

December 1995

Developing Groupware Applications Using Client/ Server with Event-Pipe Method

Nian-Shing Chen

National Sun Yat-sen University

Craig Chao

National Sun Yat-sen University

Meng-Hsiang Hsu

National Sun Yat-sen University

Follow this and additional works at: <http://aisel.aisnet.org/pacis1995>

Recommended Citation

Chen, Nian-Shing; Chao, Craig; and Hsu, Meng-Hsiang, "Developing Groupware Applications Using Client/Server with Event-Pipe Method" (1995). *PACIS 1995 Proceedings*. 34.

<http://aisel.aisnet.org/pacis1995/34>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 1995 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Developing Groupware Applications Using Client/Server with Event-Pipe Method

Nian-Shing Chen
Department of Information
Management
National Sun Yat-sen University

Craig Kuo-Jen Chao
Institute of Information
Management
National Sun Yat-sen University

Meng-Hsiang Hsu
Institute of Information
Management
National Sun Yat-sen University

In this paper we present a new methodology for developing groupware systems. We first propose a three-layer model for groupware systems from the functional point of view. In the model, we divide functions of a groupware system into three major levels, application, coordination and communication. A client/server architecture is then presented for incorporating the three functional layers into a group system. Finally, a technique called client/server event-pipe is used in the design methodology.

Many non-groupware applications built on window systems are available today. Those applications are mainly used for personal computing and run on a single host, so they are not concerned with any coordination and communication. If we can add coordination and communication onto these applications without any modification to the source code, they would then become groupware applications. This approach can save a lot of time and effort while developing groupware systems.

Keywords:

Group Decision Support Systems (GDSS), Groupware systems, Client/Server computing, Window systems, Graphical user interface, Software design methodology

1 Introduction

Computer networking is getting more and more popular, which moves the computing world from personal computing into distributed computing. In the past, most software systems were designed to be used by individuals. They only support the interaction between a user and the system. There is little support for user-to-user interaction even with systems designed for multi-user applications. Today, the merging of computer and communication technologies allow for distributed computing. The technological advances in distributed computing environments further break down barriers of time and geography. Therefore, people nowadays are looking for computer software that can support them when working together in real time and in dispersed locations.

Various research areas have emerged recently that concern the issues related to group systems for supporting group problem solving. There are several different names have been used to describe this type of application, such as groupware [Ellis 1991],

computer-supported cooperative work (CSCW) [Kraemer 1988], electronic meeting systems [Dennis 1988], group decision support systems (GDSS) [Bui 1986, Nunamaker 1989]. In the NeXT world, they use the term "inter-personal computing". Because these many kinds of cooperative systems apply to a broad range of applications, we will refer to them as groupware systems.

Heintz suggests that a GDSS system can operate under three layers of software: an application-layer, a GDSS environment, and an operating system environment [Heintz 1993]. This kind of classification is helpful for seeing message interaction among different layers, such as in the object-oriented architecture described by the author. We also proposed another model for groupware systems from the functional point of view. In this model, we divide a groupware system into three functional levels, application, coordination and communication as shown in Figure 1.

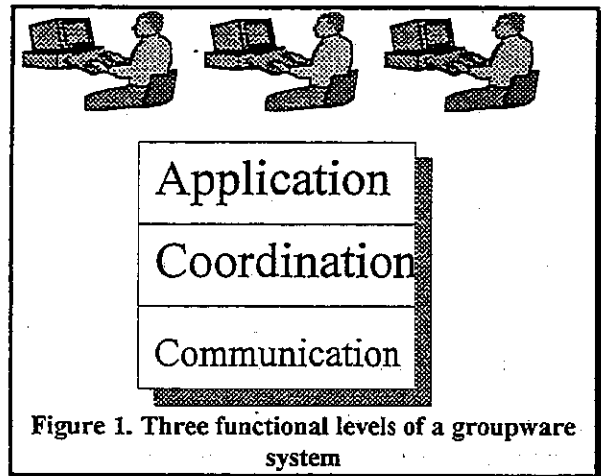


Figure 1. Three functional levels of a groupware system

1.1 Application

Groupware systems are designed to support people working together in many areas. Different groupware systems emphasize and address on different areas of supporting. The top functional layer of a groupware system is its application. For example, a group editor [Ellis 1991] should provide all the editing functions, such as typing characters, copying, deleting and pasting, just like a traditional editor. These kinds of functions are not related to any issues of group activities, and we call them the application functions or domain functions.

This research was supported in part by NSC under the contract NSC84-2416-H-110-009-F2.

1.2 Coordination

The middle functional layer of a groupware system is coordination. Groupware systems need some concurrency control to resolve conflicts among participants' simultaneous operations. Coordination provides an integration and harmonious adjustment of the individual's work efforts toward the common goal of a group. For example, with a group editor, one participant might delete a sentence while a second participant inserts a word into the sentence. To maintain data consistency and integrity within a collaborative group work, it may be useful to adopt a database [Heintz 1993].

1.3 Communication

The primary benefit of groupware is that it allows people to work together in real time, even when separated by geographically. To provide people with distributed working environments, it is preferable to build groupware systems on LANs and/or WANs. Group members' interaction and coordination activities all rely on communication. There is no doubt that a computer-based or computer mediated communication should be provided by any groupware systems. Because communication is fundamental to all groupware systems, it is located on the bottom layer of a groupware system.

2 Client/Server architecture for groupware systems

As described in the previous section, there are three functional layers for a groupware system. We now provide a client/server architecture in Figure 2 to incorporate these three functions into a groupware system.

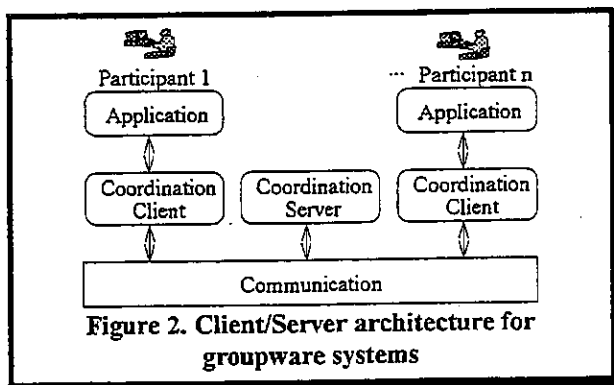


Figure 2. Client/Server architecture for groupware systems

There are a variety of collaborative and/or cooperative tasks in multiple domains and environments. A task addressed to one specific domain of an application would have coordination issues that are different from tasks in other domains. Many participants can be involved in one task at the same time and cause tremendous consistency problems in their work. To simplify the complexity of coordination, it is beneficial to use one coordination server to handle all the consistency issues occurring among participants' activities. The

proposed Client/Server architecture for groupware systems have the following advantages:

1. From the functionality point of view, each layer plays one specific role in a groupware system. This makes groupware designers focus on one issue at a time while developing a complicated groupware system. This can not only reduce the complexity of groupware systems analysis and design but will also improve the modularity of the software systems.
2. The property of the modularized software can increase software flexibility. Whenever a new technology is introduced, groupware developers can easily modify their program to incorporate the new technology simply by replacing the corresponding software module. For example, with current communication technology, transmission speed and bandwidth are not enough to support groupware with video and/or audio applications. High-speed network technology is currently under developing, however, it will be available in the near future.
3. In the Client/Server paradigm, server and client are two autonomous active processes. Hence, using client/server architecture, software can get maximum utilization of machine resources and people can acquire great concurrency while many participants are working together. Each participant runs the same application together with a coordination client on their own workstation to join one group activity. A coordination server along with these coordination clients provide a harmonious and consistent shared environment through communication to all the participants.

It is worth noting that the coordination server can be run on a dedicated workstation or on one of participant's workstations when a centralized coordinator is considered. For reason of efficiency and robustness, distributed coordinators running on different machines can also be applied. However, the implementation of distributed coordinators is more difficult than then centralized one.

3 The Design Methodology

3.1 The Window Systems

A window is an area of the terminal/workstation screen that behaves like a separate screen. This allows windows to support user communications with many applications on a single workstation. A *window system* [Johnson 1990] contains a window server and some window clients. There is only one window server per workstation but multiple window clients can communicate with this one window server. Window clients and window server can be on the same host, or on the different hosts

communicating over a network. A window server manages a workstation screen and communicates with window clients to perform screen operations.

For example, four applications running on four different workstations may keep four windows open on one workstation, one window per application. The user can interact with different applications by switching between windows by using a keyboard, a mouse, or other pointing input device. Since many windows can be opened on the same physical screen of a workstation, a window manager is responsible for managing the workstation physical screen. The manipulation of each window in a window system is event-driven through message passing. A window system keeps track of all the windows' status and takes proper actions whenever an event occurs. Figure 3 is an example to show the flow of events in a window system.

