in the Bank Forward Model reveal that the bank has control over the access to the required resources. This control gives the bank influence over its dependents and enables it to change their behaviors and goals. The same kind of influence is also enjoyed by the consolidator in the Consolidator Model although the consolidator has one dependency on the bank and is not completely independent. On the contrary, such pattern of dependencies is absent in the Biller Direct and IA models and therefore neither the biller nor the IA has influence over other roles in the presentment and payment process. Because of the lack of influence of the biller and the weaknesses of the Biller Direct Model discussed earlier, this model may eventually be phased out when other models become more popular. The reasons that this model exists in the market have been discussed and will not be repeated here.

Like the biller in the Biller Direct Model, the IA depends on other roles for resources but does not provide any resource needed by them (except the user) in return. Because these resources are crucial to the operation of the IA and they can only be obtained from the bank and biller, the IA's dependence on these two actors is high [19]. Despite the similarities, the IA is different from the biller in two aspects. First, IA's bargaining power is weak compared to other members in the network, but large billers can obtain strong bargaining power through other channels. Second, unlike the biller, IA does not have the resources to complete any of its actions alone. As a result, IA is a more dependent role and it would be very difficult for it to request resources. This pattern of dependency will always exist in the IA Model and we will discuss the reasons shortly.

Change in Governance Control

In each of the three existing models, the role that performs the EBPP service also designs the service agreement that governs the dependencies on the supply of bill information and definition of bill format and content. There may be situations in the Bank Forward and Consolidator models where influential actors (large billers and banks) could make some negotiations, but in most cases, the provider of the EBPP service lays out the terms on the agreement or contract and will make the terms and conditions favorable to itself.

In the IA Model, however, the service agreements would be between the bank and the customer, and between the biller and the customer, and they would be entirely laid out by the bank and biller. Because this model does not give any immediate advantage to the bank or biller¹³, it is unlikely that they would devise conditions that are favorable to the customer. Under the same reason, such agreements would not be permanent and could be terminated easily by the bank and biller.

Due to the absence of permanent governance controls, the supply of critical resources (bill information and fund transfer) to the IA is uncertain. Organizational vulnerability derives from the possibility of a critical resource becoming unavailable [19]. When applied to the IA Model, it is apparent that the model is vulnerable because the possibility of failing to obtain critical resources is relatively high. The vulnerability of IA Billing would reduce its attractiveness to potential partners and users.

The first two weaknesses discussed in this section (change in roles, distribution of resources) are not unique to IA Billing, but also appear in other models. The other weaknesses (differences in goals, redistribution of activities, change in pattern of dependencies, and change in governance controls) only appear in the IA Model, and are more significant barriers. Because redistribution of activities and change in pattern of

¹³ There may be advantages to them such as reduction on postage and paper costs, but they would be insignificant compared to the drawbacks.

dependencies are results of the weak governance controls, it is apparent that governance control plays a very vital part in the survival of a billing model. Two permanent governance controls (similar to those in the Consolidator Model) are needed to improve the situation:

- Governance control to ensure that the bank will always complete transfers when requested by the IA, and at a reasonable cost.
- Governance control on the supply of bill information to ensure the supply and standardization of bill information from the biller. Whether the biller should be responsible to notify the IA or its user when new billing information comes up is an arguable topic because IA is autonomous enough to retrieve bills at the appropriate time, and it may be unfair for the biller to have to take up this responsibility. If this responsibility is given to the biller, the biller will depend on the IA to transfer the bills to the customer and dependency #2 will be reversed.

The first governance control can secure the fund transfer dependency (dependency #5 in figure 14) in the DND, and the second one eliminates the redistribution of activities and protects the supply of bill information (dependency #2). Since IA would no longer depend on the biller for definition of bill format and content under these governance controls, dependency #2 would be modified. The resultant DND would still remain the same, but the critical resources would be permanent, IA's dependence on others would decrease, and the model would be less vulnerable.

Unlike the Consolidator Model, IA Billing can never be as advantageous to the bank and biller because of IA's goal. Therefore, there must be an actor supporting the IA Model (i.e., vendor/service provider of the model) who can compel the bank and biller to co-operate so that the governance controls specified above can be achieved. This actor should be able to collect tremendous relative influence on the bank and biller through dependency relations in other processes because these two parties would have considerable resistance to IA Billing.

