

# Incomplete contracting issues in information systems development outsourcing \*

William B. Richmond  
and Abraham Seidmann

University of Rochester, Rochester, NY, USA

Andrew B. Whinston

University of Texas at Austin, Austin, TX, USA



**William B. Richmond** is Assistant Professor in the William E. Simon Graduate School of Business Administration at the University of Rochester. His main fields of research are economics of information systems and decision support systems. His articles have appeared in such journals as *ACM Transactions on Database Systems*, and *Journal of Economic Dynamics and Control*.



**Abraham Seidmann** is an Associate Professor and Areas Coordinator of Computers and Information Systems, Management Science and Operations Management at the William E. Simon Graduate School of Business Administration at the University of Rochester, Rochester, NY. He is a Department Editor on *Manufacturing and Automated Production in IIE Transactions*, and an Associate Editor for *Production Planning and Controls*, *Journal of Intelligent Manufacturing* and *International Journal of Flexible Manufacturing Systems*. Dr. Seidmann is the author of many research articles mostly published in such journals as *IIE Transactions*, *Operations Research*, *International Journal of Production Research*, *Management Science* and *IEEE Transactions*. His current research and consulting activities include Management and Analysis of Automated Production Systems, Computer Integrated Manufacturing (CIM), Management Information Systems, and Stochastic Models for Scheduling and Control.

\* This research was in part supported by the William E. Simon Graduate School of Business Administration, University of Rochester.

Correspondence to: William B. Richmond, William E. Simon Graduate School of Business Administration, University of Rochester, Rochester, NY 14627, USA

Outsourcing is the subcontracting of some or all the information systems functions by one firm to another. An incomplete contracting framework is used to examine the relative merits of outsourcing certain information systems development tasks. The focus is on investigating the effects of information asymmetry and different profit sharing rules on the decision of whether to outsource or to use an internal development team. The modeling indicates that the value generated from outsourcing the development effort comes primarily from the specific investments made by the external group, and that outsourcing dominates internal development when this investment is relatively more important than investments by the internal user group. This provides one economic explanation for the coexistence of both internal development teams and of various outsourcing services.

**Keywords:** Outsourcing, Contracting, Information systems development, Economics of information systems.

## 1. Introduction

The strategic role of many business information systems (IS), and the complexities of deploying and managing advanced software projects, have led to a growing number of concerns about managing corporate information systems [27]. These concerns are compounded by the large number of reports about systems failures and cost overruns [25]. Many problems facing the manage-

**Andrew B. Whinston** is Professor of Information Systems on the faculty of the college and Graduate School of Business, Department of Management Science and Information Systems at the University of Texas at Austin. His primary teaching interest is management information systems. His current research interests include data base management and applications of artificial intelligence to economics and management. He has also studied applied economics, regulatory economics, and accounting theory. Among his numerous publications, he has co-authored two books (with C. Holsapple and R. Bonczek), *Foundations of Decision Support Systems* (Academic Press, 1981), and *Micro Database Management - Practical Techniques for Application Development* (Academic Press, 1985). He has been a consultant to various companies, governmental agencies, and international organizations on data processing questions.

ment of information systems resources are organizational and are due to the large number of diverse participants and activities in the delivery and use of information [1]. Therefore, a major management issue is generating an organizational form that coordinates these participants and activities and that provides the proper incentives for delivering and using information successfully. Outsourcing certain IS functions is one such organizational form that IS managers are investigating to provide the incentives to reduce costs and improve service quality.

Outsourcing is the subcontracting of some or all of the IS development and processing activities to an outside vendor. Although much of the media's attention is directed at outsourcing facilities management, outsourcing encompasses all information processing activities including software development, systems integration, and network management. A recent survey by the Ledge-way Group, Inc. estimated that U.S. firms would spend about \$24 billion on outsourcing support activities for their IS groups in 1990. Almost 53% of that amount would be devoted to custom development, 25% to system integration and the rest to planning, design and education [17]. Additionally, a separate study found that facilities management outsourcing was a \$5.9 billion business in 1989, and is expected to grow to \$12.9 billion by 1995 [27]. Although outsourcing has existed in one form or another for many years (for example the proliferation of timesharing service bureaus in the 1960s), Kodak's recent announcements of its operating agreements with IBM, DEC and Businessland [24,28] have focused management's attention to this area.

This heightened awareness of outsourcing has prompted questions about its efficacy. In effect, all firms considering outsourcing are trying to decide the optimal degree of integration for their IS services. Identifying the functions to outsource and the functions to provide internally is an organizational design question and is the domain of the economic theory of organizations and markets.

The theory of organizations and markets is a subset of economics dealing with the structure of the firm or organization. It is based on the assumption that contracts and organizational form affect people's incentives to perform tasks. In IS research, the empirical and field study ap-

proaches generally take incentives as given or implicit. The theoretical implications of contracts on the IS function are similarly lacking. Notable exceptions are works by Mendelson [18], Gurbaxani and Kemerer [8], and Whang [30] who investigate incentive issues in setting transfer pricing policies for IS services. Despite the significant interest in IS outsourcing, there has been no theoretical research addressing this issue. Our paper is a first attempt at formally investigating some key organizational and contracting issues in the outsourcing decision.

We center our analysis in this paper around the *IS development effort* and the roles of incomplete contracts and outsourcing in this context. We do so because software development projects require a substantial amount of resources and can become a major source of disputes between the user and the developer communities [25]. These information system development projects are viewed as a series of exchanges between the *development group* and the *user group* which pays for the development effort. Each one of these transactions represents such steps as the definition, design or the delivery of a particular subsystem. Other transactions include enhancements and modifications to an existing application. Specific investments made by the user group and the development team at each stage of the project increase the net benefits of future transactions. Facilitating these specific investments is particularly important when dealing with customized information systems, because of the possible difficulty in monitoring the development effort and metering the final outputs.

In Section 2, we present an overview of some economic approaches to contract theory and define certain common contracting issues. It is noted, however, that Section 2 is neither a comprehensive survey, nor a critical overview of the area. Our intention is to provide enough background material to begin a preliminary investigation of certain contracting issues in the management of IS outsourcing. Readers interested in a more detailed list of references can refer to [7,11,12,26]. In Section 3, we develop an incomplete contracting model that explores the conditions under which users will find that outsourcing dominates internal development. Section 4 summarizes our conclusions and discusses topics for extending our research.

## 2. Contract economics concerns in the management of information systems

The contract theory approach was introduced by Coase [3]. Coase explains why resources are sometimes allocated via the market and sometimes via an entrepreneur. He conjectures that a firm exists because it lowers certain transaction costs. For example, the cost of negotiating and concluding a contract will be lower when a firm substitutes one contract for a series of contracts. Coase also conjectures that the existence of a firm may be based on the optimal length of a contract. He claims that the longer the contract, the more difficult and costly it is to specify the contracting parties' tasks. When short-term contracts are unacceptable, firms are likely to be established. Coase concludes that this is more important for services than for the production of goods.

Using contract theory as a foundation for studying the issues facing IS has three benefits. First, contract theory forms a basis for constructing models that provide new insights into the outsourcing issues facing the managers of information systems. These models may help us understand why a firm outsources its facilities management or internalizes some of its information systems development functions and outsources others. Contract theory will enable researchers to focus on the incentive issues involved in managing the IS function. Proper incentives are necessary for motivating and coordinating the numerous people involved in executing the IS functions, and these incentives play a significant role in facilitating the interaction between the user group and the IS development group [6,33].

Second, these models, predictions, and theories can then be tested by empirical, lab, and field studies. Combining the theoretical models with empirical validation provides important practical results for corporate and IS managers. Similar theoretical models have been subject to empirical tests for other areas of the firm. For example, the studies by Mian and Smith [20] explain why companies outsource certain accounts receivable functions; Klein, Crawford and Alchian [15] discuss the integration of GM and Fisher Body; Monteverde and Teece [22] measure transaction-specific skills and use those measures to explain backward integration in Ford and GM.

Third, since contracts and the organizational form are largely under management's control, we expect that understanding the impact of the contractual framework can help improve the management of IS resources.

