ICIS 1996 Proceedings

International Conference on Information Systems (ICIS)

December 1996

Software Contracting: A Process Model Approach

Ari Heiskanen
University of Helsinki

Michael Newman
University of Manchester and Vrije Universiteit, Amsterdam

Jouni Similae
University of Oulu and CCC Software Professionals

SOFTWARE CONTRACTING: A PROCESS MODEL APPROACH

Ari Heiskanen
University of Helsinki

Michael Newman
University of Manchester and
Vrije Universiteit, Amsterdam

Jouni Similä
University of Oulu and
CCC Software Professionals

Abstract

Experiences of making software development and purchase contracts between a user organization and several software vendors are analyzed over a decade-long period. Three information systems histories are presented as case examples. The analysis is based on observations of the contract negotiations. Transaction cost framework is used to explain the forms of the joint organizational relations between the parties. The analysis yields a descriptive model that depicts in a concise way how the relationships have evolved. The model identifies encounters between the vendors and the client which may change the contractual state between the parties. The passages between consecutive encounters are called episodes. By perceiving contractual systems development as a series of encounters and episodes, it is possible to identify the critical turning points of development work and to display the dynamics of contracting. The paper ends with a discussion of the findings and some implications for researchers and practitioners.

1. INTRODUCTION

Our purpose is to describe and explain the complex negotiation processes of the software procurement history of several major administrative information systems (IS) of the University of Helsinki (HU). To do this, we use an approach from the process research tradition (Gersick 1991; Newman and Robey 1992; Robey and Newman 1996). The time span of our investigation covers a period from the mid-1980s up to the present. We analyze the negotiation processes from the inside, as seen by the participants. While there exist good treatments of IS contracting (e.g., Whang 1992; Richmond, Seidmann and Whinston 1992; Fitzgerald and Willcocks 1994), we hope that our case gives fruitful empirical insights by showing concisely how the dynamic contractual relationships between software developers (vendors) and users (clients) have been managed. Furthermore, we believe that our way to model the dynamics of software procurement is easily applied in other circumstances.

The main point of view of this paper is that of the first author in his role as the Chief Information Systems Officer of the University. The organizational records, gathered form the mid-1980s up to now, as well as personal observations and notes, constitute the research data. We will follow Mintzberg's (1979) recommendation of a direct and descriptive research strategy, relying on simple methods, tracing the flow of decisions in the University. Additional information has been obtained from the managers of the vendors concerned. They have agreed to participate in this intellectual exercise of self-reflection. Even deeper involvement from the vendors' side is represented by the third author. He has also acted in the practical role of a Strategic Business Area manager of one of the vendors concerned during the years 1985 to 1987, and continuing until now in other managerial tasks with the same vendor.

Our approach is a further development and enlargement of the Newman and Robey model of user-developer interaction. In their process model, they identified three main elements: (1) the antecedent conditions, (2) the possible interaction states between the users and developers (acceptance, equivocation, rejection), and (3) the development trajectory of the interaction process. The interaction process consisted of "equilibrium" state progress passages, called episodes, and critical events between the episodes, labeled encounters. Encounters change the nature of the interaction. An example of an encounter can be the hand-over of a test system to the users. This may change the state of the interaction from acceptance to equivocation or even rejection when the users begin to suspect that the proposed system does not fulfill their needs.

Our plan to produce the model of software contracting follows. We first present three actual IS procurement histories. Next we analyse how the contracting parties have been able to control each other. After that we begin the model building through methodical sense making of the situations (for the details of the research approach, see Heiskanen 1995). As Schön (1983) puts it, practitioners do not consider that they have formed a satisfactory account of phenomena in any practice situation until they have framed it in terms of their overarching theory. An overarching theory does not give a rule that can be applied to predict or control a particular event, but it supplies language from which to construct particular descriptions and themes from which to develop particular interpretations (Schön 1983, pp. 273-274). For our case this means that we identify the possible archetypal relationship states between the contracting parties (developers and users) using the notions of transaction cost theory, and we present our main result by outlining a way of modeling the dynamics of the software contracting process.

2. IS CONTRACTING IN OUR CASE

There are five software vendors in the histories below. Three of them have agreed to participate in our research and allowed the use of their real names: KT-Tietokeskus (KTT), the State Computing Centre (SCC), and CCC Software Professionals. The other vendors are denoted by pseudonyms AAA and BBB. These latter two only had a minor role in 1987 and 1988; therefore, we did not offer them an opportunity to participate. The client side has been represented by the administrative EDP Office of the University, headed by the first author throughout the study period.

