Journal of International Information Management

Volume 1 | Issue 1 Article 7

1992

Vertical integration of computing facilities, a transaction cost approach

Michael Piptea
Iris Systems

Dan Pieptea
Illinois State University

Vertical integration of computing facilities, a transaction cost approach

Michael Pieptea
Iris Systems

Dan Pieptea
Illinois State University

ABSTRACT

This paper develops a generalized model for computing capacity by identifying the main activities in a computer facility as formal transactions. The formal transactions carried out in a computing environment are characterized along two dimensions: freqency of occurrence and idiosyncrasy of assets. We make recommendations as to how these formal transactons should be executed - across markets, internally or using intermediate forms of governance. It is shown that transactions in a computing facility fall into one of three categories: procurement of assets, maintenance and modification of assets, and collection, storage and processing of data into information. Formal transactions in a computing facility such as procurement of application software, recruiting for different positions, data entry and transformation of data into information are discussed in detail. The efficiency of possible forms of governance is analyzed for each formal transaction. Generalized efficiency criteria for the market approach versus managerial control are provided. The conclusions find an application in defining the boundaries of end-user computing as well as other computing facilities such as data centers.

INTRODUCTION

One of the greatest challenges of business-oriented computing facilities is to increase personal and organizational productivity while responding to the ever-changing business conditions. Most traditional computing facilities face difficulties that prevent them from providing a fast service to the end user. These difficulties are seldom solved by placing computer resources under the direct management of the end user.

In this paper we argue that one of the major causes for the traditional inefficiency of computer centers is vertical over-integration. We evaluate the degree to which a computing facility should be vertically integrated. Activities in a data center such as procurement of application software, recruiting, data entry, data processing and others, are viewed as generalized transactions in the sense of the transaction cost economic theory. We identify and characterize the formal transactions carried out in a computing environment along two dimensions, frequency of occurrence and idiosyncrasy of the investment under uncertainty. We recommend ways in which these formal transactions can be efficiently executed - across markets, internally, or using intermediate forms of governance. Our conclusions have a direct application in defining the boundaries of end-user computing. We feel that greater flexibility of computing power can be achieved by employing economic principles of the transaction cost theory in the process of organizing a computing facility.

THE FRAMEWORK

Transaction cost economics is defined by Oliver Williamson (1979) as an "interdisciplinary undertaking that joins economics with aspects of organization theory and overlaps extensively with contract law." As Ronald Coase (1937) shows, markets and firms are alternative means of completing a related set of transactions. Contracting out (the market approach) and managerial authority represent two competing alternatives for organizing transactions. In this setting, the firm presents itself as a set of contractual relationships. The ideal boundaries of the firm are such that total transaction costs are minimized. We argue that the concept can be extended to the departmental level, in this case a computing facility. Under competition, the choice between markets and hierarchies is driven by efficiency, given that uncertainty of outcome and opportunistic behavior increase the cost of using markets. Oliver Williamson (1975) shows that bounded rationality (i.e., the inability to define all future states in a transactional relation) and the tendency to opportunistic behavior (due to the small number of parties involved in an exchange) make costs of enforcing contracts eventually prohibitively high and thus market governance to fail and be replaced by hierarchies. Later, Ian Macneil (1978) and William Ouchi (1980) describe additional forms of governance and insist on the existence of intermediate modes of organizing exchanges.

The most fruitful level of analysis of the vertical integration issue, is not the organization, but the transaction (Barney and Ouchi, 1986). In order to establish the boundaries of an organization, each transaction within the organization must be analyzed. Depending upon the characteristics of the transactions, one can decide upon the most efficient way to carry out the pertinent activities.

CHARACTERISTICS OF TRANSACTIONS

Three dimensions characterize a transaction: frequency of occurrence, the degree to which an asset employed in the transaction (or the investment) is idiosyncratic and uncertainty of outcome. Frequency of occurrence may be classified as occasional or recurring. Assets involved in a transaction can be firm-specific, mixed and non firm-specific. In this paper we use the terms firm-specific asset, idiosyncratic investment, and idiosyncratic asset interchangeably. Idiosyncratic investments are in essence non-marketable investments. They occur in conjunction with non-transferable, specialized designs. The third dimension, uncertainty, is always present to a certain degree.