Contracting and integration issues arise whenever a firm must decide whether to integrate a transaction<sup>1</sup> into its internal operations or to outsource the transaction. For the information systems function, transactions include developing, maintaining and operating various hardware and software systems, all of which can be either outsourced or executed internally. Three of the drivers of integration identified in the contracting literature are discussed by Perry [23]. These drivers are: Technological economies, market imperfections and transaction economies. *Technological economies* occur when the production frontier is greater if trades are executed internally rather than externally. There is conflicting evidence on the existence of economies of scale for various IS activities. For facilities management and information processing, there appear to be either technological diseconomies [13] or no economies [19]. For software development, Banker and Kemerer show that the existence of economies of scale depends on the company and the size of the software project [2]. *Market imperfections* arise from imperfect competition, externalities and imperfect or asymmetric information. Mendelson [19] examines the importance of asymmetric information in managing the IS function, but does not relate it to the outsourcing decision. *Transaction economies* occur when the cost of the exchange process is lower internally than in the market. These costs include those of writing and enforcing the contract, communication, coordination and monitoring, among others [12,32]. Our paper investigates the role of transaction economies in the development of customized software systems. This development process involves on-going, interrelated transactions and requires a high level of specific investments. Transaction economic models are well suited to addressing the impact of organizational form on the interactions among these transactions and the

<sup>1</sup> A transaction is an exchange between two or more parties. A transaction may be for a good, such as the purchase or lease of a computer, or for a service, such as the hiring of a contract programmer.

incentives for making optimal specific investments.

### 2.1. *Contracts*

A contract is an agreement between two or more parties. The agreement specifies the terms of a transaction between or among the parties. A transaction will occur only if there are perceived gains from trade (i.e., each party will agree to execute a transaction if and only if they believe it is in their best interest to do so). Typical contracts in our context are defined between the information systems department and the organization, an end-user group or a specific end-user; between an outside vendor and the organization, an end-user group or a specific end-user; or between the information systems department and an outside vendor. For new system development, the contract specifies the developer's responsibilities, the system's completion date, the system's cost, the end user's responsibilities, etc. For system execution, the contract specifies response time, cost per CPU second, cost per megabyte of storage, etc. The organizational form (intrafirm versus interfirm) within which the transaction is executed affects the possible terms in the contract. The terms of the contract may be implicit or explicit; they may be either subsumed or specified as part of a larger organizational contract, or they may be drawn up for the specific transaction.

To affect the agreement, the terms of the contract must be enforceable. For the terms of the contract to be enforceable, they must be verifiable. The contract's conditions are *verifiable* if a third party, frequently the courts, can determine whether a condition of the contract has been met. A condition in the contract is only *observable* if the contracting parties can decide if a condition of the contract has been met, but they cannot prove that the condition has been met. Thus verifiability is stronger than observability.

When the terms of the contract are based on output, the output must be measurable to be verifiable; however, many IS outputs are not easily measurable with current metrics (and therefore are not readily verifiable). For example, the quality of a system design report, user-friendliness across applications, and the system's maintainability are all examples of information system development outputs with characteristics that are

not measurable. In general, the literature distinguishes between two types of contracts: Comprehensive contracts and incomplete contracts. The nature and the merit of these approaches are evaluated briefly and are then discussed relative to incomplete contracting models in the context of IS management.

#### 2.1.1. *Comprehensive (complete) contracts*

A comprehensive (complete) contract specifies every possible contingency. It partitions the future states of the world. For each element of the partition, a comprehensive contract specifies the actions or payoffs for each party. Such a contract never needs to be revised or complemented. If such a contract can be written, then the contract's execution is independent of the institutional setting [10]. Because the agent's action is specified (explicitly, or induced by the prespecified payoffs) for every state of the world, the agent will take the same action in each state regardless of whether the agent is internal or external to the firm. For example, if a comprehensive contract can be written for software development, then the results should be the same if the developers are the internal IS group or an external vendor. Comprehensive contracts are rare in the IS world because it is difficult to specify every contingency beforehand. A contract for the purchase of computer supplies could be a possible example of a comprehensive contract.

#### 2.1.2. *Incomplete contracts*

Most business contracts are incomplete; when an unforeseen contingency arises, the contracting parties must bargain to resolve the conflict. In this sense, the contract must provide for arbitration and renegotiation, including early termination of the contract. Here, the institutional setting affects the bargaining process. A transaction executed between independent parties will allocate the gains from trading differently than a transaction executed within a company, where management has the right to force the execution or cancellation of a transaction.

For example, assume that a company is developing a customized production control information system. During the development process, an outside software house announces a low-cost, packaged version of that system. If the packaged version has sufficient functionality to meet the

requirements of the company, the company may consider aborting their development project and purchasing the newly announced package. The implications of this decision are strongly influenced by whether the production control information system is being developed internally by the IS department without a formal contract, or by an outside contractor. If an outside contractor is developing the system specifically for that corporation, then the company's decision to request the discontinuance of the development effort would have to consider the possibility of an expensive renegotiation of the contract. On the other hand, if the internal IS department is developing the system, cancelling the project typically would not result in expensive contract renegotiations. (Note, this does not imply, however, that all development should be internal.)

Many outsourcing contracts are incomplete, since rapidly changing technology and organizational environments make it impossible to specify every contingency in a contract. For example, the IBM-Kodak contract requires IBM to maintain a 'state-of-the-art' data center during the course of the contract. The contract does not specify the exact technology (e.g., the operating system version, disk storage technology or the communication systems interfaces) to be used during this time interval.<sup>2</sup>

## 2.2. *Incomplete contracting models*

Incomplete contracting models examine those cases where it is impossible or undesirable to govern transaction execution with a comprehensive contract. Under this umbrella, researchers have used different formalisms to explore various facets of the contracting relationship. According to Holmstrom and Tirole [10], two such incomplete contracting approaches are transaction cost economics (e.g., [32]) and property rights models (e.g., [7,9]).

Transaction cost economics focuses on the implications of different organizational forms when the transacting parties make specific investments. Specific investments are those investments that are most valuable in one specific setting or rela-

tionship, or as Perry [23] states: "Bilateral monopoly between a buyer and seller arises because gains from trade are enhanced by investments in assets which are specialized to their exchange. Williamson calls this 'asset specificity'. Asset specificity may arise from investments in: (1) specific physical capital; (2) specific human capital; (3) site specific capital; (4) dedicated capital; or (5) brand name capital. Such transaction specific assets give rise to what Klein, Crawford and Alchian [15] call 'appropriable quasi-rents', which are the difference between the value of the asset in this use and the value in its next best use." For example, the time a system analyst spends learning about a user department's operating procedures, business priorities, and decision modes is a specific investment. This specific investment increases the system analyst's value when the analyst continues to develop systems for this particular user department. It is of less value when the analyst later develops systems for other departments with different operating procedures, business priorities, and decision modes. The existence of specific investments can lock the participants into a particular relationship. This relationship should generate a surplus value (the value from trading within the relationship minus the value from trading outside the relationship) from the specific investments made earlier. How this surplus value is divided depends on each party's bargaining power, which depends on the organizational form.

In the property rights literature (e.g., [7,9]), organizational form is defined by the ownership of the productive assets. Asset ownership gives the owner certain rights, such as the right to use an asset, the right to deny someone else the use of the asset, and the right to sell the asset. The asset's owner may voluntarily give some rights away. The rights that the owner does not specifically give away in a contract are termed the owner's residual rights. Asset ownership increases bargaining power, thus affecting the division of the gross surplus (the value of the trade minus the cost of the trade). The organizational form does not affect who bears these costs and who receives the value. (The organizational form may affect who bears the costs, but the property rights literature generally abstracts from these considerations without explicitly addressing this issue.) It only affects how the gross surplus is

<sup>2</sup> Private discussions with Frank Palm, IBM Project Executive—Eastman Kodak facilities (1990).

divided. The division of the surplus affects the level of the relationship-specific investment. According to Grossman and Hart [7], the transaction's costs and benefits, and the specific investment costs, are inalienable to particular trading participants. A typical goal in this line of research is to determine the socially optimal ownership structure, which allocates ownership rights to minimize the ex ante investment distortions.

The incomplete contracting approach presented by Holmstrom and Tirole [10] is similar to the property rights approach described above. They focus on the effect of organizational form on ex ante investments. In these cases, the financial streams are transferred with ownership rather than being inalienable to the respective decision makers, so decision rights about an asset's use are tied to the residual return stream from using the asset (although, as noted, incentive contracts can transfer some of the return stream without transferring asset ownership).