We have chosen three IS procurement histories to be discussed. The purpose of the choice is to illustrate how a large variety of development trajectories can easily be captured into a graphical representation. All three systems are successful in the sense that they are in production use, they do not pose any serious problems to IS management, and the users appear to be at least moderately satisfied with them, according to two user information satisfaction surveys.

2.1 The Student Information System of the University of Helsinki, SISUH

SISUH consists of two main pieces of software, CERS for the centralised part of the system, running on VAX/VMS, and an additional PC/DOS system, DERS, to be used in the departments. The first portions of the software of CERS were delivered through direct purchase from CCC, leading to use in the summer of 1987. The vendor choice was based on an informal scanning. The reasons to choose CCC were its personnel's good record of achievements and their strong academic background.

The University strove also for arrangements for competition between software houses. One of these competitions was about DERS. The specifications of DERS were sketched in the spring of 1987 and a bidding competition for DERS was started in the summer of 1987. The main competitors were software houses AAA, CCC, and SCC. The first tenders of all these bidders had about the same price, but the planned functionality of the system delivered by SCC was inferior as compared to that of AAA and CCC, so the competition was between AAA and CCC. During negotiations, the prices of the two tenders became nearly equal. The contract was won by CCC, mainly because of its acquaintance with the University as well as with CERS, a system closely related to DERS. The use of DERS began in the autumn of 1988.

The cooperation between the University and CCC in CERS/DERS development evolved over the years, typically from fixed-price deliveries of pieces of software into contract programmer hiring. It appeared that the change-over of the maintenance and further development of CERS/DERS from CCC to the EDP Office would be the most economic option, because the personnel of the EDP Office, acting as gatekeepers (Heiskanen and Similä 1992) between users and outside developers, considered it easier to develop the software by themselves. Therefore, the University decided in 1990 to develop the software without CCC's assistance.

2.2 The Personnel and Job System, PERJOB

The history of PERJOB development began with competitive bidding in November 1986. The EDP Office was in charge when the specifications were prepared prior to the bidding competition, and it represented the users in the bidding software houses AAA, BBB, CCC and SCC. The contract for PERJOB was won by AAA, because it eventually promised to deliver the specified software free of charge. The motives for this kind of behavior are analyzed in section 3. The first version of the system was delivered successfully in the beginning of 1988.

AAA was not able to deliver a reporting package for PERJOB in 1988. It seemed to the client that this was caused by a personnel shortage. Consequently, the EDP Office made a decision to stop the co-operation with AAA and continue the development of PERJOB with another software house, CCC. The University had used CCC's services in developing the student record system with acceptable performance. This was the reason to choose CCC, and so to avoid the work that would have been incurred in the search of a totally new software house.

A problem with PERJOB has been that it is not integrated with the payroll system. In the spring of 1994, the University decided to purchase a new payroll package from SCC. At the same time, HU and SCC negotiated a set of contracts concerning the development of a personnel system and general co-operation. HU would act as a pilot client receiving the new system, called PERSONNEL, with a reduced price, as a reward of its development effort. These contracts implied a strategic level alliance in developing the personnel system. The roots of this alliance were in the consulting work from 1989 to 1991 by SCC to develop the payroll office functions of HU.

2.3 Accounting

The purchase of the accounting package in 1991 presents an extremely straightforward case: it was a direct purchase without bidding competition from a single vendor, Tietovoima, which had earlier been chosen by the State to develop an accounting package for all state bureaus. In the beginning of 1992, KTT purchased the accounting systems business from the owner of Tietovoima. This deal entailed the change-over of the accounting system development and support personnel from Tietovoima to KTT. The contract made between HU and Tietovoima remained valid with KTT without renegotiations. During 1991, the choice of software vendor of accounting packages for the state bureaus was extremely simple, because there was only one state-approved option. Later the State abandoned its guiding role and relaxed the statutes allowing other vendors to bid as well.

3. PERSPECTIVES OF IS CONTRACTING

IS development can be classified as three major approaches: in-house development, contractual customized development with outside vendors, and software product purchase (Grudin 1991; Saarinen and Vepsäläinen 1994). In this section, we outline in general terms our experience of how the two different organizations, the client and the vendors, have been able to exercise control over each other in contract negotiations that punctuate (or sometimes overlap) the IS development work. It is well known that IS outsourcing is dynamic (for example, Fitzgerald and Willcocks 1994). Moreover, it is not possible to specify every contingency into a closed contract over a long period of time (Richmond, Seidmann and Whinston 1992; McFarlan and Nolan 1995, p. 17). The negotiations fill the void that emerges out of the incomplete contracting issue. The control mechanisms available to the parties form the basis for contracts.