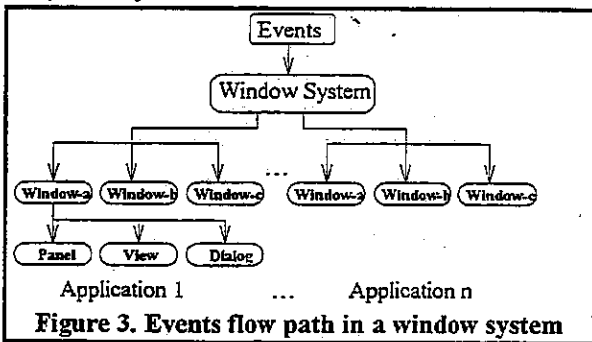


Figure 3. Events flow path in a window system

Most applications designed today are built on window systems for supporting graphical user interface (GUI). As described above, an application may open many windows on a workstation and many applications may run on one workstation at the same time.

3.2 Events Recording and Playback

Since all events first pass through the window system, it is possible for us to record and play back events in a window system, with the capability of implementing event-based macros or complete self-running demonstrations for window applications. Some commercially

In a basic window system, events are passed from the window system to their corresponding application windows.



In a window system with event recording and playback capability, events can be passed from window system to event recorder and then to application windows. That is, events can be saved by the event recorder and replayed whenever you want.

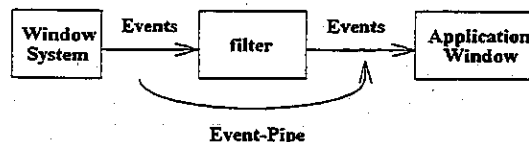


From the discussion above, we know that window

systems are event-driven and events can also be captured from window system and played back to application windows. It is possible for us to do additional control or some kind of value-added work over the captured events. In other words, we can add any extra functions desired by writing a program which takes captured events as input and produces processed events as output. The program acts something like a filter in UNIX. We will call this kind of mechanism in window systems an *event-pipe*. The next section describes how to apply event-pipes to developing groupware applications.

3.2 Using Event-Pipes in Window Systems

In a multitasking window system environment, functions of every object are activated by events. There are many types of events, such as mouse click, keyboard input and pull-down menu. If we can capture all the input events of an application, and pass the processed events back to the application after adding on coordination issues, the application can then take proper actions corresponding to groupware requirements. We can also redirect the processed events to all applications running under the same group, so that those applications act like a groupware application. This means that we can develop a groupware application without modifying the source code of the original application.



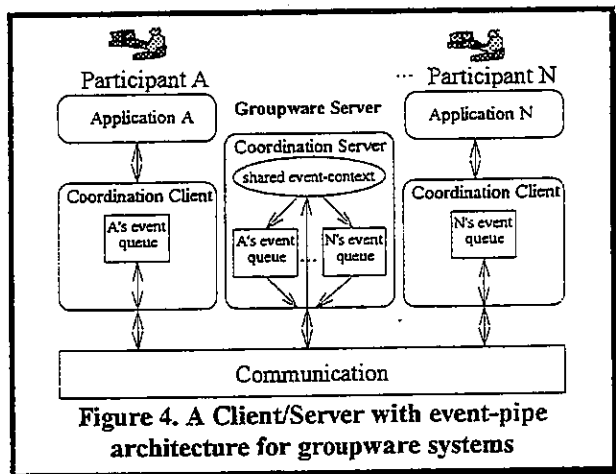
In an event-driven message passing environment, all generated events are sent to the system queue first, and then passed to the event queue of corresponding applications. Since window systems provide mechanisms for developers to record and play back these events, it is applicable to implementing event pipes.

3.3 Putting it together: The Methodology

By applying the client/server architecture for groupware systems described in Section 2 and the concepts of event-pipe discussed in the previous sections, we are now ready to describe the design methodology for groupware systems development. The proposed design methodology is a Client/Server Event-Pipe Architecture (CSEPA) for groupware systems as shown in Figure 4. To explain the design methodology in more detail, let's first define some terms used in the figure.

A context is a run-time working environment of an application. In a window system, each application is event-driven through message passing. Hence, we can use an event or a sequence of events to represent the context of an application. We will use event-context to describe the working environment of an

application. By recording and playing back an event-context, we can set an application to a desired working environment. In a client/server groupware system, each participant runs a groupware client to work concurrently with the others. The groupware server should maintain a current group outcome which is contributed to by every client. Let's call the group outcome a shared event-context.



We now describe the sequence of actions taken by each client and server working within the architecture as follows:

Client:

1. Register with server on start-up.
2. Capture all events from application.
3. Send captured events to server.
4. Process events received from server.
5. Pass processed events to application.
6. Do some add-on functions.

Server:

1. Respond to clients' registration requests.
2. Process events received from clients, keeping track of each client's event context.
3. Send processed (with add-on coordination) events to clients' corresponding event queues.
4. Maintain group current outcome in shared event-context.
5. Handle clients' working environments.

Most groupware systems need to have the property of "What You See Is What I See" [Stefik 1987, Ellis 1991]. This property can be implemented using a client/server with event-pipe technique proposed in the design methodology. Whenever a groupware client A has just done an action, for example drawing a circle, the corresponding events of this circle are sent to the groupware server. The coordination server then sends the events to every clients' event queues. This makes the circle appear on each groupware client's window. Before passing the events to the application for displaying the circle, each coordination client should preserve its current event-context. Right after playing back the events of client, each coordination client then restore its

original event-context such that each participant can resume his/her work. The sequence of actions, saving event-context, playing back events and restoring event-context is called event-context switching.

At present, a lot of non-groupware applications are available. If we can add these applications, without any changes to the source code, to the corresponding coordination and communication software, they would then become groupware applications. This approach can save a lot of time and effort in developing groupware systems. Applying the proposed design methodology can help us to achieve the goal.

For applying the methodology, applications with the following characteristics are preferable:

1. They run on an event-driven message passing window system.
2. The corresponding groupware can be easily divided into application, coordination and communication.
3. Each participant runs the same application to work together with the others.
4. Existing application software is available; groupware developers need only add the functionality of corresponding coordination and communication.

Applications which are not suitable for applying the methodology:

1. Those that can not be run on window systems.
2. The corresponding groupware can not be easily divided into application, coordination and communication.
3. Each participant runs different applications to work together with the others.
4. None of existing application software is available.
5. Programming the application software is just a piece of cake.

4 The Prototype Implementation

We have presented a methodology with client/server event-pipe architecture for developing groupware applications. The architecture can be applied to any window systems. We here have designed a group paint prototype on the window system of chinese MS-Windows 3.1. Many people are familiar with MS-Windows which has a bundled paint program called Paint-Brush. Naturally, it is not a groupware because Paint-Brush can support only one user at a time. We will use this existing nongroupware program to implement a group paint prototype by applying the proposed CSEPA method. Any window systems with event-pipe mechanism support as well as adopting client/server programming can easily apply CSEPA to design groupware applications. The prototype demonstrates this view point.

As the network communication of this prototype,

we use the shareware Trumpt winsock.dll. It is compliant with Microsoft's winsock 1.1 standard for developing TCP/IP based applications. Due to the popularity of TCP/IP protocols, our group paint application can be run on Internet as well as most of networks supporting TCP/IP. Designer can also use any other communication protocols while applying CSEPA method. The modularity design of the proposed architecture makes it free from any specific protocols.

Figure 5 is the screen dump picture of original Chinese Paint-Brush. Figure 6 is the screen dump picture after add-on one extra menu called "Group Paint" with three options "Redraw", "cray" and "nschen". Here cray and nschen are two users currently cooperating on this group paint system. Now they can work together to create a picture with topic as "My_Sweet_Home". A few minutes after, a beautiful picture is completed as shown in Figure 7. To see what has been done by cray, just choose the option "cray" from the Group Paint menu. One can see in Figure 8, only cray's works are shown. Figure 9 expresses the same thing except the user is changed to "nschen". As a final note, a realtime groupware like group paint should provide the feature of WYSIWIS (What You See Is What I See) to all the participants. For example, one participant draws a circle on his screen, this circle should be shown on all the others participants' screen at the same time. The coordinated issues required for the group paint can then be implemented by the Group Paint Server.