The pertinent question of this paper is the impact of these contracting issues on specific investments and the outsourcing decision. In the next section, we extend Holmstrom and Tirole's framework to an information system development environment and use it to examine the effects of asymmetric information and of different sharing rules when determining the optimal amount of specific investment and whether to outsource at all. We also examine whether negotiation between the user group and the developer group could lead to a particular sharing arrangement.

### 3. IS development and incomplete contracting

As noted in the introduction, one popular outsourcing activity is systems integration. A recent *Computerworld* article [14] cites several management concerns regarding outsourcing system integration projects. One of these concerns is that systems integrators do not really understand the company's way of doing business; therefore they do not really understand what they have been asked to take on. Clearly, this concern has a basis in transaction cost economics and can be partially alleviated by inducing the users and the vendors to make appropriate specific investments. Inducing these specific investments is one of the most difficult issues associated with establishing work-

ing relationships between service providers and users.

In this section we model a contracting policy for IS development aimed at inducing the user and development groups to make the specific investments needed to alleviate the concern stated above. We examine two main options for IS development: Using an internal development team that operates under direct control of the user group, and outsourcing the development effort to an external vendor. In Section 3.1, we introduce the modeling framework and motivate its use. In Section 3.2, we describe the details of the contracting model, which is specialized in Section 3.3 by assuming that the value and the cost of software enhancements are common knowledge to the user and development groups. Under this assumption, we analyze the decision to use an internal versus external development group. In Section 3.4, we assume that the user group knows the actual value of an enhancement, but not its cost, and the development group knows the cost of the enhancement, but not its value to the users.

#### 3.1. *The modeling framework: Introduction and motivation*

Our modeling centers around the IS development effort, because software development projects require a substantial amount of resources and at times become a major source of disputes between the user and the developer communities. Any information system development project implies a series of exchanges between the development group and the users who pay for the development effort. These exchanges must be properly governed to minimize the associated agency costs.<sup>3</sup> Two commonly used management strategies are internal development and outsourcing. In this paper, we model and examine the effects of different profit sharing rules on the decision of whether to outsource or to use an internal development team.

<sup>3</sup> Agency costs are the costs of contracting an agent to perform a task. Agency costs include those of monitoring the agent, bonding and the residual loss incurred when the cost of fully enforcing the contract exceeds the benefits. The agent can be either an employee or an independent contractor [12].

Consider a *user group* contemplating the installation of an information system for a new business function. This system will be designed and implemented either by an *internal or external development group*. The contract between the user and the development groups must specify the initial scope of the information system and the terms for compensating the development group. The contract also must contain provisions for specifying how to decide whether to implement any future enhancements or modifications and for sharing the gains from the implemented enhancements and modifications. As is generally the case with information systems, design changes can be expected after the contract for the initial system is finalized. These changes can occur during the system development process or after system installation. At the early stages of the system development process, it is impossible to identify all these changes, their value, or their cost. Because of these reasons, the user group cannot contract for the complete information system (the information system resulting from the initial development plus all future enhancements). Similarly, the development group may be reluctant to agree to a fixed time or price contract that allows the users to specify future enhancements or modifications loosely.

Specific investments made by the user group and the development group can raise the chance of identifying high-value enhancements and can lower the cost of implementing the enhancements. These enhancements frequently account for a significant portion of the information systems' life cycle costs [5], so identifying high-value enhancements and lowering the costs of implementing these enhancements is critical to efficient and effective IS operations. For instance, consider a bank developing a financial information system aimed at managing their checking accounts. The development group can make specific investments in formulating a monolithic data model that takes into account the bank's view of the depositors' financial transactions in checking, savings, personal loans and other related areas. Similarly, the user group can make specific investments in understanding the data model covering all related areas of retail banking. Learning this data model provides the user group with an understanding of the interrelationships among all the various banking applications. This knowledge

(generated by the user group's specific investment) is useful for identifying and defining useful future enhancements, such as an integrated customer (rather than account) risk management system. If the development group is already familiar with the data models spanning the other related application areas (as a result of making their specific investment as mentioned earlier), then implementing such integrated enhancements in the future will entail lower development costs.

The organizational form and the sharing rule both affect whether an enhancement will be implemented. The organizational form determines who can veto a design change. We explore two options. One option is to employ a salaried, user-controlled, in-house development team to construct the information system. The other option is to outsource the development effort to an external development team. The external development team is assumed to be incentive driven.<sup>4</sup> Under user-controlled, in-house development, the user decides unilaterally whether to implement an enhancement. The internal development group must comply with the requests for enhancements made by the users. Under outsourcing, both the user and the developer must agree on making an enhancement. If the users feel that the value of the enhancement is too low, they can veto its implementation. Likewise, if the developer feels that the costs are too high, they can refuse to implement the enhancement.

The sharing rule specifies the division of any monetary surplus between the user group and the development group. The value to the user group of implementing an enhancement is denoted by  $v$  and the cost to the development group of implementing an enhancement is denoted by  $c$ . The gross surplus is  $v-c$ . The sharing rule divides this gross surplus between the user and the development groups. The analysis presented below explains how the sharing rule affects whether an enhancement will be implemented by influencing the level of specific investment made by both the users and developers. Additionally, when there is *asymmetric information*, the sharing rule affects

<sup>4</sup> This is a typical situation for many system development and system integration companies. The high labor intensity of their services mandates the use of measures that tie their costs to performance: Private communication with Professor Kevin Murphy, Harvard Business School.

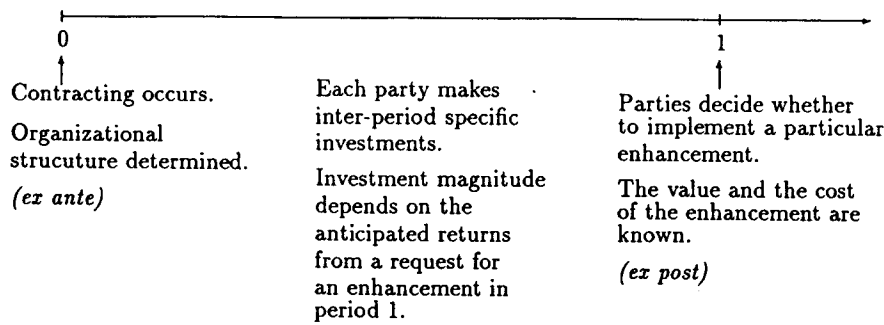


Fig. 1. Contracting and development time line.

the expected value to the development group and to the user group of concealing the true cost or value of the enhancement. Misrepresentation of the parameter values can result in omitting net positive-valued enhancements.

We assume a case where the initial cost of developing the system by the internal IS group would be the same as outsourcing the system development to an independent software house.<sup>5</sup> Under this condition, the optimal organizational form depends on the contracting and agency costs.

### 3.2. Model

The enhancement's value,  $\nu$ , may turn out to be either high ( $\nu_H$ ) or low ( $\nu_L$ ). The cost,  $c$ , of implementing the enhancement may be either high ( $c_H$ ) or low ( $c_L$ ). By investing added effort, the user group personnel can increase the probability that an enhancement will have the higher value,  $\nu_H$ . Similarly, the development group can make specific investments to increase the probability that any future enhancement will entail a low cost,  $c_L$ .

The user group's investment costs are given by  $f(x)$ , where  $x = \Pr(\nu = \nu_H)$ . The developer's investment costs are given by  $g(y)$ , where  $y = \Pr(c = c_L)$ . We define  $f(x)$  and  $g(y)$  as strictly increasing, convex functions; therefore, increasing the probability of a high-value or a low-cost enhancement is increasingly expensive. For simplicity of exposition, we assume that  $f(x)$  and  $g(y)$

<sup>5</sup> Our model can easily be expanded to consider the case where the initial cost of internal development differs from the cost of external development. If the cost of the initial system differs, but the cost of enhancements are the same, then the value of outsourcing need only be adjusted by this constant.

are independent, so the total cost of the specific investments made for an enhancement is  $f(x) + g(y)$ .

We use a two-stage model to capture the sequential nature of the development process. The contracting and development process model is illustrated in fig. 1:

- In period 0, the parties determine the sharing rule and organizational form. The developer and the users know the functional form of  $f(x)$  and  $g(y)$ , and they know the values  $c_H$ ,  $c_L$ ,  $\nu_H$  and  $\nu_L$ , but not the value or cost for any specific enhancement.