There are three kinds of "bad" costs involved in the software purchase process. First, there is the cost or risk of getting a wrong or poorly designed system. Second, there is the cost of paying too much for the right system. The third kind is more indirect: the client must also generally avoid paying too little for the system (Page, Williams and Boyd 1993; Fitzgerald and Willcocks 1994). The client must realize that the vendor has to make a profit for itself in order to continue serving the client in future projects. The potential "bad" cost in this case will be realized through the disillusionment of the vendor with the present project, materializing perhaps in personnel reassignments, delays in project schedules or even total project failures.

3.1 Client's Perspective

From the client's point of view, we see three different mechanisms that may be of help in retaining sufficient control. The first control mechanism, suitable in small projects, is the introduction of client gatekeepers (Heiskanen and Similä 1992) who have the possibility to take over the work, as happened in our case with the CERS/DERS development.

The second control mechanism emphasizes the long-range relationship between the software house and the client. This is because there is a trade-off between the emphasis on competition leading to reduced price of the work and a need for creating long-standing and mutually beneficial relationships between users and developers. Sometimes this leads to a strategic alliance. This was the case when HU and SCC developed the personnel system. The rationale for the client was to get a high class software package with a reduced price, adapted to the circumstances of the client. The vendor was supposed to collect its revenues from other clients.

The third control mechanism may come out of the certified quality management and assurance system. For example, the ISO 9000 certificate is conditional and it is awarded for a limited period. If the client finds the vendor's performance unacceptable, a powerful means for improvement is a complaint to the standardization organization that awards (and withdraws) the certificates.

3.2 Vendor's Perspective

From the vendor's point of view, there are several possible control mechanisms. The decisive factor is the strategic or tactical importance of the client or project. The client may be important because it represents a desirable business sector. The project may be desirable because of the particular development environment.

The first control mechanism is the price level acceptable in the bidding process. The vendor must strive toward reasonable compensation in order to survive economically. The compensation may, however, be viewed in a short-term or long-term perspective.

The second control mechanism, an aspiration toward a long-range relationship with the client, may at times conflict with the first control mechanism in the short-term as the following example illustrates. The contractual price level is a very powerful control mechanism. Especially in public bidding, the vendor can in effect force the client's hand by lowering the price to a level where other bidders are not willing to enter. This happened in the case of PERJOB where contractor AAA, in its desire to expand into the public administration sector and to get a reference project of a particular tool environment, effectively forced the other bidders out of the game by its pricing policy. The use of the price level control mechanism in the described manner is very dangerous if the expected business opportunities fail to materialize. This evidently happened in the case of AAA, leading to disillusionment with the project, reassignment of project personnel, and finally to project failure.

The price level control mechanism can be also used in a reverse manner in order to break involvement with a client at least temporarily. A further negative effect of the use of the price level control mechanism, at least in an exaggerated manner, is its effect on future bidding. The vendor may profile itself as too expensive to be included in future bidding or too cheap to bid profitably.

The second main control mechanism, aspiration toward a long-range relationship with the client, is in general more preferable from both the vendor's and the client's viewpoints. This aspiration may be realized in other ways than by contractual price level control. For example, a vendor may by assigning highly qualified persons in the early projects to make, in effect, a lasting impression and a strategic investment for getting future projects from the client.

A third control mechanism, and perhaps the most important, consists of a set of measures for generally improving the software production maturity of the vendor. The most important element here is the development of a preferably certified quality assurance system and its use as a marketing mechanism (Similä, Kuvaja and Krzanik 1995). In the long term, this is the only control mechanism that has a lasting effect from the vendor's point of view.

3.3 Combining the Views of Client and Vendor

A summary of control mechanisms from the client's and the vendor's points of view is in Table 1. Several other control mechanisms have been described in IS outsourcing literature (for example, Lacity and Hirschheim 1993), but these have been the most vital ones in our experience.

An important conclusion from our experiences is that a bilateral dependency develops between the client and the vendor when they are engaged in customized software development. Both parties need the cooperation and input of each other in order to be successful. Therefore the relationship between them cannot be explained as a clear-cut market transaction, but neither do the parties form an integrated organizational entity, a hierarchy. It seems that we are here dealing with some kind of an intermediate organization type between market and hierarchy. In order to understand the special features of this hybrid arrangement, we take a brief look on the relevant literature.