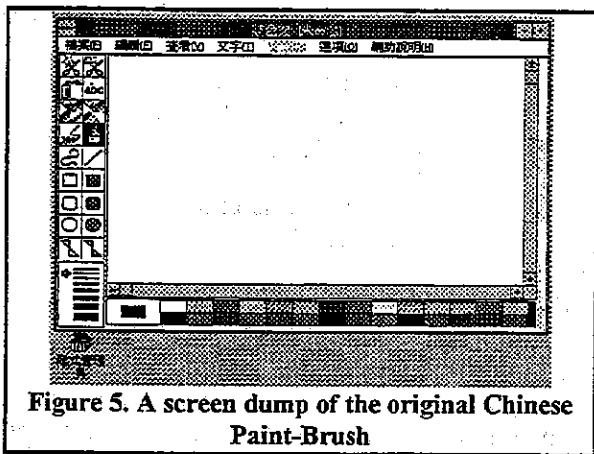


Figure 5. A screen dump of the original Chinese Paint-Brush

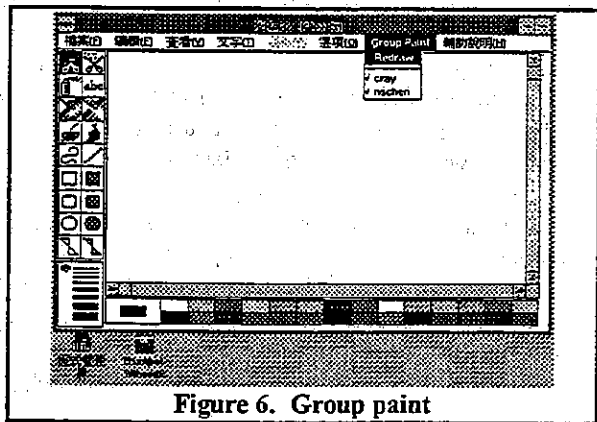


Figure 6. Group paint

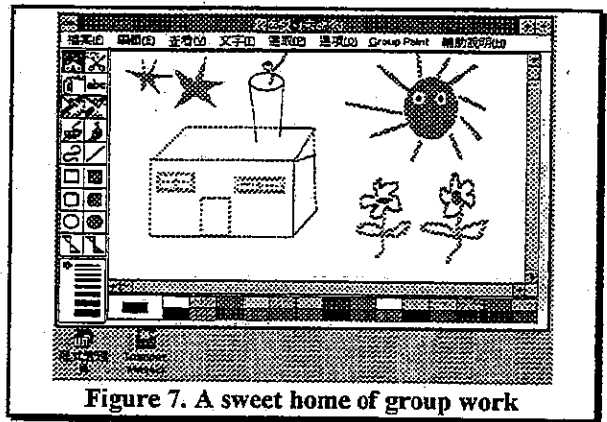


Figure 7. A sweet home of group work

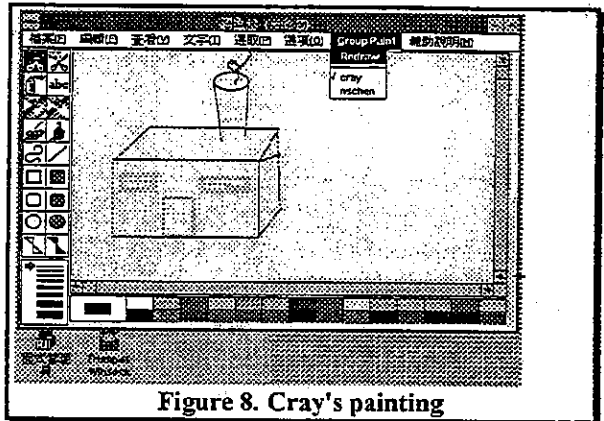


Figure 8. Cray's painting

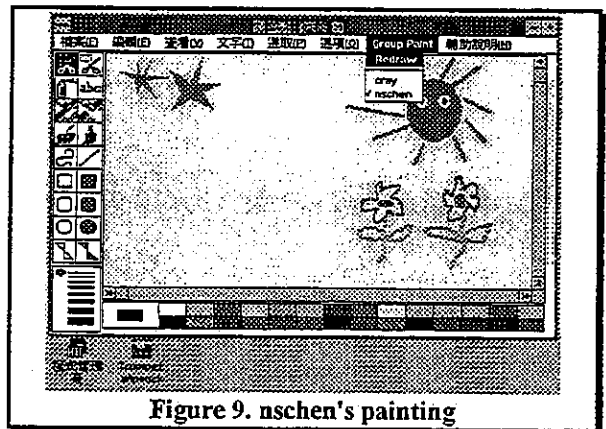


Figure 9. nschen's painting

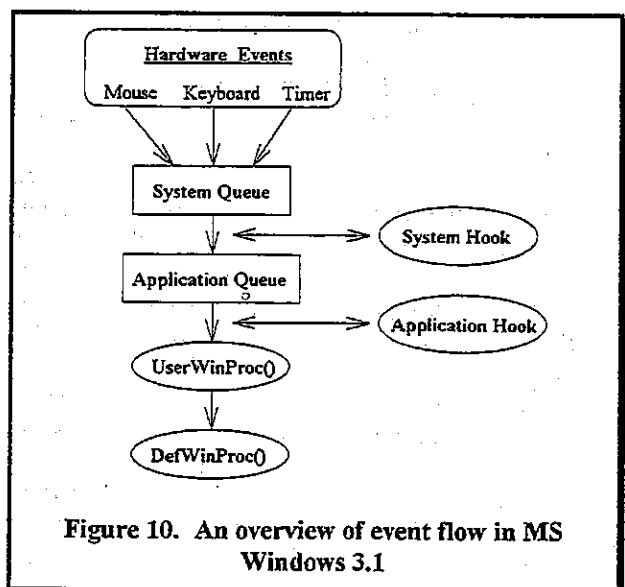


Figure 10. An overview of event flow in MS Windows 3.1

In MS-Windows 3.1, there is a function called "hook". By using the system calls of SetWindowsHookEx(), CallNextHookEx() and UnHookWindowsHookEx(), we can easily capture system or application related events, and then pass them back to the application after some kinds of proper processing. Figure 10 shows an overview of event flow in MS-Windows 3.1.

In addition, we also use some object-oriented programming techniques, such as the subclassing functions supported by MS-Windows 3.1. By using subclassing, we can add some new menus to Paint-Brush to enable the group paint client to reflect some concurrency control actions. With this capability, we can provide the end user with a more user friendly and transparent environment while several users are editing a shared object.

In a window system, only one active window is allowed to accept mouse or keyboard events at a time. Hence for the non-active windows, we have to keep their corresponding events in the event queue of the coordination server. Once a non-active window becomes active, the events are sent from the server to the corresponding coordination client and then passed to the active window.

For simplicity reasons, we implement a centralized coordination server for the prototype. However, it is also possible to implement distributed coordination servers to support more users or to improve system performance. The proposed architecture has this flexibility of extension.

5 Conclusions

We have proposed a design methodology for groupware systems development. The major feature is that it can develop new groupware systems from existing software running on window systems, such as X-Windows, MS-Windows and MAC-Windows, without any modification to their source code. This approach can save a lot of time and effort while developing groupware systems. The methodology has the following advantages:

1. It is good for rapid prototyping implementation of groupware systems.
2. Many powerful applications are very complicated, such as Excel, Word, etc. Rewriting the software would require a lot of additional effort to develop the corresponding groupware.
3. The source code of most application software are not available to groupware developers. Even though developers would like to design such applications, they may lack the necessary domain knowledge or experience.
4. Many new non-groupware applications software running on window systems are undergoing continued development. Once they become available, by applying the design methodology, groupware developers

can easily incorporate them with the corresponding coordination and communication modules. Those applications would then become groupware applications without additional work on the applications themselves.

5. This approach provides a good way for software designers to do "value-added" work on existing software.

The next step for us is to implement more groupware applications on other window system platforms, such as X-Windows, MAC-Windows or NeXTSTEP, by applying the design methodology. Those prototypes can help us to explore the implementation issues needed while adopting the methodology.

References

- Bernstein, P.A. "Middleware-- An architecture for distributed system services", DEC, 1993, Cambridge Research Lab.
- Bui, T.X. and Jarke, M., "Communications design for Co-op: a group decision support systems", ACM Trans. on Office Information Systems, 1986, Vol. 4, No. 2, 463-486.
- Chen, N.S. and Hsu, M.H.(1995), "Developing Groupware Applications using Middleware Support", accepted by 1995 ISDSS Conference.
- Dennis, A., George, J., Jessup, L., Nunamaker, J.F. and Vogel, D., Information technology to support electronic meetings, MIS Quarterly, 1988.
- Ellis, C., Gibbs, S. and Rein, G., "Groupware: Some issues and experiences", Communications of the ACM, 1991, 34, No. 1, 39-58.
- Heintz, T.J., "An object-oriented architecture for group decision support systems", Group Decision and Negotiation, 1993, Vol. 2, 5-25.
- Johnson and Reichard, "Advanced X-Window applications programming", MIT Press, 1990.
- Kraemer, K.L. and King, J., "Computer-based Systems for Cooperative work and group decision making", ACM Computing Survey, 1988, Vol. 20, 115-146.
- Liang, T.P., Lai, H.C., Chen, N. S. and Wei, H.S., "When Client/Server Isn't Enough: Coordinating Multiple Distributed Tasks", IEEE Computer, 1994, Vol. 27, No. 5, pp73-79.
- Lim, F.J. and Benbasat, I., "A communication-based framework for group interfaces in computer-supported collaboration", IEEE, 1991, pp610-620.
- Morrison, J., Morrison, M. and Vogel, D., "Software to support business teams", Group Decision and Negotiation", 1992, Vol. 2, 91-115
- Nunamaker, J.F., Vogel, D. and Konsynski, B., "Interaction of Task and Technology to Support Large Groups", Decision Support Systems, 1989, pp139-152.
- Patterson, J.F., "Comparing the programming

demands of single-user and multi-user applications". In Proc. of User Interface Software and Technology, 1991, pp87-91

Stefik, M., Bobrow, D.G., Foster, G., Lanning, S., and Tartar, D., "WYSIWIS revised: Early experiences with multiuser interfaces", ACM Trans. Office Information Systems, 1987, Vol. 5, No. 2, 147-186.