- Next, the system development process starts and each party makes specific investments to increase the expected net value of future enhancements to the system.

- In period 1, the users identify an enhancement opportunity and the parties decide whether to implement that enhancement. The gross surplus from implementing any enhancement is then divided between the developer and the end user.

The sharing rule allocates  $\alpha$  portion ( $0 \leq \alpha \leq 1$ ) of the resulting gross surplus ( $\nu_i - c_j$ ) to the user group, and the remaining portion ( $1 - \alpha$ ) of the gross surplus to the development group. This sharing rule leads to contracts of the form<sup>6</sup>

$$\text{User share} = \alpha(\nu_i - c_j), \quad i, j \in \{H, L\}, \quad (1)$$

<sup>6</sup> As noted by one of the anonymous referees, the model in 1 and 2 can be generalized by adding a constant transfer payment,  $\delta$ , from one company to the other, possibly in the form of under-pricing fixed costs services. Using this contracting form, an external development group can pay  $\delta$  to buy the franchise to develop the system. The  $\alpha$  level can then be set to maximize the total surplus ( $\nu_i - c_j - f(x) - g(y)$ ). Now, the transfer payment,  $\delta$ , must be negotiated. We assume that  $\delta = 0$ , because we know that Andersen Consulting, Co. uses sharing contracts similar to that described in equations (1) and (2).

$$\text{Developer share} = (1 - \alpha)(v_i - c_j),$$

$$i, j \in \{H, L\}. \quad (2)$$

This sharing rule is motivated by the practices of Andersen Consulting Co., a leading vendor of customized software and software integration projects. This vendor divides both the financial gains and losses of certain systems integration projects with its customers [14]. We use this sharing rule to emphasize the importance of incentives for the external development group.

The value of  $\alpha$  is an index of the user group's relative bargaining power. The value of  $\alpha$  approaches 1 when the user group has more bargaining power, which is the case under user-controlled, in-house development. Under outsourcing, the bargaining power will be more evenly distributed, so we would expect a lower value of  $\alpha$ . Given this contracting scheme, outsourcing will occur only when the user group's net payoff (its share of the gains minus the cost of its specific investment) is higher under outsourcing than under internal development ( $\alpha = 1$ ).

### 3.3. Common knowledge of $v$ and $c$

We begin with the simplest case. In Period 1, the actual cost and value of the enhancement ( $v_i$  and  $c_j$ ;  $i, j \in \{H, L\}$ ) are realized, and their realizations are common knowledge to the user and development groups. This information is used when determining whether to implement a particular enhancement. This provides a base case and enables us to assess the effect of reducing information asymmetry on organizational form. It also provides a first-order approximation to the case where the user group and the development group are experienced in developing information systems, and can provide reasonable estimates of the other party's parameter value. Common knowledge of the cost and value precludes either party from lying about its true parameter value.

In this case, the gross expected gain from implementing an enhancement at time 0 is given by

$$\begin{aligned} E\{G(x, y)\} = & \max(0, [v_H - c_L])xy \\ & + \max(0, [v_L - c_L])(1-x)y \\ & + \max(0, [v_H - c_H])x(1-y) \\ & + \max(0, [v_L - c_H])(1-x) \\ & \times (1-y). \end{aligned} \quad (3)$$

US, the user group's portion of the expected surplus minus its investment costs, is given by

$$US \equiv \alpha E\{G(x, y)\} - f(x), \quad (4)$$

and DS, the development group's portion of the expected surplus minus its investment costs, is given by

$$DS \equiv (1 - \alpha)E\{G(x, y)\} - g(y). \quad (5)$$

The privately optimal specific investment levels are determined from

$$\frac{\partial}{\partial x} US = \frac{\partial}{\partial x} [\alpha E\{G(x, y)\} - f(x)] = 0, \quad (6)$$

and

$$\frac{\partial}{\partial y} DS = \frac{\partial}{\partial y} [(1 - \alpha)E\{G(x, y)\} - g(y)] = 0, \quad (7)$$

for a given value of  $\alpha$ .

Since in this model the internal development employees are salaried, all gains from implementing an enhancement accrue to the user group. This constitutes the threat point from which the user group can bargain with an external development group.<sup>7</sup> Outsourcing is feasible if it generates a positive surplus for the development group and the user's surplus is greater than under internal development:  $DS \geq 0$  and  $US \geq US_{\alpha=1}$ . If we allowed  $\delta > 0$ , then outsourcing would be preferred whenever the total surplus was greater than the surplus generated under internal development.

Given our description of the problem, we partitioned the feasible state space into six general cases, as shown in fig. 2. For the most part, the closed-form analytic solutions were found to be complex and provided no insights. For this reason, we conducted a series of numerical simulations which provide insights by examining the effects of different parameter values on the nature of the outsourcing decision. For each of the six cases in fig. 2, we examined between 81 and

<sup>7</sup> If the internal development group's employees were to receive bonuses, incentives or side-payments based on their work's net value, then the difference between using an internal development group and outsourcing would arise from the difference in their bargaining power. Under the contracting schemes that we are considering ( $\delta = 0$ ), this difference would be reflected by different  $\alpha$  levels.

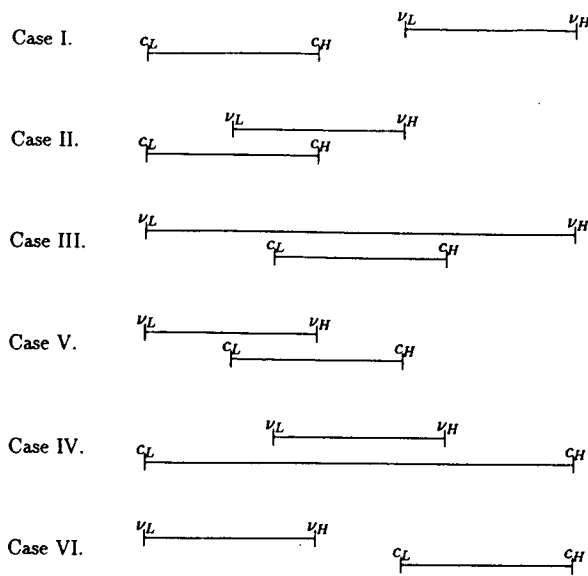


Fig. 2. General cases relating costs to enhancement values.

243 subcases. The specific investment cost functions were of the form  $ax^2$  and  $by^2$ ,  $a, b \in \{10, 250, 500\}$ . The highest value ( $v_H$  or  $c_H$ ) was held constant at 310. The second highest value varied from 210–310. The third highest value

varied from 110–290, and the lowest value in each case varied from 10–280. The difference between each adjacent value varied from 10–100. For each case, we calculated the user surplus, the developer surplus, the social surplus and the optimal specific investment levels ( $x$  and  $y$ ) for  $\alpha$  between 0 and 1 in 0.01 increments. The parameter values used were arbitrarily chosen to examine a wide range of scenarios. In reality, the parameter values will not be equally likely, and the likelihood of each set of parameter values will affect the likelihood of internal and external development.

### 3.3.1. Case I

In Case I, an enhancement will always be made, because the value of an enhancement will always exceed its costs (see fig. 2). The details of the closed-form results from this case are given in table 1. The results indicate that the user group's specific investment,  $x_\alpha^*$ , increases linearly with their share of the gains. The user group's specific investment also increases as the relative value of their investment to the cost of their investment increases (i.e., as  $(v_H - v_L)/2a$  increases). Similarly, the development group's specific invest-

Table 1

Case I: Closed-form analytical results under common knowledge.

$$E\{G(x, y)\} = (v_H - v_L)x + (c_H - c_L)y + (v_L - c_H)$$

$$US = \alpha[(v_H - v_L)x + (c_H - c_L)y + (v_L - c_H)] - ax^2$$

$$DS = (1 - \alpha)[(v_H - v_L)x + (c_H - c_L)y + (v_L - c_H)] - by^2$$

$$x_\alpha^* = \min\left(\frac{\alpha(v_H - v_L)}{2a}, 1\right)$$

$$y_\alpha^* = \min\left(\frac{(1 - \alpha)(c_H - c_L)}{2b}, 1\right)$$

$$\alpha^* = \frac{2b(v_H - c_H)^2}{2b(v_H - c_H)^2 + 2a(c_H - c_L)^2}$$

$$\text{Outsource} \Leftrightarrow \left\{ \frac{(v_H - v_L)^2}{2a} + \frac{(c_H - c_L)}{2b} + (v_L - c_H) < \frac{(c_H - c_L)^2}{b} \right\}$$

$E\{G(x, y)\}$  = the expected value of making an enhancement at period 0.