Table 1. Summary of Control Mechanisms

CLIENT	VENDOR
Introduction of client gatekeeper	Control of contractual price level
Aspiration toward long-range relationship with vendor	Aspiration toward long-range relationship with client
Use of quality system certification body for pressuring vendor	Software production improvement measures
Bidding vendors against each other	Other control mechanisms
Other control mechanisms	

4. WORK GOVERNANCE MODES IN IS DEVELOPMENT: A STRUCTURAL ANALYSIS

The bilateral dependency of a software vendor and its client can be understood through several theoretical frames. Some of them are mentioned here in order to illustrate the variety of approaches. Gurbaxani and Whang (1991) discuss transaction cost theory and agency theory. McWilliams and Gray (1995) add environmental uncertainty (an organization theory construct) and resource-based theory (a strategic management construct). The contracting dilemma is analyzed, for example, by Richmond, Seidmann and Whinston and by Whang. Bakos and Brynjolfsson (1993), using the economic theory of incomplete contracts, indicate that a buyer will often maximize profits by limiting its options and reducing its own bargaining power. Asanuma (1989) describes the Japanese style of manufacturing where the client can exercise managerial control upon the vendors by using at least two vendors for each service purchased outside, the vendors being classified according to the desirability to continue business with them. Powell (1990) argues that relational or network forms of organization are often a viable way of economic exchange, instead of tight vertical integration or loose market.

We have chosen transaction cost theory as the basis for our analysis, because we felt that this theory is well known, conceptually parsimonious, informative, and easily applicable in our case. Transaction cost theory is targeted to the analysis of make-or-buy decisions. The essential question is whether the (client) organization should make a product or service internally, using a hierarchy to control actor co-ordination, or purchase the product or service outside, using the mechanisms of markets (Williamson 1991; Lacity and Hirschheim 1993). The problem is to decide which governance mode is the most economical one in a given situation.

The transaction cost theory divides costs into two categories: production costs and transaction costs. The latter consists of the costs of monitoring, controlling and managing transactions. To (over)simplify, transaction cost theory says that if a purchase is simple, the market mechanism should dominate, because it is more efficient in production. Therefore, simple products or services should be bought, not produced internally. When the purchase complexity grows, the internal production becomes more viable, because the transaction costs of a complex purchase may become too great. The reason for this price growth may, for example, be the opportunistic behavior of the vendor, or the specialized resources that have to be developed for this special occasion. There is also a hybrid form that is a joint organization of the client and the vendor (Williamson 1991). These three modes (market, hybrid, hierarchy) are the archetypal arrangements that can prevail between the developer (vendor) and the user (client). A fourth organizational archetype often mentioned in this context, the clan (Ouchi 1979), seems inappropriate, because it is unreasonable to rely on the existence of a shared value and belief system between the contracting parties.

Williamson (1991, p. 277) argues that there are three key dimensions along which the three governance modes can be differentiated. The first dimension concerns the contract law that is applied when resolving disputes between developers and users. The second dimension is about the adaptability of the user-developer relationship when (unanticipated) changes occur. The third dimension is the possible use of incentive and control instruments.

Williamson's argument is that market and hierarchy are the polar extremes of organizing the work, and the hybrid form is an intermediate arrangement according to all of the above dimensions. In the ideal market, the identity of the client and vendor are irrelevant: the products are standardized in such a way that the price of the product gives enough information for decisions. Prices follow the changes in supply and demand; this is the dominant way of adaptation, and no negotiation is needed. The contract is a straightforward deal and fundamental disputes are cleared through litigation. The incentive is the price paid by the client. It is very consequential to the vendor, because if there is no deal, there will be no payment.

In a hierarchy, the coordination of actors follows the command line of the organization. The disputes are solved through administrative fiat. Adaptation presupposes (intraorganizational) negotiations. Williamson's argument is that the incentives are usually much lower in powerfulness in the hierarchy than in the market. They consist of means such as using accounting information to reveal the profitability of projects or using career rewards and penalties.

5. HOW TO MODEL THE DYNAMICS OF THE SOFTWARE CONTRACTING PROCESS

In this section, we present our tentative method for modeling the dynamic software contracting process. The idea is to apply the classification market-hybrid-hierarchy to the IS development trajectories. This we will do by presenting the trajectories as lines punctuated by negotiation encounters that may change the state of the process from one class to another. The passages between the negotiation encounters, the episodes, mean development work that does not change significantly its cooperative character between the client and the vendor.

As the first part of our model, we have tabulated the development encounters in Table 2 for SISUH, in Table 3 for PERJOB, and in Table 4 for the accounting system. In order to save space, episodes are not included in this tabulation. Each of the encounters is consecutively numbered, dated (year/month), and described. The encounters as well as the ensuing episodes were identified and classified by the first author. This work was based on his analysis of the archived process data, consisting of some thirty folders of organizational documents. The data were familiar to him, because he had participated in all of the client's decisions concerning the relationships with IS vendors. The identity of the encounters and episodes was confirmed by the third author on CCC's part. The managers of the vendors did not make any objections when the episode/encounter descriptions were presented to them. The second part of our model presents the "shapes" of the process by connecting the encounters with the episodes. They are in Figures 1, 2, and 3, including the respective encounter numbers. With these two sources, we believe that the reader is able to easily grasp the essence of the dynamics of the long processes.