It is common knowledge that banks(especially large ones) has been a very influential group in the society. Although not as influential as the banks, large billers also gain significant relative influence on members in the society through their businesses. To find an actor that can compel these two parties to compromise on a model they do not like would be extremely difficult. Furthermore, because the number of billers is large and they do not group together and follow a set of common code of conduct like banks do (banking regulations and common rules of business), it would be very tedious for the vendor of IA Billing to convince these billers to accept the model. The Bank Forward and Consolidator Models do not face this problem because the billers who enroll have an interest to use the service.

Considering the above restrictions, a state government may be the only suitable candidate to advocate IA Billing because it is able to pass laws to provide the necessary conditions for the model to exist, and its goal include working for its citizens. However, it would be very unusual for a government to intervene in EBPP because it is not a vital service to the general public and there is currently no monopoly in the business. Even if the government decided to promote IA Billing, the model would only work in that country. Getting other countries to follow would be almost impossible.

After reviewing the weaknesses of the IA Model, we have gained a better understanding on the barriers to introduce it to the market. Despite the advantages offered by this model to the user (as discussed in 3.1), we have demonstrated in this section that the obstacles preventing this model to work are too difficult to overcome. From this chapter, we have shown the importance to assess the weaknesses of a new technology by examining the dependency relations involved. Regardless of how many virtues it has, if the technology exhibits similar dependency relations as the IA Model, it has a high chance of failure.

6.0 Conclusion

Although many models of Electronic Bill Presentment and Payment (EBPP) exist in the market, none of them is oriented around the customer's needs. After observing the wide applications of Intelligent Agent (IA) in other areas of e-commerce, we recognized a suitability to use this technology to implement a customer-oriented model of EBPP. To investigate the reasons of why this model has not come into existence became the goal of this thesis.

We hypothesized in the beginning of our research that it would be both technically and economically feasible to implement an EBPP model operated by IA, and the answer to our research question should lie in the strategic and organizational feasibilities of launching such a model into the market.

The first step to prove our hypothesis was to build a prototype of the model. By building the prototype and analyzing the technical and economical issues involved, we showed that:

- The application could be built with common software packages and basic computer science knowledge.
- The application could provide all the functions available in existing applications.
- The performance of the application is satisfactory on a general PC.
- The implementation cost of the application is very low, and the running cost is presumably comparable to that of a Consolidator application.

The above results proved that it is both technically and economically feasible to implement an IA Model.

To determine the strategic feasibility of the model, we first represented the IA Model along with the three most prevalent models (Biller Direct, Bank Forward, and

Consolidator) into Dependency Network Diagrams (DND). From the DNDs we could observe the dependency relations among actors in each model. By comparing and analyzing the DNDs and applying theories on organizational topics obtained from literature research to the results, we arrived at the answer of our thesis question. The answer was divided into six parts: change in roles, distribution of resources, differences in goals, redistribution of activities, change in pattern of dependencies, and change in governance control.

As a customer-oriented model, IA Billing does not balance the interests of actors in the network. It favors the interests of its user (the customer) and induces resistance from other actors. Furthermore, due to the distinct pattern of dependencies in the IA Model and redistribution of activities, IA is highly dependent on the bank and the biller and does not have any influence on them. These barriers could be resolved if there were permanent governance controls on critical resources that give favorable conditions to the IA Model. However, this can only be achieved if the vendor of the model has significant influence on banks and billers so that it can compel them to co-operate and accept the required governance controls. We deduced that a state government may be the only possible solution, but it would be very unusual and unlikely that a government would intervene in this kind of business. Therefore, the IA Model has not existed and probably never will.

DND has been used as the main tool to study the dependency relations in EBPP models. We chose to use this tool because it is designed to capture and depict dependency relations among role-actors in a network. It provides a standard and systematic means to break down inter-organizational relations by identifying the dependencies among the members and enables us to analyze these interdependencies. DND is particularly useful for our study because it does a good job in analyzing and predicting the impact of implementing new IT into a process and suits our purpose to analyze the strategic feasibility of billing models.