US = the user's expected surplus at period 0.

PS = the developer's expected surplus at period 0.

$x_\alpha^*$  = user's optimal investment level for a given  $\alpha$ .

$y_\alpha^*$  = developer's optimal investment level for a given  $\alpha$ .

$\alpha^*$  =  $\alpha$  level that leads to the highest total surplus.

Outsource = the condition under which outsourcing dominates internal development

Table 2  
Example 1. Internal development dominates outsourcing.

$f(x) = 100x^2$	$g(y) = 500y^2$
$\nu_H = 310$	$\nu_L = 210$
$c_H = 200$	$c_L = 150$

ment,  $y_\alpha^*$ , increases linearly with its share of the gains  $(1 - \alpha)$ , and it increases as the value of their investment increases relative to the cost of their investment (i.e., as  $(c_H - c_L)/2b$  increases).

We found that the decision to use a salaried, internal development group rather than co-venture with an outside contractor depends on the relative values of  $\nu_H$ ,  $\nu_L$ ,  $c_H$ ,  $c_L$ ,  $f(x)$  and  $g(y)$ . The exact relationship is shown in the equation in table 1, but basically, outsourcing dominates when the value of the development group's investment relative to the cost of its investment is larger than the relative value of the user group's investment to its costs. Given  $f(x) = ax^2$  and  $g(y) = by^2$  for a given  $f(x)$  and  $g(y)$ , the larger  $c_H - c_L$  relative to  $\nu_H - \nu_L$ , the more likely it is that outsourcing will produce a larger user surplus. (Note that for this case we found that the social surplus (US + DS) was invariably higher under outsourcing (when  $\alpha = \alpha^*$ ). For a given  $\nu_H$ ,  $\nu_L$ ,  $c_H$ , and  $c_L$ , a 'small'  $g(y)$  relative to  $f(x)$  is also more likely to result in outsourcing producing a larger user surplus than internal development. We present two numerical examples illustrating these points: The first example depicts a case where internal development dominates outsourcing, and the second example depicts a case where outsourcing dominates internal development.

Table 2 contains the parameter values for Example 1 where internal development dominates outsourcing. In this example, the user surplus under internal development is US = 35.00. Under outsourcing, the maximum user surplus is US = 34.43 at  $\alpha = 0.99$ , so internal development dominates outsourcing. Figure 3a shows the user surplus, US, and the developer surplus, DS, for  $\alpha$  ranging from 0 to 1. The user surplus increases monotonically with  $\alpha$ . Initially, the developer surplus increases even though the developer's share of the gains decreases. This happens because the user's investment is increasing and this investment increases the gross surplus faster than the decrease in developer's share. The value of the

development group's surplus peaks at  $\alpha = 0.38$ . At greater values of  $\alpha$ , even though the user group's investment is still increasing, it does not raise the gross surplus enough to compensate for the decrease in the development group's investment and its decreasing share of the gains.

Table 3 contains the parameter values for Example 2 where outsourcing dominates internal development. In this example, the user surplus is US = 10.25 under internal development. Outsourcing can provide a user surplus as great as 18.09. Outsourcing dominates internal development for  $\alpha$  ranging from 0.21–0.99. For this range of  $\alpha$  values, the user surplus increases from 10.41 at  $\alpha = 0.21$  to 18.09 at  $\alpha = 0.60$  and then decreases to 10.64 at  $\alpha = 0.99$ . The developer's surplus drops monotonically with  $\alpha$  from 35.00 ( $\alpha = 0$ ) to 0.11 ( $\alpha = 0.99$ ) (see fig. 3b). Note in fig. 3b that the value of US does not increase monotonically with  $\alpha$ : Beyond a certain threshold point ( $\alpha = 0.60$  in this case), the value of US starts declining. This happens because higher values of  $\alpha$  represent a reduced incentive for the developer and thus a much greater probability of realizing  $c_j = c_H$  values.

### 3.3.2. Case II

In Case II, an enhancement will *not* be implemented if and only if  $c_j = c_H$  and  $\nu_i = \nu_L$ . We found that outsourcing predominates most of the simulations conducted for this case; however, internal development also occurred relatively frequently; although this may be an artifact of the parameters we used. For a given  $\nu_H - \nu_L$  and  $c_H - c_L$ , internal development was more likely to dominate outsourcing when the user group's cost of making specific investments was relatively lower than the development group's cost of making specific investments. This occurs when the development group's investment is low, even when  $\alpha = 0$ . As  $\alpha$  increases, the user's share of the payoff increases faster than the total gains decrease, so the user is better off developing the

Table 3  
Example 2. Outsourcing dominates internal development.

$f(x) = 100x^2$	$g(y) = 100y^2$
$\nu_H = 310$	$\nu_L = 300$
$c_H = 290$	$c_L = 190$

system internally. Additionally, for a given  $v_H - v_L$ , as  $c_H - c_L$  decreased, internal development dominated external development for a larger

range of user and developer group specific investment cost functions. As  $c_H - c_L$  increased, internal development dominated external develop-

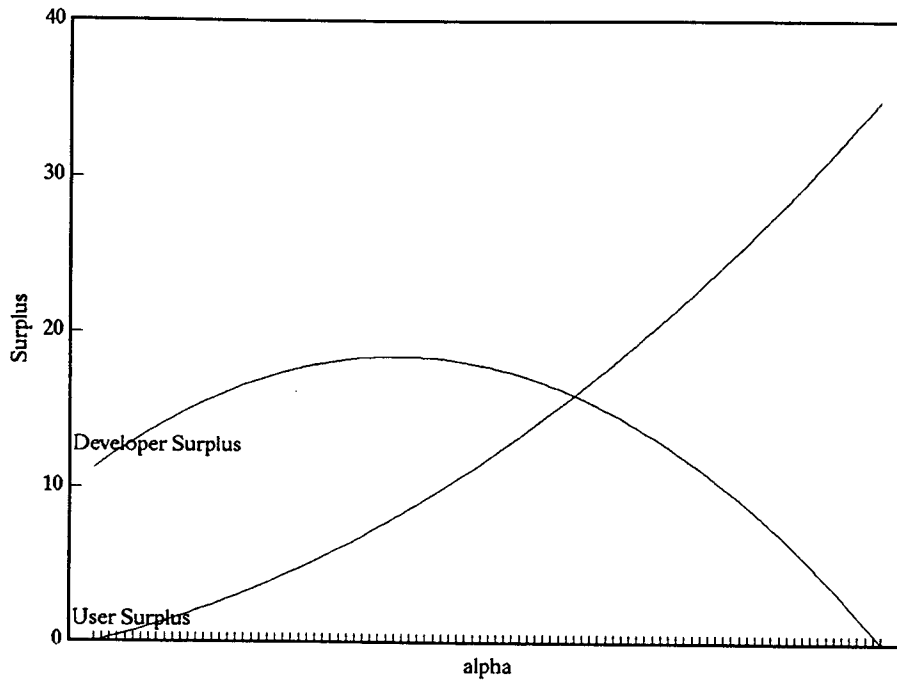


Fig. 3a. Internal development dominates outsourcing.