6. DISCUSSION AND CONCLUSIONS

The motive of writing, the meaning of this paper for the practitioner authors, is to gain more understanding of what has been the essence of their experience, to get more professionalism in vendor/client control, and to increase self-understanding. As Boland (1991) says of these kinds of self-reflective studies, intellectual curiosity toward one's own professional practice can produce insights that help to improve the quality of management information systems both in particular and general cases.

Table 2. The Encounters of SISUH Development

Enc. no	Date	Encounter description
1	1985/9	Preliminary specifications for CERS prepared.
2	1985/9	Informal contract between HU and CCC for refining the specifications.
3	1985/11	Contract between HU and CCC concerning the core part of CERS software.
4	1986/6	An experimental project for developing departmental data processing begins. The idea of DERS emerges within this project.
5	1987/6	The specifications of DERS completed within HU. Request for proposals of development mailed to vendors.
6	1987/10	CCC chosen to be the vendor.
7	1990/5	HU takes over the development (perfective maintenance) of CERS and DERS.

Table 3. The Encounters of Personnel Systems Development

Enc. no	Date	Encounter description
1	1986/11	Specifications for PERJOB prepared within HU and a request for proposals is mailed to vendors.
2	1987/2-6	AAA wins the contract, development begins but the final contract must be approved by the State officials. Finally the PERJOB contract is signed between HU and AAA.
3	1988/6	HU decides to change the software vendor because AAA is not able to deliver the reporting package.
4	1989/6	HU orders consultancy work from SCC for developing the functions of the payroll office. This leads to a long development process, including the adoption of a new data entry and reporting package for payroll.
5	1993/3	HU and SCC agree about making a feasibility study of a new personnel system.
6	1993/12 – 1994/6	SCC proposes wide cooperation with HU, including a pilot project for developing an integrated personnel-payroll system. This proposal leads to a prolonged negotiation encounter, related to the choice of the payroll system (SCC) in March 1994.
7	1994/1-3	A bidding competition is arranged by a consortium of Finnish universities in order to purchase accounting, personnel and payroll systems. The consortium is unable to make a concerted decision. In this connection, HU has informal negotiations with KTT about the possibilities of choosing KTT as the payroll/personnel services vendor.

Table 4. The Encounters of the Purchase of the Accounting Package

Enc. no	Date	Encounter description
1	1991/6-8	HU asks for a tender from Tietovoima about the delivery of the accounting package, which has just been approved by the State Officials. The tender is accepted immediately. The contract of delivery is signed and the adoption project begins, leading to the use of the software in February 1992.

The process maps are helpful in depicting the software contracting dynamics over several years. By taking a longitudinal position, we can begin to see the emergence of patterns in the trajectories and the significance of alliances with specific software houses (e.g., CCC and SCC). What is revealed is a picture of organizational learning (two-way) as parties adjust and negotiate over time. The maps indicate the importance of establishing and fulfilling successful contracts at an early stage. Moreover, the theoretical lens of Williamson acts as a powerful explanatory vehicle.

It seems that transaction cost theory gives us intellectual machinery to explain our experience in relatively simple terms. In the initial purchase, the goal of efficiency in developing the IS concerned seems to dominate. Therefore, solutions are sought from market. Later, during the use and maintenance phase, the "costs" of using an outside vendor may become too great (cf. SISUH), because the "gatekeepers" must devote their time to act as the intermediates between the end users and the (outside) developers. The development reverts to in-house (Figure 1).