Electronic Marketplace with Multimedia Representation: SEA of Flowers

Ho Geun Lee
University of Science and
Technology, Hong Kong

Clive Wrigley
McGill University
Canada

Rene Wagenaar
Erasmus University
The Netherlands

Abstract

This paper reports on an investigation into the feasibility of an electronic marketplace for the florist chain where cut flowers and potted plants are traded. Despite its potential advantages over conventional auction markets, introducing the electronic market into the florist chain has been challenged by difficulties of product representation. We delineate an electronic auction system that incorporates computer-based trading with multimedia representations for cut flower trades. The electronic auction system decouples product flows from market processes. Since the decoupling brings economic efficiencies as well as risks to traders, the adoption of the electronic auction is highly dependent on decisions of potential market participants. The transition to the electronic market becomes feasible only when benefits gained by the decoupling exceed risks involved. A business simulation game is proposed to test economic behavior and theories behind the transition to this new electronic alternative. Simulated Electronic Auction (SEA) of Flowers, designed for the simulation game, is presented.

1. Introduction

The innovations in information technologies of the last two decades have radically reduced the time and cost of processing and communicating information. The use of this electronic communication allows companies to take advantage of two types of inter-organizational information systems: *electronic integration* or *electronic markets* (Malone et al. 1987). When a supplier and a procurer use information technology to create joint processes in value-added chains, they are taking advantages of the electronic integration effect. Electronic Data Interchange aims primarily to exploit the electronic integration effect. One benefit of this effect is curtailed time and cost of communications. Another benefit is that the supplier can ship the products *just in time* for use in procurer's manufacturing process.

The electronic market effect occurs in the case of computer-based markets where information technologies serve as intermediaries between buyers and suppliers. Electronic markets substantially reduce the need for buyers and suppliers to contact a large number of alternative trading partners individually (Bakos 1991). Acting as either broker or auctioneer between buyers and sellers, electronic market systems facilitate parts or the whole transaction process: from search to contract formation and to trade settlement.

Electronic markets are of increasing interest because of their advantages over conventional markets. Screen based trading systems can become 24-hour global markets and provide regulatory advantages such as electronic audit and surveillance. In addition, the use of information technologies reduce transaction costs incurred in the market process.

The largest number of flower transactions in the world takes place in the Netherlands. Dutch auctions play an important role to tune supplies with demands of cut flowers and potted plants around the world: growers bring their products into Dutch auctions, where buyers get together to purchase flowers. In spite of potential advantages of electronic alternative over current auctions, introducing electronic markets into the florist chain was challenged by difficulties of product representation (van Heck and Groen 1994). Unlike other commodities, such as cotton or securities, flowers can hardly be represented using text only. Electronic market systems for the florist chain need to incorporate multimedia representation (picture) for the product representation. Thanks to recent innovations in multimedia technology and wider bandwidth communications such as ISDN and ATM, the merge of computer-based trading systems with multimedia communications has become economically feasible.

Without intervention of government authorities, the transition from conventional auctions to electronic markets is feasible only when all intended market participants are convinced of its benefits. Electronic markets in the Dutch florist chain aims to decouple product flows from market processes. This decoupling brings both benefits and risks to market participants. In particular, buyers may face uncertain product qualities because pictures substitute for real flowers in electronic marketplaces. Unless the advantages of the electronic market exceed its risks, the transition is not feasible even though other market participants, such as growers and auctions, are eager for the new electronic alternative.

The principle research question is: *Can the benefits of electronic markets be sufficient to overcome its risks?* Data from real auction markets do not allow the direct testing of trader's choice between different market institutions. Analytic models of trader choice also cannot lead to a unique prediction of traders' behavior. Recently, laboratory experiments have become recognized as a useful and valid way to test economic behavior under different market microstructures, mainly in financial markets (Smith 1982, Clemons and Weber 1991). The trade-off between uncertain

product quality and increased market efficiency is subject to bounded rationality of market participants. The dynamics of traders' behavior can be captured only by a business simulation game. Additionally, the experimental results of the simulation game can provide insights on the implementation of electronic markets later on.

We delineate Simulated Electronic Auction (SEA) of Flowers, an inter-organizational system that simulates electronic markets for the florist chain. The objective of the simulation game is two fold: (1) to demonstrate the feasibility of electronic markets to market participants, and (2) to gain insights into the best design for the full-fledged implementation. This paper describes the initial design of SEA of Flowers, together with incentives to introduce it to flower markets. Real experiments with detailed hypotheses will be carried out in the near future.

Section 2 describes the Dutch florist chain, together with its current clock auction systems. In section 3, we discuss electronic auctions under a market taxonomy and explain the institutional rules which govern trading. This is followed by a discussion of driving forces behind the transition to the electronic auction: potential benefits and risks involved in the electronic auction. Section 5 describes SEA of Flowers, including its design philosophy, the underlining economic model, the use of business scenarios and the automated roles used. Finally, we discuss the significance of this prototype design and future research directions.

2. Flower Markets

2.1 International Florist Chain

Flower industry, or florist chain, is one of the major economic sectors in the Netherlands. The flower industry consists of cultivation and trade of cut flowers and potted plants. Holland has an almost 80% share of the world market. In 1993, the total trade volume exceeded US\$ 3 billion and the market is growing at an annual growth rate of 10%. There are several organizations involved in the international florist chain: growers, buyers, auctions, banks and transport companies.

The auction is a co-operative of growers and is obliged to sell all products of its member organizations (growers) through their auction processes. For the sale of their products, the growers pay auctions between 5 and 6 percent commission. Seven international flower auctions are currently under operations in the Netherlands. Bloemenvelling Aalsmeer (VBA), the largest auction located near Amsterdam, is a co-operative of about 5,000 growers and controls 43% of the market. On a typical day VBA processes trading of around 14 million cut flowers and 1.5 million potted plants. Buyers can be large organizations, such as exporters and wholesalers, as well as small ones, such as florists or street vendors. Banks and transport companies have offices in auctions to expedite payment and delivery.

2.2 Dutch Clock Auction

Since cut flowers are highly perishable goods, fast transaction and delivery are vital factors in the florist chain. Auction markets perform two functions for this purpose: market and distribution. Flowers and potted plants, brought

to the auction by growers, are sold at the price determined by the Dutch auction process (market), and then handed over to buyers (distribution).

Flowers are cut and brought to the market during the evening or night preceding the auction. Growers either bring the cut flowers themselves or have them delivered by transport companies. Once flowers arrive at the market, they are kept in large refrigerated areas until the time of auction. All the flowers are inspected by the auction's own inspectors (Flower Master) who check the quality of each lot. Their remarks are recorded in computers so that they can be displayed during the auction.