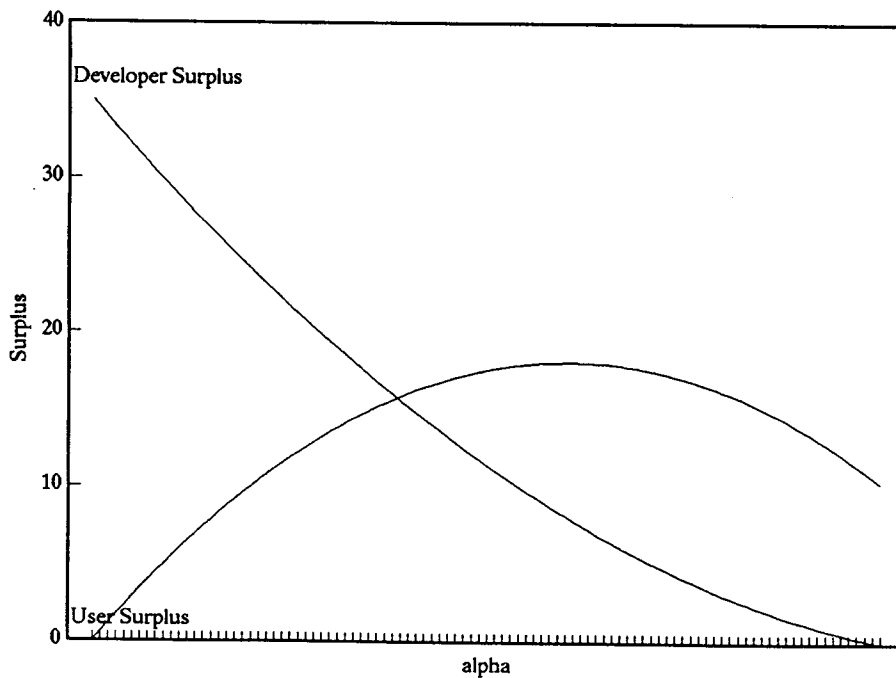


Fig. 3b. Outsourcing dominates internal development.

ment for a smaller range of user specific investment cost functions; the same pattern was observed for the developer specific investment cost functions. *These results imply that in this case internal development is more likely as the relative value of the user's specific investment increases.*

### 3.3.3. Case III

In this case, an enhancement is implemented if and only if it has a high value (i.e.,  $v_i = v_H$ ). We found that internal development dominates outsourcing in most of the simulations, because the user must make a specific investment for any gains to occur. To induce a high specific investment requires that the user receives a large portion of the gains. Outsourcing dominated internal development when both  $c_H - c_L$  was large relative to  $v_H - c_H$  and both parties' costs of making specific investments were low. Interestingly, for a given  $f(x)$ ,  $g(y)$  and  $v_H - c_H$ , as  $c_H - c_L$  increased, the smallest  $\alpha$  that makes outsourcing feasible first increased, and then decreased. In these cases, both party's specific investment level was zero until a critical threshold point was reached, after which both parties started to invest.

### 3.3.4. Case IV

Here an enhancement will only be implemented when its cost is low ( $c_j = c_L$ ). Therefore, it is always in the user group's best interest to outsource, since the development group must make some specific investment for any gains to occur. Determining the optimal sharing rule, however, can be very difficult as indicated in fig. 4.

For example, fig. 4 depicts both the user group's surplus and the development group's surplus as a function of  $\alpha$ , for  $f(x) = 10x^2$ ,  $g(y) = 10y^2$ ,  $v_H = 210$ ,  $v_L = 160$ ,  $c_H = 310$  and  $c_L = 150$ . A Nash bargaining solution that maximizes the product of the user group's and the development group's surplus is reached at  $\alpha = 0.80$ . At  $\alpha = 0.80$ , the user group's surplus is 18.8, and the development group's surplus is 3.6. In this case, the user group receives its maximum user surplus, but the development group receives 36.8 less than its maximum developer surplus. In such a situation, it is conceivable that even though outsourcing is optimal, the user group and the development group will be unable to negotiate a sharing rule that is acceptable to both parties. In these cases in particular, expanding the contracting

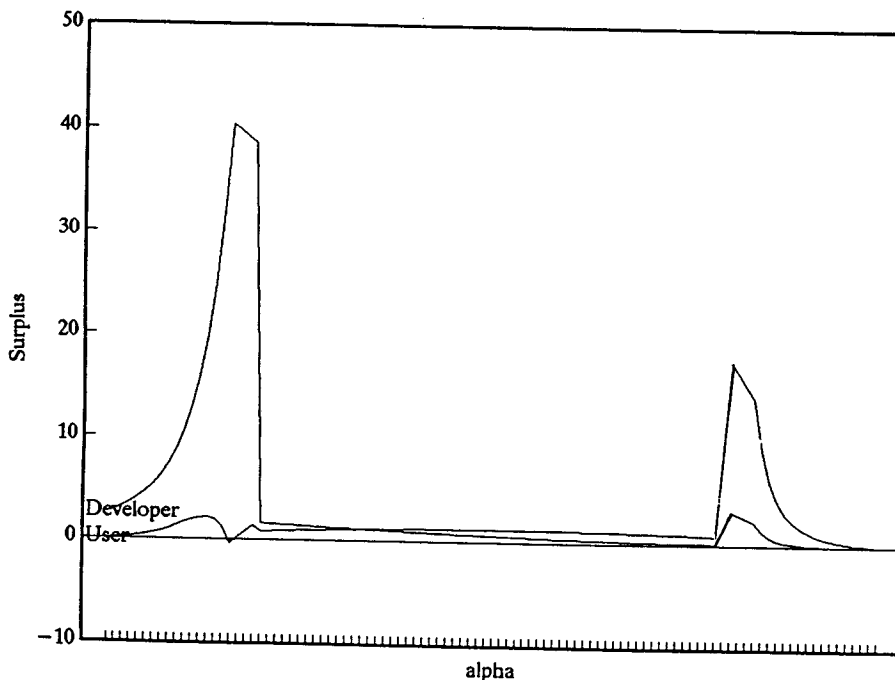


Fig. 4. User surplus and developer surplus for Case V.

scheme to include transfer payments ( $\delta > 0$ ) could make reaching an optimal agreement easier, and would make both parties better off. If transfer payments were allowed, the  $\alpha$  level could be set to maximize the total net surplus. The transfer payment would ensure that the user group received an amount that would at least cover the value of internal development. Such a problem was also identified in some of the Case I examples we studied. Other examples of Case IV exhibit a unimodal US function and a monotonic PS function similar to the one depicted in fig. 3b.

### 3.3.5. Case V

An enhancement will be implemented here only when its cost is low and its value is high ( $c_j = c_L$  and  $v_i = v_H$ ). *Outsourcing dominates in this case*, because  $\alpha = 1$  implies that the development group will make no specific investment ( $y = 0$ ), so  $c = c_H$ .

### 3.3.6. Case VI

In Case VI, no enhancements will ever be made, since the cost of the enhancement will always exceed the value of that enhancement (see fig. 2).

## 3.4. Asymmetric information

When asymmetric information conditions exist, both parties know the potential value of  $v_H$ ,  $v_L$ ,  $c_H$  and  $c_L$  in period 0. However, in period 1, the user group knows the actual value,  $v_i$ , but not the actual cost  $c_j$ ; the development group knows the actual cost,  $c_j$ , but not the value of the enhancement to the user,  $v_i$ . We assume that the ex post cost to the developer and the ex post value to the user are not verifiable and therefore cannot be contracted. At period 1 (ex post), the users know with certainty the value of the enhancement (i.e., whether  $v_i = v_H$  or  $v_i = v_L$ ). The developers know with certainty how much implementing the enhancement will cost them (i.e., whether  $c_j = c_H$  or  $c_j = c_L$ ). They must now decide whether to implement the enhancement. This section explores the impact of asymmetric information between the user and the development groups on: The outsourcing decision, the user surplus, and the developer surplus.

Under asymmetric information, evaluating the sharing rules is more difficult than when  $v$  and  $c$  are common knowledge. The expected gains from implementing an enhancement are evaluated below on a case by case basis. We follow the same six general cases of fig. 2.