FORMS OF GOVERNANCE

Market governance uses as instrument the classical contract, where identity of parties involved in the transaction is considered meaningless, the nature of the agreement is precisely delimited and consequences of nonperformance fairly predictable from the very beginning (Macneil, 1978). Trilateral governance, also known as neo-classical contracting, brings in third party assistance to resolve possible disputes. Neo-classical contract law stems from the admittance that given the conditions of bounded rationality and opportunistic behavior, there is a need for both parties to trust the equity mechanism in order to make the transaction work. Bilateral governance steps in when two parties develop a long-term relationship maintained by transaction specific economies. In that case, the relationship may become increasingly of administrative type. Bilateral governance is fueled by hazardous reliance on market conditions and the possibility of recovering together specialized costs. Unified governance of the exchange happens when the transaction is organized internally, as is the case of in-house application development. Vertical integration replaces the market mechanism with the administrative authority.

GENERALIZED MODEL FOR COMPUTING FACILITIES

Business computing facilities, ranging from computer centers to workstations, are in the business of providing the end-user with information by executing a series of formal transactions and employing various assets. In developing a generalized model for computing capacity, we construct on the work of Oliver Williamson (1979) who matches governance structures to transactions. His system of classify transactions is described in Fig. 1.

Figure 1. The Williamson Framework: Matching Formal Transactions with Governance Structures

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="3">INVESTMENT CHARACTERISTICS</td></tr>
<tr><td>Non-Specific</td><td>Mixed</td><td>Idiosyncratic</td></tr>
<tr><td rowspan="2">FREQUENCY</td><td>OCCASIONAL</td><td>Market Governance (Classical Contract)</td><td colspan="2">Trilateral Governance (Neo-Classical Contract)</td></tr>
<tr><td>RECURRENT</td><td>Market Governance (Classical Contract)</td><td colspan="2">Relational Contracting
Bilateral Governance | Unified Governance</td></tr>
</table>

For transactions that involve non-specific assets, frequency of recurrence does not matter, and market governance (classical contract) is recommended. The closer to an idiosyncratic investment an occasional transaction requires, the more appropriate a trilateral governance. Bilateral forms of governance are recommended for recurrent transactions that require customized assets and unified governance (pure vertical integration) is appropriate for recurrent transactions utilizing highly idiosyncratic assets.

The purer forms of governance, i.e., market and the unified structure are less sensitive to uncertainty. Under pressure of uncertainty, bilateral governance tends to become a unified structure. Some trilateral relationships evolve into market contracts by simplification of transactions.

In our generalized model for computing capacity, we identify and analyze the assets first and the transactions second. In a computing environment we distinguish four categories of assets: data, hardware and system software, application software and human resources. The data is the most perishable asset of all. Data can be firm non-specific (e.g., stock quotes) but most often is a firm-specific asset. Hardware and system software, in most cases, are not idiosyncratic investments. Sometimes, specialized equipment may be unique to the firm, like certain data acquisition devices, but in general business computing hardware is a non-specific asset to the firm.

A computing facility runs a portfolio of application software that has a fairly wide range of firm specificity. Some of the applications are firm-specific to a great extent, other are industry specific, while some others have a low degree of firm specificity. Among the latter are financial systems. In general, we see firm specificity increase when going from Transaction Processing Systems (TPS) to Structured Decision Systems (SDS) and further to Decision Support Systems (DSS) (Pieptea and Anderson, 1987).

Human resources in a computing environment cover a range of firm specificity, in many cases related to the other assets that they are involved with. Listed in decreasing order of firm specificity, human resources in the data center are the manager, the information analyst, the system designer and the programmer. However, exceptions from this order exist: a maintenance programmer assigned on a firm-specific peice of application software is closer to the firm than a development programmer. We shall elaborate later on this idea.

Transactions executed in a computing facility fall into one of the three major categories: (1) procurement of assets, (2) maintenance and modification of assets, and (3) collection, storage and processing of data into information.

Procurement of hardware and system software is usually a non-recurring transaction involving non firm-specific assets. One may think of a very specialized piece of equipment to be developed in house, but generally the business community goes to the market for executing this type of transactions and thus, reality validates the model.

Procurement of application software is typically a set of non-recurring transactions involving resources of a wide range of firm specificity. Thus, according to the model, it makes more sense to buy (or lease) a piece of financial software (low firm specificity) and have it installed by a consulting firm rather than develop the software in house. On the other hand, internal resources are better utilized in a highly idiosyncratic development effort. Here the model can serve as a useful guide. Too many times software procurement is synonymous to software development. At the other extreme, consulting firms are hired for what in essence is an open-ended project. In many cases, by reasonably simplifying requirements, one can find market solutions to what looks like an internal development. Customization of packaged software can be successfully accomplished using a long-term relationship (trilateral or bilateral form of governance). Firm-specific application software should be developed internally (unified governance).