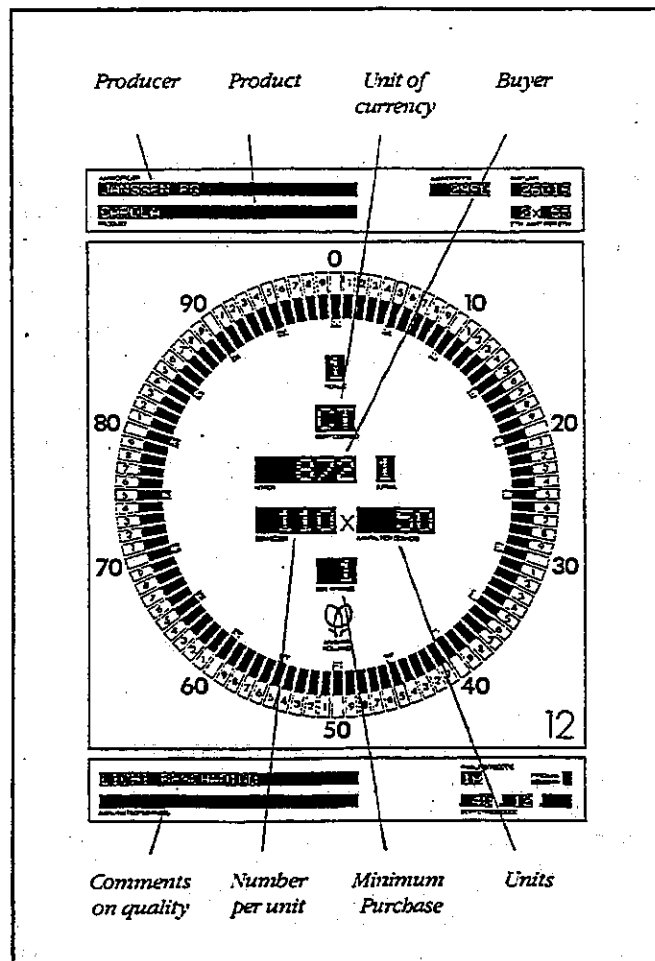


Figure 1 Clocks for Dutch Auction

The auction takes place in auction rooms which are equipped with clocks and buyers' bid desks. Auction normally starts early in the morning and goes on until all the products are sold. When a buyer registers, he is given a buyer's card coded with his own number. When the buyer's card is put into the buyers' bid desk in the auction room, it unlocks the push button needed for bidding. The products are sold by Dutch auction rules: an auctioneer begins by asking a high price and gradually lowers the price until some bidder takes the offer. Clocks are used for this purpose. The clock runs from the highest to the lowest prices and a buyer stops the clock by pushing a button when the running light along the clock reaches the price he

wants to pay. The person who first stops the clock becomes the buyer, and his number appears on the clock board. During the auction process, the clock board also displays product information such as producer, product name, supply lots, comments on quality (provided by the Flower Master), and so on (See Figure 1).

It should be noted that flowers go through the auction hall during the auction so that buyers can make purchasing decisions based on what they see. After being sold, the lots are driven out and are loaded into vans or trucks arranged by buyers. Buyers pay cash at the cashier's office or have the bill sent through their banks. In this way, products auctioned in the morning can be sold the same evening or the next morning at florists and retailers in Europe, USA, Canada and practically any other part of the world.

3. Electronic Auction (EA)

3.1 Market Structures and EA

Market structures can be classified into four types depending on how traders search for their counterparts: (1) direct search markets where traders must seek out compatible trading partners independently; (2) brokered markets where traders employ agents (brokers) to conduct the search; (3) dealer markets where traders exchange goods with dealers who hold their own inventories and are always willing to buy and sell; and (4) auction markets where traders transact directly against the orders of other traders by communicating through a single central intermediary (Garbade 1982). Information technologies can be employed to create electronic markets in either brokered markets, or dealer markets or auction markets.

Computer Reservation Systems (CRS), such as SABRE or Apollo in the airline industry, can be classified as electronic brokerage. By subscribing to SABRE, travel agencies can have access to flight schedules and fares of airline companies around the world. Another example of the electronic brokerage is FAST, a computer network broker for electronic parts and components (Neches 1993). A consumer can send a request for quote (RFQ) to FAST via Internet or commercial networks. FAST distributes the RFQ and receives quotes from vendors in an attempt to link the customer's request with the best quote. Electronic shopping systems such as CompuServe or Prodigy can also be characterized as an electronic brokerage, as they help consumers to search for products which best fit their preferences.

Well-known examples of electronic market systems in dealer markets are SEAQ in the International Stock Exchange in London, and NASDAQ in the United States (Cohen *et al.* 1986). In the SEAQ, dealers display their quotes over computer terminals so that customers can find the best dealer bid or the best dealer ask. Similarly, dealers can expose orders to the market by displaying quotes on the NASDAQ system in the OTC market in the United States. NASDAQ and SEAQ do not execute trading: they simply help investors execute at the best price by displaying dealer bid-offer quotes in electronic boards.

Computer-based trading systems in auction markets (electronic auction) automate order matching between buyers and sellers. The electronic auction was pioneered by Toronto

Stock Exchange's CATS (Computer Assisted Trading System), where orders to sell or to buy are electronically submitted and automatically matched by a certain matching rules based on price and quantity (Freund 1991). Other examples in financial markets include Instinet, INTEX, SOFFIX, and Globex (Cohen and Schwartz 1989). Electronic auction markets are also introduced in some commodity spot markets. For example, TELCOT is an electronic auction system for cotton trading (Lindsey *et al.* 1990). TELCOT automates all aspects of cotton trading, from the matching of buyers and sellers to the transfer of funds and titles. Electronic markets for the florist chain belongs to this category since orders to buy and to sell are matched by computer-based market systems.

3.2 Trade Governing Rules in EA

In electronic auction systems, bids and offers for products are submitted by traders via computer terminals to a computer-based market system. Buyers and sellers can place their orders using standard messages and protocols over commercial networks. When order messages are received by the computerized market system, they are automatically entered into the database. The trade match algorithms then transform the streams of bids and offers into transactions based on trade governing rules.

Since orders are matched automatically, trade governing rules used for the order matching are crucial to all market participants: buyers, sellers and market makers (trading system developers). Continuous trading and periodic call trading have been widely used as trade governing rules in financial auction markets (Whitcomb 1985). In the continuous trading, transactions occur whenever a buy and a sell order cross. In the periodic call trading, by contrast, orders are accumulated over a period of time, rather than being transacted immediately, and are transacted later in a batch. That is, the periodic call trading aggregates buy and sell orders and seeks a market price that equates supply and demand.

Neither continuous trading nor periodic call trading is appropriate for a trade governing rule in SEA of Flowers because of differences between flower markets and financial markets. Since SEA of Flowers aims to replace current flower markets, its trade governing rules should be as close to the Dutch clock auction as possible. In addition, the electronic auction need to provide buyers with product information, such as price, quality and growers, so that buyers can decide their bid prices before the auction is held. This contrasts with financial markets, where buyers can make purchasing decisions without seeing products.

SEA of Flowers provides an opportunity to test diverse trade governing rules that can be adopted by trading systems developers. Since different rules result in different market prices and performances, the decision on the trade governing rule is critical when electronic markets emerge. The SEA of Flowers prototype employs a trade governing rule that is implementable as well as close to current price discovery mechanisms in flower auctions. Other forms of trade governing rules can be implemented to explore alternate institutional rules.

4. Decoupling of Products Flows and Markets

Current flower auction markets refer to a physical location where suppliers and buyers get together to sell or to buy flowers or potted plants. As a result of market formation, various bid and ask prices are quoted and goods are handed over from suppliers to buyers at market prices. Product flows are coupled with the market process that determines the actual transaction prices. Electronic auction systems, on the other hand, decouple product flows from price determination mechanism in markets. Bids and offers for products are submitted electronically and computer-based trading systems determine the market price based on bids and offers. Products stay at sellers' places during the market process and flow after being sold out via on-line transactions.

The decoupling of product flows from the market process may result in risks as well as benefits. Before discussing the design of the electronic auction system for the florist chain, we elaborate its potential benefits and pitfalls. In the florist chain, the decoupling can be advantageous from three perspectives: (1) interactivity between traders and market, (2) network externality, and (3) efficient transactions.

4.1 Increase of Interactivity of Traders with Markets

Growers: In general, sellers establish reservation (ask) prices because they do not have perfect information about the consequences of their actions in markets. The reservation price plays a role as sequentially rational rules under incomplete market information (Stigler 1961). In current Dutch clock auctions, sellers are not allowed to specify their reservation prices. Once being cut, flowers should be sold out whatever the market price is, since cut flowers are perishable goods. The auction market sets up the minimum price for each product offered to the market. But the minimum price is a system price to protect market prices, rather than supporting reservation prices of sellers. For a grower, a reservation price reflects individual costs and profit margins. If product flows are decoupled from the market process, a grower can specify his reservation price, i.e., the lowest price at which he is willing to sell. If there is no buyer who is willing to pay higher than the grower's reservation price, a grower may withdraw his products from the market and offer them to the market again later on, since flowers are not harvested until sold. However, there are only a few days of flexibility for fresh cut flowers.

Buyers: Decoupling allows buyers (wholesalers and exporters) to respond flexibly to local retailers' demands. In the current auction system, buyers can get the information of available products only on the day of auction. Decoupling can expand this time horizon to several days. This expansion is significant to buyers who are in principle sellers in retailer markets. Prices of flowers change significantly day by day depending on supply and demand. For instance, it is not uncommon that a price of rose sometimes varies up to 20% or 30% in sequential trading days. If the electronic auction provides a database of products in the market, together with their auction

schedules, big buyers can communicate with retailers based on the database. They can use the response from retailers to make purchasing decisions, such as what to buy, how many lots to buy, and how much to pay. Since buyers have better information in advance on the supply from growers as well as the demand from retailers, they can come up with better bidding strategies, increasing the possibility of higher profits. This interaction with auction and retailers should allow buyers to smooth out the supply and demand fluctuations.