### 3.4.1. Case I

In period 0 (ex ante), both parties already know that  $v_L > c_H$ ; therefore, an enhancement will always be implemented. In the case of a salaried, internal development group ( $\alpha = 1$ ,  $y = 0 \Rightarrow c_j = c_H$ ) the user surplus can be derived from equations 3 and 4 as

$$\begin{aligned} US_{\alpha=1} &= (v_H - c_H)x + (v_L - c_H)(1-x) - f(x) \end{aligned} \quad (8)$$

$$= (v_H - v_L)x + (v_L - c_H) - f(x). \quad (9)$$

If the system is outsourced, both the user and the development groups may have an incentive to misrepresent their parameter values. In this case, we expect that the user group will always claim that an enhancement's value is low, and the development group will always claim that an enhancement's cost is high. This misrepresentation enables the user group to capture  $v_H - v_L$  when  $v_i = v_H$ , and the development group to capture  $c_H - c_L$  when  $c_j = c_L$ . Here we get

$$\begin{aligned} US_{\alpha \neq 1} &= \alpha(v_L - c_H)xy + (v_H - v_L)xy \\ &\quad + \alpha(v_L - c_H)(1-x)y \\ &\quad + \alpha(v_L - c_H)x(1-y) \\ &\quad + (v_H - v_L)x(1-y) \\ &\quad + \alpha(v_L - c_H)(1-x)(1-y) - f(x) \end{aligned} \quad (10)$$

$$= [\alpha(v_L - c_H) + (v_H - v_L)]x + \alpha(v_L - c_H)(1-x) - f(x) \quad (11)$$

$$= (v_H - v_L)x + \alpha(v_L - c_H) - f(x). \quad (12)$$

In this case,  $US_{\alpha=1} > US_{\alpha \neq 1}$ . Based on equation (6), the user group's specific investment,  $x$ , is the same under outsourcing as under internal development, so  $US_{\alpha=1} - US_{\alpha \neq 1} = (1 - \alpha)(v_L - c_H) >$

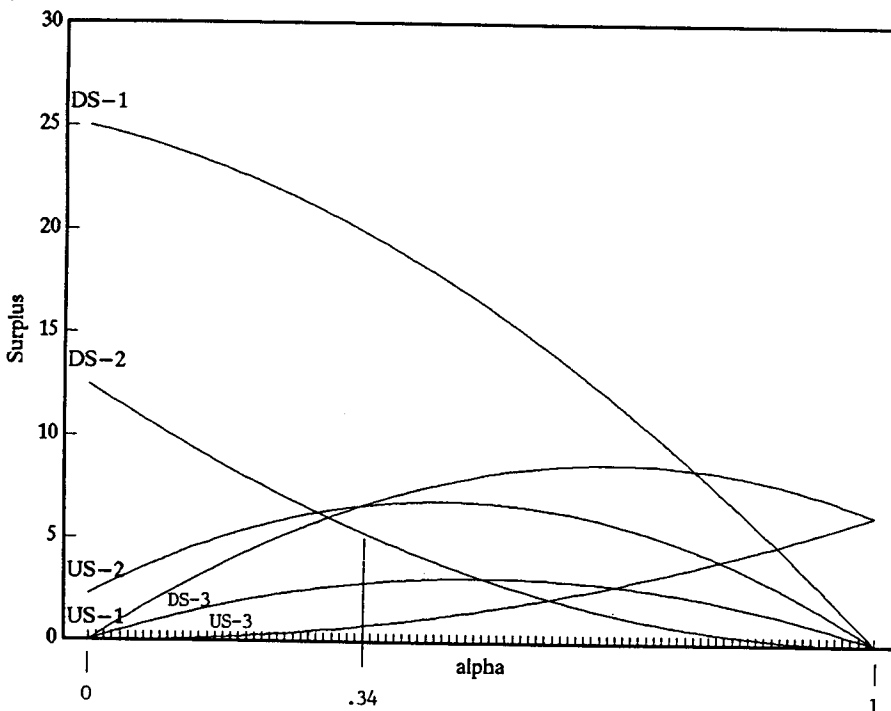
0; therefore, *internal development dominates outsourcing*.

### 3.4.2. Case II

In this case, the decision to implement an enhancement will depend on the values ( $v_i$  and  $c_j$ ) *declared* by the user and the developer. The declared values may not necessarily be the *actual* values. Because either or both parties may lie, some positive net-valued enhancements may not be implemented (e.g., if  $v_i = v_H$  and  $c_j = c_L$ , but the declared values are  $v_L$  and  $c_H$  the enhancement will not be implemented, even though it has a positive net value). An enhancement will *not* be implemented if and only if the *declared* values are  $c_j = c_H$  and  $v_i = v_L$ . The user group has an incentive to declare  $v_i = v_L$  if the value of an enhancement is high ( $v_i = v_H$ ), *and* if the user believes that the cost of the enhancement is low ( $c_j = c_L$ ), *and* if the user believes that the devel-

oper will tell the truth about the cost of the enhancement. In this case, the user group confiscates  $v_H - v_L$  of the gains. Similarly, the development group has an incentive to report  $c_j = c_H$  if the cost of an enhancement is low ( $c_j = c_L$ ), *and* if the developer believes that the value of the enhancement is high ( $v_i = v_H$ ), *and* if the developer believes that the user will tell the truth about the value of the enhancement. In this case, the development group confiscates  $c_H - c_L$  of the gains.

The value of outsourcing depends upon the strategy that the user and developer take for revealing their parameter values. Determining the value of outsourcing requires solving a single-stage, two-player game. This game represents the user and development groups' strategies for revealing their parameter values. The expressions for the payoffs from these strategies are functions of  $v_i$ ,  $c_j$ ,  $\alpha$ ,  $x$ ,  $y$ ,  $a$  and  $b$ . These expressions turn



- DS-1 Developer surplus when both the user and developer tells the truth.
- US-1 User surplus when both the user and developer tells the truth.
- DS-2 Developer surplus when the developer tells the truth and the user lies.
- US-2 User surplus when the developer tells the truth and the user lies.
- DS-3 Developer surplus when the user tells the truth and the developer lies.
- US-3 User surplus when the user tells the truth and the developer lies.

Fig. 5. User and developer surplus for different truth-telling strategies.

out to be lengthy and complex, and instead of presenting them, we chose to use a simulation approach similar to the one conducted earlier in this paper. The simulation results indicate that: (a) when internal development was optimal under common knowledge, it is also optimal under asymmetric information; (b) when outsourcing was optimal under common knowledge, it may also be optimal under asymmetric information; and (c) in the case of outsourcing, the truth-telling policy varies with  $\alpha$  for a given data set. For example, fig. 5 shows that for  $\alpha < 0.34$  the optimal strategy is for the user group to misrepresent its parameter value, and for the developer to declare its parameter value truthfully. When  $\alpha > 0.34$ , the optimal strategy is for both the user and development groups to tell the truth.

### 3.4.3. Case III

Here, an enhancement will be implemented if and only if it has a high value ( $v_i = v_H$ ). Using a salaried internal development group ( $\alpha = 1 \Rightarrow y = 0 \Rightarrow c_j = c_H$ ) the user surplus obtained from equation (4) is

$$US_{\alpha=1} = (v_H - c_H)x - f(x). \quad (13)$$

Under outsourcing, in each and every instance, the development group has the incentive to declare that its actual cost is high, ( $c = c_H$ ). The user surplus is

$$US_{\alpha \neq 1} = \alpha(v_H - c_H)x - f(x), \quad (14)$$

which is less than or equal to using internal development. Therefore, *internal development dominates outsourcing*, since it provides a greater user surplus. (An argument parallel to the one in Section 3.4.1 shows that the user group's specific investment will be the same under outsourcing and under internal development.)

### 3.4.4. Case V

Here an enhancement will only be implemented when its cost is low ( $c_j = c_L$ ). The development group lacks an incentive to misrepresent its actual costs, but the user group has the incentive to claim that all enhancements have a low value ( $v_i = v_L$ ). Using a salaried, internal development group ( $\alpha = 1 \Rightarrow y = 0 \Rightarrow c_j = c_H$ ) no investment will be implemented, so the user surplus derived from equation (4) is

$$US_{\alpha} = 0. \quad (15)$$

Outsourcing the system development results in a user surplus of

$$US_{\alpha \neq 1} = (v_H - v_L)xy + \alpha(v_L - c_L)y - f(x), \quad (16)$$

and a developer surplus given by

$$DS_{\alpha \neq 1} = (1 - \alpha)(v_L - c_L)y - g(y). \quad (17)$$

Based on equations (6) and (7)  $DS_{\alpha \neq 1} > 0$  and  $US_{\alpha \neq 1} > 0$ . Therefore, *outsourcing dominates internal development*.

### 3.4.5. Case IV

Here an enhancement will only be implemented when its cost is low and its value is high ( $c_j = c_L$  and  $v_i = v_H$ ). One can verify that in this case, neither the development group nor the user group has an incentive to lie about its actual parameter value. Since both the user and the development groups tell the truth, this case reduces to the case where  $v_i$  and  $c_j$  are common knowledge (Section 3.3.4). *Outsourcing dominates in this case*, because  $\alpha = 1$  implies that the development group will make no specific investment ( $y = 0 \Rightarrow c_j = c_H$ ).