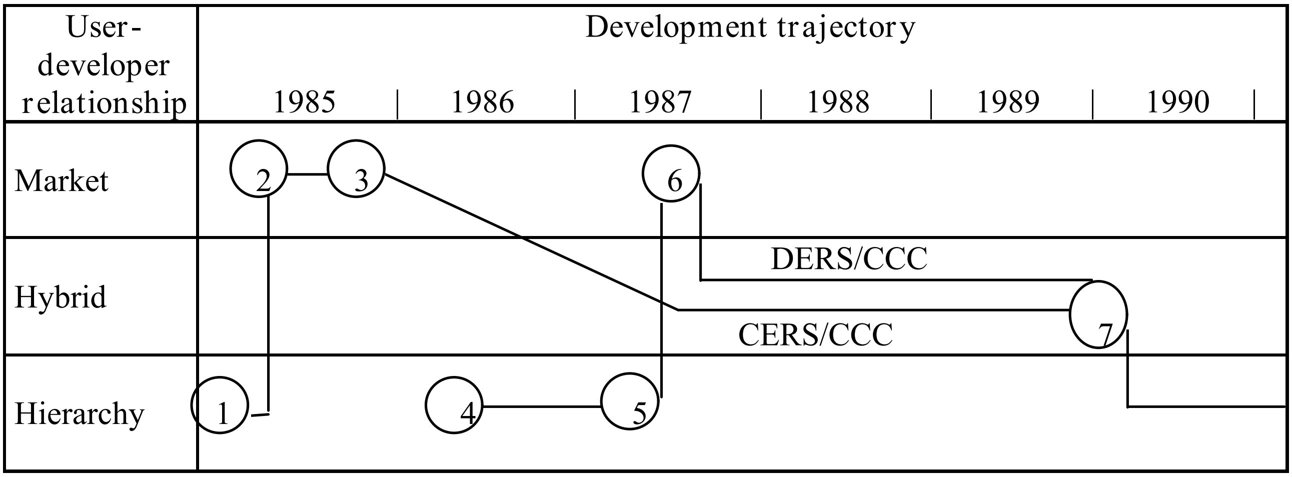

Figure 1. The Development Trajectory of SISUH (CERS and DERS)

We can also give a broad explanation to the shapes of the development processes according to the maturity of the application area. Accounting is a very mature field, and therefore the IS contracts also are of market type (Figure 3). The data processing needs in student administration seem to be rather immature and constantly evolving. Therefore, the ISs in this area need constant tailoring as the users learn new ways to do their business. In our case, this tailoring has meant internal development. Personnel systems seem to be somewhere between the extremes of mature and immature. Payroll functions have traditionally been well developed in forms of software packages, while the new administrative principles of decentralizing decision making have required new (and evolving) ISs. Therefore, the hybrid form of development in this area seems appropriate, especially because the cost of developing a first rate system requires a considerable amount of resources (Figure 2).

Our experiences are in accordance with the IS Procurement Principle by Saarinen and Vepsäläinen:

1. routine applications (common to many organizations, well-specified requirements, such as accounting) can best be implemented by acquiring a software package, i.e., through a market transaction;