4.2 Network Externality

The benefits realized by individual participants in an electronic marketplace increases as more organizations join the market. This property, known as network externality, can affect the dynamics of the introduction and adoption of electronic market systems (Katz and Shapiro 1985). Electronic market systems with large installed bases create more value for their participants, who are provided with a wider selection of potential buyers and sellers. There are seven international flower auctions in Holland currently under operation. Each auction market has several auction rooms: an individual auction room can accommodate from 200 to 400 buyers. For example, VBA has five auction rooms, each with two to four clocks.

In current clock auction systems, a product offered can be exposed to buyers sitting in a single auction room. Bids of buyers are also restricted to products auctioned in a single auction room. The electronic auction can remove physical barriers to integrating several auction rooms into a virtual single market. In the long run, the electronic auction also paves the way for integrating seven auctions into the larger international market. This integration will enable buyers and sellers to find better trading partners in terms of price and quality.

4.3 Efficient Transactions

Economic efficiency can be gained when information flows for the market process are decoupled from the product flows. Because traders place bids and offers through computer terminals, they do not have to incur travel costs to come to the markets. The decoupling also brings an economic efficiency to trade settlement: delivery. In the florist chain, growers specialize in one or two types of flowers, thus cultivating a few products in a large scale. Buyers purchase a wide range of flowers from several growers and bundle them into packages for retailers. A wholesaler purchasing a large number of lots from a nearby grower may have goods delivered directly to his location, if it is cheaper than current indirect deliveries. If the direct delivery is not justified economically, traders will continue to use the current distribution at auction markets.

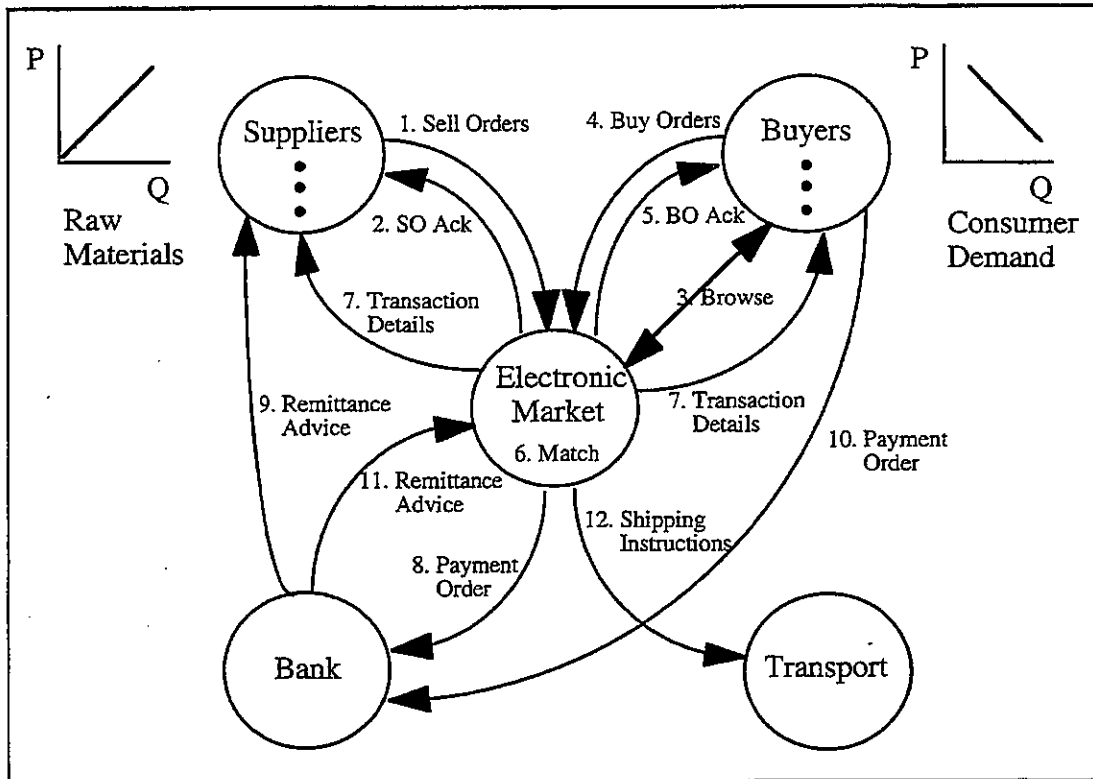


Figure 2: SEA Games Scenario

Auction market organizations have strong motivation toward decoupling product from information flows. They can accommodate ever increasing trading volumes without expansion of physical infrastructure. With annual growth of 10% in the trading volume, all seven auction markets have been increasing their storage spaces and auction room capacities. Even VBA with a refrigerated space of 30,000 m² is currently concerned with its lack of capacity to handle daily transactions. Direct delivery from growers to buyers, if any, will ease this capacity problem. Even in the case of indirect delivery, the decoupling enables auction organizations to utilize the infrastructure more efficiently. Since flowers and potted plants can be delivered to the market by flexible schedules, the use of storage space can be curtailed or even eliminated. This contrasts with the current distribution method, in which all products offered to the market have to be delivered and stored altogether by the night before the auction.

4.4 Product Representation using Multimedia

In the current clock auction system, markets provide a centralized product quality control point, since all products are delivered and inspected by a professional Flower Master. The Flower Master checks the quality of each supply lot and grades them accordingly. Buyers have an opportunity to inspect the quality themselves before the auction and also during the auction when the products pass through the auction hall. Buyers decide their bid prices on the basis of the remarks made by the Flower Master and on the basis of

what they see. In the electronic auction, by contrast, buyers purchase flowers without seeing them. Pictures in the product database (product catalog) replace the original products. Therefore buyers may face uncertain product quality.

There are several aspects which should be addressed to overcome this limitation. First, the technology of multimedia should provide high quality pictures good enough to substitute for original flowers. Second, this multimedia representation and communications should incur reasonable costs so that all market participants can afford it. Finally, markets need to have a different product control mechanism to ensure that the picture accurately represents the product quality.

Our conclusion from the above analysis is that the formation of electronic auctions results when the risk of uncertain product quality is less than the benefits gained through the decoupling. Once the benefits outweigh the uncertain quality, transition to the electronic auction becomes feasible. Analytic methods cannot determine this trade-off point since traders' judgments involve a certain degree of subjectivity. Alternatively, fully implemented systems are expensive ways to find negative results. We propose that a business simulation game be used to test economic behavior and theories behind the transition to electronic auctions. SEA of Flowers is designed for this experimental purpose. Finally, the results of market simulations can provide insights on the implementation of real electronic markets.

Document	Document Name in SEA	Document	Document Name in SEA
Buy Order	GA-BO	Market Transaction	GA-TNS
Buy Order Ack	GA-BOACK	Failed Market Order	GA-FMO
Buy Order Cancel	GA-BOCAN	Payment Order	GA-PAY
Sell Order	GA-SO	Remittance Advice	GA-REM
Sell Order Ack	GA-SOACK	Financial Return Notice	GA-FRN
Sell Order Cancel	GA-SOCAN	Account Summary	GA-ACC

Table 1: Business Documents

5. SEA Design Prototype

5.1 SEA Background

The Simulated Electronic Auction (SEA) project was formed by three independent lines of investigation. Wagenaar (1991) developed the EDIGame to provide a simulation tool for understanding logistical functions within a port scenario. Wrigley (1992, 1995) developed the IOSGame to provide a general gaming environment for building and experimenting with electronic commerce scenarios. Although the IOSGame uses EDI to support bilateral trading partner relationships, neither the EDIGame nor the IOSGame provided automated search or price discovery among players. Lee and Lee (1994) provided trade matching mechanisms for price discovery based on single or multiple product attributes, mainly for commodities. This experience, and some of the software from these three projects, formed the inputs to the SEA. During the early part of 1994 we built the prototype and added multimedia product representation capabilities.

SEA is designed as a technology platform by which electronic markets of *any* product can be built. The features of the electronic auction in SEA hold for other products as well as flowers. As a first application, we chose the Dutch florist chain because cut flowers are challenging products for multimedia representation. Our assumption is that, if shortcomings of multimedia representations are outweighed by economic gains in flower markets, this finding will hold for other goods whose product representation is less challenging.

5.2 Market Scenario

The central feature of the SEA is that it is played within the context of a business scenario. A business scenario is defined as a number of interdependent firms who coordinate their activities to produce and deliver economic goods within a well-defined market. It includes not only different firm types but also many firms of each type. Firms of the same type compete for market share or profits while firms of different types engage in trade either through bilateral contracts or through a computer mediated market mechanism. Figure 2 provides the example market scenario along with the sequenced exchange of documents. A business scenario specifies: (1) the role definitions of each enterprise in the economic sector and (2) their business protocols. A business protocol is the defined sequence of messages and their contents, exchanged among two or more

trading partners such that commerce occurs smoothly. A business protocol must necessarily specify the message type for each business purpose, the anticipated response to each message, and clear rules as to the appropriate action in case of error or message time-out. All trading partners should know the proper sequence of messages for offering to buy or sell from the market, and for completing the trade cycle in terms of goods' delivery and payment.