### 3.4.6. Case VI

Case VI is obvious, since no enhancement will ever be implemented (see fig. 2).

## 4. Conclusions

Outsourcing is the subcontracting of some or all of the information systems functions by one firm to another. This activity has recently received increasing attention in IS circles, with a promise of a new solution to many of the problems associated with creating and operating an information system. To date, however, there have been few, if any, theoretical studies on the relative benefits of outsourcing over internalizing IS functions. This paper centers around the IS development effort, because software development projects require a substantial amount of resources and at times become a major source of disputes between the user and the developer communities. Any information system development project implies a series of exchanges between the development group and the users who pay for the development effort. These exchanges

must be properly governed to minimize the associated agency costs. Two commonly used management strategies are internal development and outsourcing. In this paper, we modeled and examined the effects of different profit sharing rules on the decision of whether to outsource or to use an internal development team. In this context, we further investigated the impact of common knowledge versus information asymmetry regarding the project development cost and its business value. Our results indicate that:

- The net benefits to the users from outsourcing are neither uniformly negative nor positive. In some cases outsourcing dominates internal development. In other cases internal development dominates outsourcing. This explains the observed coexistence of both internal development teams and of various outsourcing services.

- Outsourcing can provide incentives for the developers to sustain specific investments and thus promote future cost reductions. Even if the IS implementation surplus is shared between the outsourcing contractor and the user group, outsourcing can result in a higher value for the user group compared to using a salaried internal development team.

- Outsourcing is always preferred when the true value and cost of an enhancement are common knowledge and the development group's specific investment is required for an enhancement's net value to be positive (i.e., when  $c_H > v_H$ ). This happens because the salaried, internal development team has less incentive to make specific investments in cost reduction. In other situations, either outsourcing or internal development may be optimal depending on the parameter values.

- In the case of information asymmetry, some positive net-valued IS enhancements may not be implemented under *outsourcing*. This happens because the user and development groups try to confiscate some of the surplus; the developers by stating inflated development costs and the users by understating the enhancement's value.

- In the case of information asymmetry, internal development is always preferred when the developer has the incentive to conceal the actual development cost, thus confiscating some of the gains. In other situations, either outsourcing or internal development may be optimal, depending

in part on the internal development group's compensation scheme.

- We empirically identified several cases where changes in the portion of the surplus diverted to the user,  $\alpha$ , results in a threshold-type functional form: the user's surplus *increases* with  $\alpha$  below that point and *decreases* with  $\alpha$  beyond that point. This happens because diminishing incentives for the developers result in an immediate decline in their willingness to make specific investments. This implies that attempting to capture all the gains ( $\alpha$  close to 1) may be detrimental to the user group.

- Finally, we found that outsourcing can be optimal (for the user and for the developer) when the user surplus does not increase monotonically with  $\alpha$ , and when the expected developer surplus is nonnegative. However, it is conceivable that the users and developers will not be able to negotiate a sharing rule acceptable to both parties. This happens when the  $\alpha$  maximizing the users' gains results in minimal developer gains and vice versa.

Overall, our main objective in this paper was to present an economic assessment of IS outsourcing policies in the context of software development. The process of software development is sequential in nature, and at each phase there is uncertainty about the expected development costs and the benefits of future development phases. As a result, there is room for specific investments that will lower the expected cost and increase the expected value of future phases. The core of our results lies in recognizing the fact that an external development group has strong incentives for making these specific investments, since these investments increase the likelihood that the next development phase will be implemented, and that this group will be retained to implement it. This is in contrast to a salaried, internal development team that has no incentive to make special efforts to lower the cost or increase the value of the succeeding development phases (they control a captive market). Hence, the value gained by the users when outsourcing the IS development task to an external group comes from the specific investments made by the external development group.

Moreover, our findings indicate that outsourcing does not always dominate internal develop-

ment. In fact, internal development dominates outsourcing when the specific investment made by the user group has a large impact on the value of the information system developed and when in addition, the specific investment made by the development group has a relatively smaller impact on the development costs. In these cases, the user group will be better off resorting to internal development, since it captures all the value of its specific investment.

The results presented in this paper are derived from a particular model; however, the existence of these general phenomena is not restricted to the particular case analyzed. Developing mathematical models, as we have in this paper, is only the first step in understanding how contracts affect the IS organization. To extract the full benefits from examining information systems management issues via contract theory, one has to determine the characteristics unique to each function in the information systems organization and develop models that highlight those unique characteristics, as was done in this paper for the software development case. A richer model would integrate the effects of contracting costs with market imperfection and technological economies. Several other concerns that deserve special attention in our context are left open for future research. These concerns include: Moral hazards, low balling, the value of incumbency, and the potential pitfalls to outsourcing due to bilateral trading conditions and their resulting potential for opportunistic behavior [21].

#### Acknowledgements

This paper has benefited from useful comments made by Professors Michael Barclay, Eugene Handel, Stacy Hole, Kevin Murphy, Clifford Smith, Karen Van Nuys, Michael Weisbach and Jerold Zimmerman at the William E. Simon Graduate School of Business Administration at the University of Rochester. We also would like to thank the editors, Professors Chris Kemerer at the MIT Sloan School of Management, J. Yannis Bakos at the Graduate School of Management at the University of California, Irvine, and the three anonymous referees for their extensive reviews. Their comments, insights and encouragement were essential for the completion of this paper.

The authors wish to thank Katherine M. Hudson, Vice-President and Director of Corporate Information Systems at Eastman Kodak Company; Larry Ford, Vice-President Information Systems, IBM Corporation; Frank Palm, Project Manager, IBM-Kodak; Elmer Beke, Assistant Project Director, IBM-Kodak; and Patrick O'Neill, Manager, Andersen Consulting, Co., for useful discussions. Finally, we wish to acknowledge Ron Hein for his careful reading, which has significantly improved the presentation style of this paper. Any errors are the sole responsibility of the authors.

This research was in part supported by the William E. Simon Graduate School of Business Administration, University of Rochester.

#### References

- [1] N. Ahituv and S. Neumann, *Principles of Information Systems for Management*, 3rd ed. (Wm. C. Brown Publishers, Dubuque, IA, 1990).
- [2] R.D. Banker and C.F. Kemerer, Scale Economies in New Software Development, *IEEE Transactions on Software Engineering*, 15, No. 10, 1199–1205 (1989).
- [3] R.H. Coase, The Nature of the Firm, *Economic*, 4, 386–405 (1937). Reprinted in: G.J. Stigler and H.E. Boulding, Eds., *Readings in Price Theory* (Richard D. Irwin, Homewood, IL).
- [4] E.F. Fama and M.C. Jensen, Separation of Ownership and Control, *Journal of Law and Economics*, 26, 301–325 (1983).
- [5] P.O. Flaatten, D.J. McCubbrey, P.D. O'Riordan and K. Burgess, *Foundations of Business Systems* (The Dryden Press, Chicago, IL, 1989).
- [6] G.I. Green, Perceived Importance of Systems Analysts' Job Skills, Roles and Non-salary Incentives, *MIS Quarterly*, 13, No. 2, 115–134 (1989).
- [7] S.J. Grossman and O.D. Hart, The Costs and Benefits of Ownership: A Theory of Vertical and Lateral Integration, *Journal of Political Economy* 94, No. 41 (1986).
- [8] V. Gurbaxani and C.F. Kemerer, An Agent-Theoretic Perspective on the Management of Information Systems, *Proceedings of the Twenty Second Annual Hawaii International Conference on System Sciences*, III, 141–150 (1989).
- [9] O. Hart and J. Moore, Property Rights and the Nature of the Firm, Discussion Paper No. TE/88/174 (1988). *Theoretical Economics Workshop, Suntory-Toyota International Centre for Economics and Related Disciplines, London School of Economics*, 10 Portugal Street, London WC2A 2HD.
- [10] B.R. Holmstrom and J. Tirole, The Theory of the Firm, in: R. Schmalensee and R. Willig, Eds., *Handbook of Industrial Organization*, Volume 1 (North-Holland, Amsterdam, 1989).