However, there are two drawbacks of the DND. It represents all dependencies in the same fashion and does not distinguish them according to their properties such as their importance to the dependents and scarcity of the resources. For example, in the Bank Forward Model, the biller's dependency on the customer (bill payment) is more crucial to the biller than the customer's dependency on the bank (paying bill) because the second dependency can be removed if the customer pays the bill by other means whereas the first one must be supplied by the customer, but they are depicted in the same manner in the diagram. Furthermore, the DND does not show the number of members represented by one "actor" (e.g., "Bank" in the Bank Forward Model can represent only one bank or many banks). Because the concentration of a dependency is related to the number of providers of that resource, it cannot be observed from the diagram. As a result, an actor that many dependencies point to may look very influential on the DND even though its influence is diluted due to the large number of members it represents.

Contributions

There are two important contributions of this research. First, most existing literatures on the topic discuss issues such as the impacts and challenges EBPP brings to the society, and comparisons are usually made between service providers. Our research, however, brings a new insight into EBPP by studying the dependency relations between actors in a model, and using the results to compare models.

Secondly, the answer to our research question is not only limited to EBPP, but can also be applied in other settings and processes, and will be especially useful for researchers and managers in the MIS domain when they develop technologies that rely on other parties for resources like the IA does. In a given process, roles can exert influence over other roles based on their relative dependencies. Influential roles can shape the activities, roles, goals, and norms of the process by compelling others to modify their behaviors. Because studying the dependency relations among roles in a business process leads to an understanding on aspects of influence and how the process is defined, MIS researchers and managers should pay attention to dependency relations when designing a

new technology. They can follow our method and use the DND to analyze their technologies and make modifications so that they are appropriate for the situation.

Limitations and Future Research

There are only a few limitations of this study that need to be addressed. First, if the IA Model were widely used, there would be a huge number of agents dialoguing over the Internet. However, no real experiment has been conducted in which such a large amount of agents work together [5] and there may be impacts that cannot be foreseen until such an experiment is carried out.

Secondly, our prototype does not include the IA/Bank and IA/Biller interfaces. We have accessed the workability of these interfaces from similar examples in the market and literature research. Therefore difficulties that we are not aware of may be present. Finally, because most factors considered in our research were gathered from literature review, there is a possibility that some real-life factors were missed.

The setting of this search is the North American market. Because banking systems and business standards vary in other countries, the results of this research may change if it is conducted in other parts of the world. Therefore, further research on this topic to study the effects of geographic settings on the feasibility of a billing model would be of great interest to the audience. Another possible study on EBPP is to determine which of the existing models in the market will prevail in the long run. Similar approach used in this thesis may be applied to solve these questions.

Bibliography:

1. Business Practices Task Force of NACHA's Council of Electronic Billing and Payment, "An Overview of Electronic Bill Presentment and Payment Operating Models," April 9, 1999.
2. Higgins, Kelly Jackson (1996) "Your Agent is calling," *Communications Week*, v622 (May 8), p. 45.
3. Maes, Pattie (1995) "Intelligent Software," *Scientific American*, v 273, n 3 (September), pp. 84-86.
4. Maes, Pattie et al. (1999) "Agents that Buy and Sell," *Communications of the ACM*, v42, n3 (March), pp. 81 – 91.
5. Nwana, Hyacinth S. et al.(1998) "Agent Mediated Electronic Commerce: Issues, Challenges, and some Viewpoints," *Autonomous Agents*, pp. 189 – 196.
6. Stoddard, Brooke C. (1997) "The Check's On Line," *Electric Perspectives*, Nov/Dec, pp. 59 – 64.
7. Wand, Yair "The Feasibility Study," *Lecture Notes*, 2000.
8. Sabherwal, R., and King, W.R. (1991) "Towards a Theory of Strategic Use of Information Resources," *Information and Management*, v20, n3, pp.191-212.
9. Wooldridge, M. and Jennings R. (1995) "Intelligent Agents: Theory and Practice", *The Knowledge Engineering Review*, v10 n2, pp.115-152.
10. Shaffer, Richard A. (1995) "Who Wins Home Banking?" *Forbes*, v156, n4, p. 163.
11. Boy, G. A. and Mathe N. (1993) "Operator Assistant Systems: An Experimental Approach Using a Telerobotics Application," *Knowledge Acquisition as Modeling* (K. M. Ford and J. M. Bradshaw eds.), pp. 271-286, Wiley: Reading, New York.
12. Du Rea, Mary, Pemberton, J. Michael. (1994) "Electronic Mail and Electronic Data Interchange: Challenges to Record Management," *Records Management Quarterly*, v28 ,n4, pp. 3-11.
13. "EC Exchange 1.1," *InfoWorld*, v20, n14, 1998, pp. 77-80.
14. Zerega, Blaise. (1998) "Solution Reduces Inventory and Administration Costs," *InfoWorld*, v20, n18, p. 116.
15. Dunn, Julie, et al. (1998) "Make the Connection," *InfoWorld*, v20, n14, pp. 74-77.