5.3 Business Documents

Companies use the documents listed in Table 1, and shown in Figure 2, to interact with the SEA market and other trading partners. Each document has its corresponding implementation in X.12 (North American EDI Standard) and EDIFACT (International EDI standard).

5.4 Economic Model

The underlying economic model of the game environment involves two counter rotational flows among firms: product and money, and bi-directional information flows. The GameMaster controls money flowing into the economy and product flowing out of the economy by adjusting consumer demand in retailer markets. The GameMaster also controls product flowing into the economy and money flowing out of the economy by adjusting input supply quantities, prices and elasticity. Each firm receives initial capitalization including goods, cash, technology, land and buildings. Between resource inflow and resource outflow a variable number of processing, or value-added activities can be designed into the scenario. Firms engage in trade and coordinate their activities through the exchange of business documents.

The GameMaster balances the model at the beginning of the game so that each firm has an equal opportunity to make profit. Each firm receives software game tokens by the GameMaster at the start of the game. These digital tokens represent physical assets that include: land, buildings, production technology, and a transport technology, as well as the goods that firms buy and sell. The game software controls token manipulation to ensure that players do not violate physical laws. For example, players cannot modify a 1000 tulips token into 5000 roses, nor arbitrarily duplicate the token without detection. Additionally, each technology token has capacity constraints per unit time to ensure that the technology behaves consistently with the simulated industry sector.

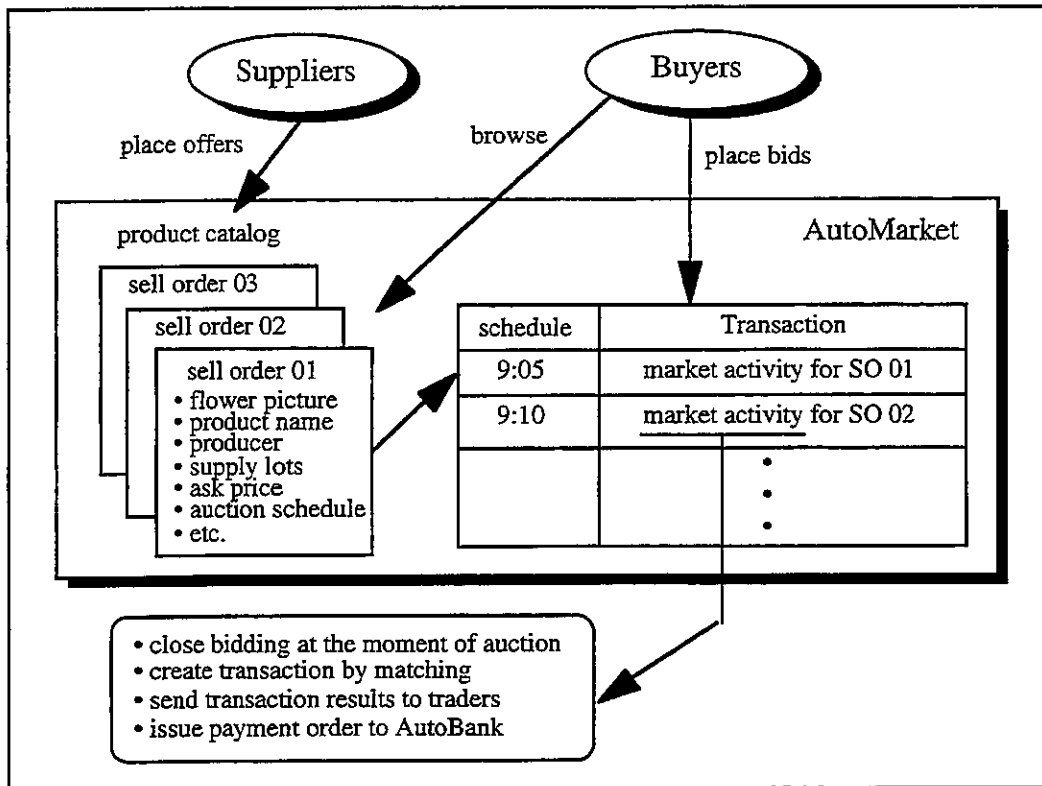


Figure 3: Simulated Electronic Auction

5.5 Player Roles

In the simplest market two roles, sellers and buyers, form the competitive market. Sellers' reservation prices are constrained by production cost and profit margin, while buyers are constrained by a backward sloping demand function of each product in retailer markets.

5.6 Automated Roles

SEA uses three automated roles that operate independent processes on the Internet. These roles run continually allowing 24 hour real time support, although some roles have certain tasks that run on a defined schedule.

AutoMarket: The purpose of the AutoMarket is to act as an intervening market mechanism between players, providing product search, price discovery and contract completion. The automated market accepts sell orders from suppliers and generates sell order acknowledgments with a unique sell order number. These sell orders are made available to buyers through a multimedia browser (see Figure 3). Buyers may select the sort sequence with which to view the current sell orders. For example, buyers may view the market by product type or by supplier ID. The market also accepts buy orders and provides buy order acknowledgments with unique buy order number. The order matching algorithm can be initiated in a variety of ways. Once matches are made, players receive either market transaction details or failed market orders. Successful market transactions also generate payment orders to the Autobank, which forwards remittance advice documents to suppliers.

AutoBank: The AutoBank maintains account balances

and transaction history for all players in the game. Each time one firm wishes to pay another firm, they send a Payment Order to the AutoBank. The AutoBank validates the transaction, debits the payer's account and credits the payee's account, then sends a Remittance Advice to the payee. Payments received in error generate a Financial Return Notice that the AutoBank returns to the payer. Additionally, the AutoBank sends periodic Account Summary statements to all firms. Except for occasional intervention by the GameMaster to handle error conditions, the AutoBank runs in background mode.

EDIMail: The EDIMail program acts as a communication gateway to the Internet by continually polling each user directory for pickup and delivery of new documents. The primary purpose of the EDIMail program is to encapsulate EDI messages inside Email envelopes according to the MIME (Multimedia Internet Mail Extensions) standard.

5.7 Trading Rules

Once products are offered by growers, the AutoMarket assigns auction schedules to lots and constructs a product catalog, which can be accessed by buyers. The product catalog contains product information necessary for buyers to make purchasing decisions, including supply lots, growers, ask prices, auction schedules and pictures of flowers (see Figure 3). Buyers bid on a blind basis for a product entered in the product catalog. All buy orders are valid as long as they are received before the scheduled auction time. At the moment of auction, the AutoMarket closes the bidding for the product and awards the product to the highest bidder,

provided the price was equal or above an ask price (reservation price) set by the producer. Equal bids are sorted on the FIFO (First In First Out) basis.

6. Discussion

With SEA of Flowers, we attempt to demonstrate that the transition from established flower auction markets to a potentially superior electronic auction is feasible. Since the adoption of an electronic marketplace is highly dependent upon the decisions of a group of potential market participants, a business simulation game is proposed as a way to test its feasibility. This paper delineates the first prototype design of SEA. A set of business simulation games will take place with experimental subjects to estimate *a priori* the potential value of the electronic alternative. Detailed and testifiable propositions will be developed based on our analyses.

SEA prototype provides a vehicle for investigating a number of research issues. The issues fall into three general categories: product attributes, trading mechanisms and market structures. The first area for investigation is to examine the conditions under which different types of products may be brought and sold using multimedia. Here the critical issue revolves around product representation quality as compared to standard attributes, which may be conveyed using textual data. The second area involves examination of different market mechanisms including price discovery and multi-attribute matching rules. The performance comparison of different trading rules provides market makers with crucial information. The third area involves experimenting with different market structures. For example, to examine the effects of expanding a market's size, SEA may be used to compare markets characterized by several small independent groups, with a market where all groups are pooled into a single larger market. Additionally, since electronic markets are forming every day, some more successful than others, one critical question is to determine the minimum size, in terms of market participants, necessary to form a stable market. Finally, SEA will allow us to look at transition feasibility from one market regime to another. For example, from a direct search market to an auction market.

SEA also has a significant educational value. Experiences with both the EDIGame and the IOSGame are that players can gain an enormous amount of hands-on experience simply by being placed in a competitive gaming environment. These benefits can be brought into the SEA environment. Electronic commerce concepts, revolving around the communications and document standards, also form part of the learning experience. In addition, SEA adds exposure to market mechanisms and multimedia product representation: both the advantages and shortcomings. For students and practitioners of information systems, response time and bandwidth requirements for different market situations become readily observable.