Procurement of human resources is, in general, an occasional transaction, unless there is a severe turnover problem. Programming and system design staff is less firm specific than information analysts and management personnel. We emphasize the programming function since it presents the most diverse features. We distinguish the development programmer from the maintenance programmer as two different positions. Because the line between the analysis and programming in maintenance is quite fuzzy (most of the time assignments are a combination of business system analysis and coding), the maintenance programmer is in general more firm-specific than the development programmer.

There are a number of alternative paths for procurement of human resources. The market alternative is provided by personnel agencies (headhunters) and use of contractors while the in-house path is represented by the use of the internal personnel department, personnel search conducted at the MIS department level or providing data processing training to potential internal candidates. For the procurement of a maintenance programmer we recommend a long-time relationship with a placement agency. The development programmer on the other hand, with stronger ties to the technology, is almost non firm-specific. This statement is validated by the relatively high turnover rate of the more technical programming staff. For development projects, with well defined specifications and timetable, contract programmers offer a more efficient alternative. The information system designer is technology specific though less than the development programmer. The information analyst and the manager are firm-specific assets and their development should be internally organized.

Maintenance and modification of hardware and system software should be and usually is executed over the market (the maintenance contract). Note that extremely short time allowed to fix potentially recurring problems (system crashes, certain on-line or real time system failures) increase the idiosyncrasy to the point that it justifies the presence of a permanently employed technical expert ("guru") on the premises. In many cases this person is known as system programmer.

Maintenance and modification of non firm-specific applications is recommended to be carried across markets or bilateral forms of governance. For example, the installation and light customization of a new release of a market General Ledger product should be done with consultants. Too much customization of a packaged software product may create important problems. Once in production, the degree of firm specificity of an application changes. Because of the urgency of production-related problems, application software maintenance is fit to be organized internally. Highly specialized applications should be definitely maintained in house. The model suggests that training for information management and analyst should be organized internally, while training should be contracted out.

Data collection is most often a recurring process. It usually employs non-specific assets, which is why many companies contract out their needs to external keying services. If, on the other hand, very specialized resources are needed, the data collection transaction is at home within the boundaries of the organization. Since the data itself is a highly firm-specific asset, its storage is a transaction that should be executed internally, rendering the maintenance of the database an important departmental function.

Data processing, driven by the recurrence argument and the firm specificity of the data, is to be carried out mostly in house. Occasional processing like disaster recovery planning should be done and is eventually done over the market. On an application basis, those applications that are non firm-specific could be eventually farmed out to service bureaus. Capturing the information on paper or screens shoudl be done internally and under proper security measures.

CONCLUSION

This paper approaches the different activities that take place in a computing environment from the transaction cost economics point of view. We identify and characterize the formal transactions in a computing organization along two dimensions, frequency of occurrence and idiosyncrasy of the assets employed in the transaction, and study the impact of uncertainty. We make recommendations as to how these formal transactions should be executed - across markets, internally or using intermediate forms of governance. The following summarizes our conclusions:

The end-user should be very selective in choosing what formal transactions related to computing he/she wants to bring under closer control. End-user computing stands a good chance to be plagued by inefficiently run activities, if it assumes the responsibility of carrying out high cost transactions. Only firm-specific and/or highly firm-specific applications belong under close control of the end-user. One way to improve the service offered by traditional computing facilities is to re-think operations in light of the transaction cost economics theory. Cost efficiency can be considerably improved if these principles are employed in the day-to-day practice of data center management.

REFERENCES

Ahituv, N. & Neuman, S. (1982). *Principles of Information Systems for Management*. Dubuque, Iowa: Wm. C. Brown Company.

Barney, J. & Ouchi, W. (1986). *Organizational Economics*. San Francisco, California: Jossey-Bass Inc.

Coase, R. (1937). The Nature of the Firm. *Economics, 4*, 386-405.

Ouchi, W. (1980). Markets, Bureaucracies and Clans. *Administrative Science Quarterly, 25*, 124-141.

Macneil, I. (1978). Contracts: Adjustment of Long-Term Economic Relations under Classical, Neo-Classical and Contract Law. *Northwestern University Law Reviews, 72*, 854-906.

Peiptea, D. & Anderson, E. (1987). Price and Value of Decision Support Systems. *MIS Quarterly, 12*, 515-526.

Williamson, O. (1975). *Markets and Hierarchies: Analysis and Antitrust Implications*, New York: Free Press.

Williamson, O. (1979). Transaction Cost Economics: The Governance of Contractual Relations. *Journal of Law and Economics, 22*, 233-261.