16. King, Julia.(2000) "Free Software Brings Small Suppliers Online," *Computer World*, v34, n6, p.2.
17. Tillquist, John, King, John L., Woo, Carson (2001) "A Representational Scheme For Analyzing IT And Inter-organizational Dependency," submitted for journal publication.
18. Tillquist, John, King, John L., Woo, Carson (2001) "Modeling IT and Organizational Coordination: The Dependency Network Diagram," submitted for journal publication.
19. Pfeffer, J., and G. Salancik, *The External Control of Organizations: A Resource Dependence Perspective*, Harper and Row: New York, 1978.
20. Emerson, R. M. (1962) "Power-dependence relations," *American Sociological Review*, v27, pp. 31-41.
21. Description on Checkfree's Electronic Bill Presentment & Payment Services, 1999, <http://www.checkfree.com>.
22. Papazoglou, Mike P. (2001) "Agent-Oriented Technology in Support of E-Business," *Communications of the ACM*, v44, n4 (April), pp. 71-77.
23. Stahl, Dale O. (1996) "Oligopolistic pricing with heterogeneous consumer search," *International Journal of Industrial Organization*, v14, n2, pp. 243-268.
24. "Online Access Agreement for Wells Fargo Online and Wells Fargo Business Online Services," 2001, <https://banking.wellsfargo.com/common/html/wibdisc.html> .
25. Thompson, J.D. *Organizations in Action*, McGraw-Hill: New York, 1967.
26. Afuah, Allan N., Bahram, Nik (1995) "The Hybercube of Innovation," *Research Policy*, v24, pp. 51-76.
27. Atkins, M. H. (1998) "The Role of Appropriability in Sustaining Competitive Advantage – An Electronic Auction System Case Study," *Journal of Strategic Information Systems*, v7, pp. 131-152.
28. Hoffer, Jeffery A., George, Joey F., Valacich, Josephs, *Modern Systems Analysis and Design*, Addition-Wesley, 1999.
29. Flynn, Jim (1998) "Net Payment: The Marriage of Document Management and Electronic Commerce," *Inform*, Nov, pp.16-21.
30. Woo, Carson "Rules and Steps for Constructing Dependency Network Diagrams," *Lecture Notes*, 2000.

31. Woo, Carson "Analysis on ICBC Case," *Lecture Notes*, 1998.
32. Patel, Jeetu, Desai, Gautam, Bromberek, Jay (1998), "Bill presentment and payment hit the Web," *Information Week*, v709, pp. 19-22.
33. Hamilton, Tyler "Infant Firm Bets on Web Billing," *Globe and Mail*, 2000; Jan 13, www.globalandmail.com.
34. Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. "Introduction to Algorithms," McGraw-Hill, 1999.
35. Cheng, M. E. "Algorithm Complexity," *Lectures notes*, 1998
www-personal.monash.edu.au/~mecheng/mec3409/softeng/complexi/node8.html.
36. Description on Royal Bank's Online Banking Services, 2001,
<http://www.royalbank.ca/online/index.html>.
37. General Questions about Telus' Online Services, 2001,
<https://onlineservices.telus.com/>.
38. Rob, Peter, Coronel, Carlos "Database Systems: Design, Implementation, and Management," *Course Technology*, 1997.
39. Wright, Robert Grandford "The Nature of Organizations," Dickenson Publishing Company, Inc., 1977.
40. Wenhong, Luo et al. (2000); "An Exploratory Framework for Understanding Electronic Bill Presentment and Payment Model Selection," *Human Systems Management*; v 19, n3, pp. 255-265.
41. "EDI Basics," 1998, <http://www.telistics.com/edidoc2.htm>.
42. "Benefits of Using Chase Online Banking," 2001, www.chase.com.