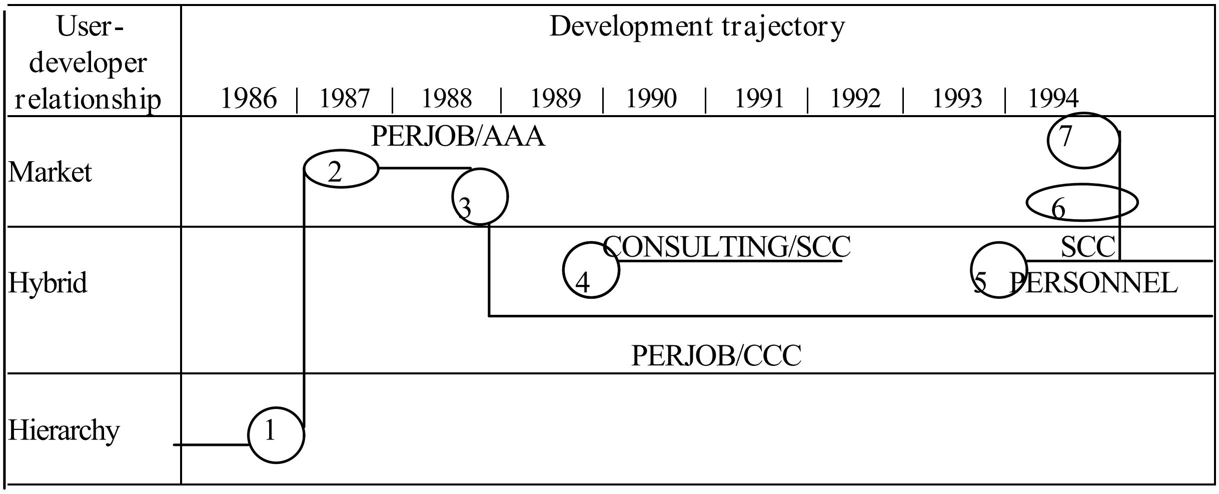

Figure 2. The Development Trajectory of the Personnel Systems

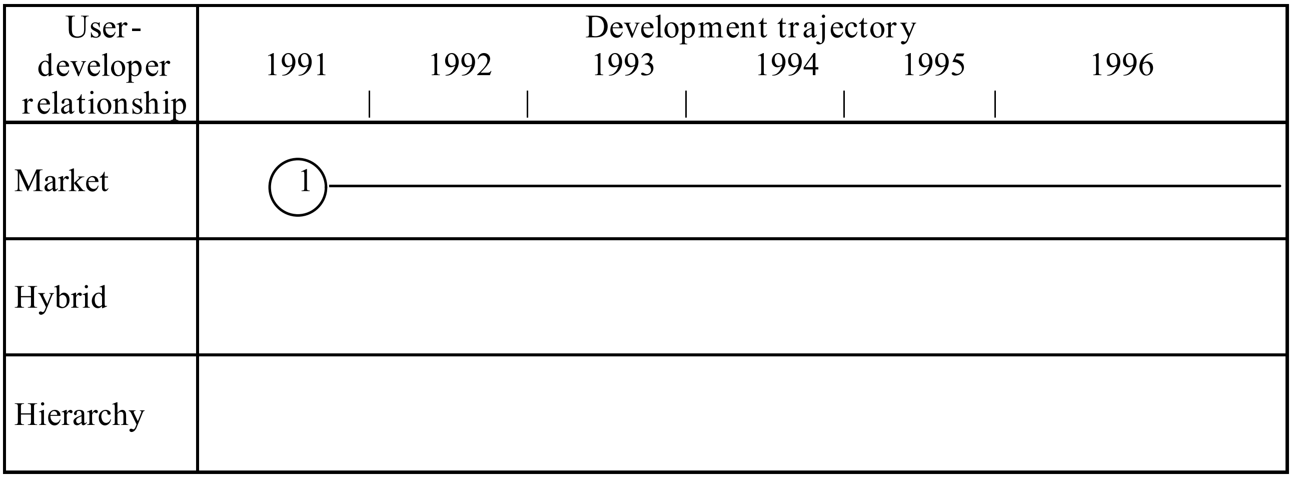

Figure 3. The Development of the Accounting System

2. standard information systems (shared functionality with a group of organizations but variety of detailed requirements, such as personnel systems) require software contracting, which, according to our argumentation, leads to a hybrid form of development organization; and

3. speculative systems (specific to one company and involving high uncertainty of requirements, such as the student record) are best left for internal development, i.e., for a hierarchy.

Saarinen and Vepsäläinen used survey data from 48 development projects. They found some support for the Procurement Principle, but it was not fully confirmed within their data set. Our three systems quite nicely obey the Procurement Principle, thus increasing its credibility. Obviously, more empirical research is needed before it is possible to say how much the intuitively appealing Procurement Principle predicts or explains actual behavior, and under which circumstances it does so.

A model is a simplification of reality. It is well known that several different interpretations can be drawn from the same evidence. Lacity and Hirschheim used two viewpoints toward IS outsourcing: economical (transaction cost theory) and political. The different views complement each other. In this work, we have simplified the reality using only the transaction cost theory. The encounters of our cases have political aspects as well. The originators of our model, Newman and Robey, used the classification acceptance-equivocation-rejection, nearly a political concept. This classification is replaced here by market-hybrid-hierarchy. It seems possible in further research to enlarge our analysis to include the political aspects.

This paper is the first publication in our research project. We will devote further research on the encounters found along the development trajectories of other ISs of HU. This should be interesting for the IS research community, because it seems that there is a lack of in-depth analysis of software contracting negotiations. One of the reasons for this is that the nuances of contract negotiations are not normally revealed to outsiders. Our plan is to use the modeling principles presented here to see which kind of patterns can be identified from a wider spectrum of IS development trajectories.

Our method of conducting this research, reflecting over the actions where the authors have been involved, has strengths and weaknesses (Heiskanen 1995). For example, the access to data is easily established and the observation period can be long with minimal research resources, but there is also the danger of post-rationalization and one-sidedness. This approach may become worthless, even harmful as a research approach, if the practitioner-researcher does not consider the reflective process to be a possibility for personal growth but aims at research results at any price. The danger of contaminated research is here great, because the practitioner has such a control over the production of the research data that it is too easy for him, for a short time frame, to "design" the data to support nearly any argumentation, this in addition to the normal threats of interpretative research. A reflective practitioner who plans to publish her or his findings should constantly fight against these dangers when collecting experiences and analyzing observations. A help in this fight is the awareness of being doubly accountable for the research, first in the research community and then in the practice community.

7. REFERENCES

Asanuma, B. "Manufacturer-Supplier Relationships in Japan and the Concept of Relation-Specific Skill." *Journal of the Japanese and International Economics*, Volume 3, March 1989, pp. 1-30.

Bakos, J. Y., and Brynjolfsson, E. "Information Technology, Incentives and the Optimal Number of Suppliers." *Journal of Management Information Systems,* Volume 10, Fall 1993, pp. 37-53.

Boland, R. J. "In Search of Management Information Systems: Explorations of Self and Organization." Paper presented in the Second Kilpisjärvi IS Seminar, Finland, 1991.

Fitzgerald, G., and Willcocks, L. "Contracts and Partnership in the Outsourcing of IT." In J. I. DeGross, S. L. Huff, and M. C. Munro, Editors, *Proceedings of the Fifteenth International Conference on Information Systems*. Vancouver, British Columbia, 1994, pp. 91-98.

Gersick, C. J. G. "Revolutionary Change Theories: A Multilevel Exploration of the Punctuated Equilibrium Paradigm." *Academy of Management Review*, Volume 16, January 1991, pp. 10-36.

Grudin, J. "Interactive Systems: Bridging the Gaps Between Developers and Users." *Computer*, April 1991, pp. 59-69.

Gurbaxani, V., and Whang, S. "The Impact of Information Systems on Organizations and Markets." *Communications of the ACM*, Volume 34, January 1991, pp. 59-72.

Heiskanen, A. "Reflecting over a Practice, Framing Issues for Scholar Understanding." *Information Technology and People*, Volume 8, 1995, pp. 3-18.

Heiskanen, A., and Similä, J. "Gatekeepers in the Action Structure of Software Contracting: A Case Study of the Evolution of User-Developer Relationships." *ACM Computer Personnel*, Volume 14, 1992, pp. 30-44.

Lacity, M. C., and Hirschheim, R. *Information Systems Outsourcing: Myths, Metaphors and Realities*. Chichester, England: Wiley, 1993.

McFarlan, F. W., and Nolan, R. L. "How to Manage an IT Outsourcing Alliance." *Sloan Management Review*, Volume 36, Winter 1995, pp. 9-23.

McWilliams, A., and Gray, S. R. "Understanding Quasi-Integration." *The Journal of Business Strategies*, Volume 12, Spring 1995, pp. 69-85 (http://coba.shsu.edu/jbs/vol12/no1/12-1-5.html).

Mintzberg, H. "An Emerging Strategy of 'Direct' Research." *Administrative Science Quarterly*, Volume 24, December 1979, pp. 582-589.

Newman, M., and Robey, D. "A Social Process Model of User-Analyst Relationships." *MIS Quarterly*, June 1992, pp. 249-266.

Ouchi, W. G. "A Conceptual Framework for the Design of Organizational Control Mechanisms." *Management Science*, Volume 25, September 1979, pp. 833-848.

Page, D.; Williams, P.; and Boyd, D. *Report of the Inquiry into the London Ambulance Service*. Report commissioned by Southwest Thames Regional Health Authority, United Kingdom, 1993

Powell, W. "Neither Market Nor Hierarchy: Networked Forms of Organization." In B. Staw and L. Cummings, Editors, *Research in Organizational Behavior*, Volume 12. Greenwich, Connecticut: JAI Press, 1990, pp. 295-336.

Richmond, W. B.; Seidmann, A.; and Whinston, A. B. "Incomplete Contracting Issues in Information Systems Development Outsourcing." *Decision Support Systems*, Volume 8, 1992, pp. 459-477.

Robey, D., and Newman, M. "Sequential Patterns in Information Systems Development: An Application of a Social Process Model." *ACM Transactions of Information Systems*, Volume 14, January 1996, pp. 30-63.

Saarinen, T., and Vepsäläinen, A. P. J. "Procurement Strategies for Information Systems." *Journal of Management Information Systems*, Volume 11, Fall 1994, pp. 187-208.

Schön, D. *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books, 1983.

Similä, J.; Kuvaja, P.; and Krzanik, L. "BOOTSTRAP: A Software Process Assessment and Improvement Methodology." *International Journal of Software Engineering and Knowledge Engineering*, Volume 5, December 1995, pp. 559-584.

Whang, S. "Contracting for Software Development." *Management Science*, Volume 38, March 1992, pp. 307-324.

Williamson, O. E. "Comparative Economic Organization: The Analysis of Discrete Structural Alternatives." *Administrative Science Quarterly*, Volume 36, June 1991, pp. 269-296.

APPENDIX

Abbreviations

AAA	A pseudonym for a software vendor
BBB	A pseudonym for a software vendor
CCC	A software vendor
CERS	The centralized part of SISUH
DERS	The departmental part of SISUH
HU	University of Helsinki
KTT	KT-Tietokeskus, an information technology vendor
PERJOB	Personnel system, including also data about jobs of the employees
PERSONNEL	A personnel system that is integrated to payroll
SCC	State Computing Centre, an information technology vendor
SISUH	The Student Record/Information System of the University of Helsinki