Acknowledgments

We gratefully acknowledge the technical assistance of Mr. Darryl Candiff from Decision and Information Systems Department at Erasmus University. The discussion with

Mr. van der Poelgeest, a manager at the VBA flower auction market, was extremely helpful in improving our model. Dr. Ronald Lee, in his position as Director of EURIDIS, was of great assistance to us in our work.

References

- Bakos, J. "A Strategic Analysis of Electronic Marketplaces," *MIS Quarterly*, September 1991, pp. 295-310.
- Clemons, E. and Weber, B. "Evaluating the prospects for alternative electronic securities market," *Proceedings of the 12th International Conference on Information Systems*, New York, 1991, pp. 53-61.
- Cohen, K., Maier, S., Schwartz, R. and Whitcomb, D. *The Microstructure of Securities Markets*, Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
- Cohen, K. and Schwartz, R. "An Electronic Call Market: Its Design and Desirability," In Henry C. Lucas, Jr. and Robert A. Schwartz, editors, *Information Technology for the Securities Markets*, Dow Jones-Irwin, Homewood, Illinois, 1989, pp. 15-58.
- Freund, W. "Trading Stock Around the Clock: The Future Growth of Global Electronic Markets," *California Management Review*, Fall 1991, pp. 87-102.
- Garbade, K. *Securities Markets*, McGraw-Hill Book Company, New York, NY, 1982.
- Katz, M. and Shapiro, C. "Network Externalities, Competition and Compatibility," *American Economic Review*, Vol. 75, Spring 1985, pp. 70-83.
- Lee, H. and Lee, R. *Intelligent Electronic Markets for Commodity Auction: Integrated Approach of Economic Theory and Social Choice Theory*, Research Monograph, EURIDIS, 1994.
- Lindsey, D., Cheney, P., Kasper, G. and Ives, B. "TELCOT: An Application of Information Technology for Competitive Advantage in the Cotton Industry," *MIS Quarterly*, Vol. 3, 1987, pp. 141-154.
- Malone, T., Yates, J. and Benjamin, R. "Electronic Markets and Electronic Hierarchies," *Communication of the ACM* (30:6) June 1987, pp. 484-497.
- Neches, A. "FAST - A Research Project in Electronic Commerce," *Electronic Markets Newsletter*, No. 9-10, October 1993, pp. 25-27.
- Smith, V. "Microeconomic Systems as an Experimental Science," *American Economic Review Proceedings*, December 1982, pp. 923-955.
- Stigler, G. "The Economics of Information," *Journal of Political Economy*, Vol. 69, June 1961, pp. 213-225.
- van Heck, E. and Groen, T. "Towards an Electronic Flower Auction," *Proceedings of the First International Conference on Agri-Chainmanagement*, Wageningen Agricultural University, Wageningen, March 1994, pp. 24-25.
- Wagenaar, R. "Management Training on EDI: The Port of Rotterdam Simulation Game," Rotterdam School of Management, Erasmus University, 1991.
- Whitcomb, D. "An International Comparison of Stock Exchange Trading Structures," In Y. Amihud, T. HO, and R. Schwartz, editors, *Market Making and The Changing Structure of the Securities Industry*, Lexington Books, Lexington, MA, 1985, pp. 237-255.

- Wrigley, C. "The IOSGame: Inter-Organizational Simulation Game," *Simulation and Gaming* (forthcoming).
- Wrigley, C. "EDI Transaction Protocols in International Trade," *Proceedings of the 5th International EDI Conference*, Bled, Slovenia, 1992, pp. 405-418.

SYSTEM DEVELOPMENT PRACTICES: A COMPARISON OF THE EXPERIENCE OF AUSTRALIAN AND UNITED STATES' ORGANIZATIONS

Anne Rouse
Department of
Information Systems
Monash University
Melbourne

David Watson
Department of
Marketing
Monash University
Melbourne

Anne Gilbert
Department of
Information Systems
Monash University
Melbourne

Rod Dilnutt
Ernst & Young
Melbourne

Abstract

This paper compares the results of industry surveys of system development practices undertaken in Australia in 1994, and the United States in 1993 as part of an international IS quality benchmarking exercise sponsored by Ernst & Young. The paper reports on the penetration of leading practices and development technologies, and the perceptions of their value. It concludes that while there are many similarities between the two countries, Australian organizations are adopting most practices at a slower rate than US organizations. The exceptions are Business Process Re-engineering, Relational Database and Client Server Architecture which are taking off very quickly in Australia. In both countries practices providing the most value are use of a systems development methodology, structured development methods, cross functional teams, and prototyping. The least valuable are software engineering and software process assessment practices, although the use of these is growing significantly in a subgroup of American organizations, where they are perceived as quite valuable.

Introduction

A number of recent studies of the most important issues facing Information Systems (IS) managers have indicated that the quality and productivity of systems development processes are primary concerns (Broadbent, Butler and Hansell, 1994; Niederman, Brancheau and Wetherbe, 1991). Consequently, an issue of critical interest to both IS practitioners and academics is which of the many competing practices and technologies are most effective in bringing about improvements in the delivery of information systems.

The IS discipline has a history of producing new methods or techniques which are heralded as the solution to ongoing problems and the so-called software crisis (Arnott and O'Donnell, 1994; Brooks, 1987). Many of these IS process innovations have failed to live up to their original promise (Fichman and Kemerer, 1993). Consequently the risk of adopting new practices or technologies is high. Few of the methods offered to industry have ever been empirically validated, and even when methods have been evaluated, the nature of the validation process means that conclusions are often applicable only to

narrowly defined situations (for examples see Verschoor and Low, 1994; Dekleva, 1992; Jenkins, Naumann and Wetherbe, 1984; Vessey and Weber, 1984). There are also many factors working against large scale validation of practices, including the fact that to run projects or activities in parallel so as to vary the method used in a scientific way is rarely possible.

An alternative approach to empirical evaluation is to survey IS managers about their perceptions of which IS practices and technologies are most likely to lead to improved quality and productivity. Further information can be obtained by asking which are the ones respondents themselves have chosen to employ, and what impact these practices have had on their systems development activities. This is the approach adopted in the studies reported in this paper.

Our paper reports on the experiences of senior IS managers in Australia and the United States (US). It is based on a national survey of Australian system development practices conducted by the authors in 1994, and on a corresponding survey conducted a year earlier in the United States (Ernst & Young/SIM, 1994). Both surveys were undertaken as part of an international benchmarking exercise sponsored by Ernst & Young's IT division.

Surprisingly there have been few studies of this nature published in the academic IS literature. One of the problems in getting responses to surveys of this kind is that the IS technologies and practices used by an organization are a very important aspect of its competitive strategy. For this reason, many organizations may not wish details of their practices, or more importantly their *success* with these practices, to be divulged. Consequently although a number of consulting organizations such as the Gartner Group and Butler and Cox do conduct such surveys, their findings are only available to particular subscriber organizations.

The literature does include a number of surveys canvassing IS manager's beliefs about the key strategic issues facing them (for example Watson, 1988; Niederman, Brancheau and Wetherbe, 1991; Badri, 1992; Doukidis, Smithson and Naoum, 1992; Pervan, 1994; Wang and Turban, 1994), but by definition these are predominantly concerned with high level, and long-term issues, and the relationship between the IS division and its parent organization. It is only in passing that they

address what Boynton, Zmud and Jacobs (1994) refer to as the *operational level* of system development practices. Some information about IS practices can be gained from surveys of this kind, though. For example Broadbent, Butler and Hansell (1994) have indicated that client server technology is seen as one of the most important IS technological developments by Australasian IS managers, and that BPR is considered one of the top four important management trends by the same group. In a similar vein, Doukakis et al (1992) reported that only 30% of Greek companies used specific IS development methodologies.

The IS literature also includes a number of investigations of the extent to which individual practices or technologies have been adopted (for examples see Necco, Tsai and Holgeson, 1989; Doke and Myers, 1987) but these don't usually consider the practice or technology in relation to the range of others which compete for IS management's attention.

Of the few more general surveys of industry practices in the academic literature, several which were undertaken in the 1980's are now, because of the rate of change in IS process innovations, becoming dated (for example Kievit and Martin 1989; Necco, Gordon, and Tsai, 1987; Carey and McLeod, 1988; Jenkins, Naumann and Wetherbe, 1984). One recent survey was undertaken though by Jones and Arnett (1993), who administered a questionnaire in 1992 to US systems analysts working in 'complex large-scale systems development projects'. Their survey (with 91 respondents) found that the most predominant tool used by surveyed analysts was still flowcharting. Jones and Arnett also found that only 36% of analysts used structured analytical tools like DFDs extensively, noting that this was still an increase over the findings of the earlier studies by Keivit and Martin, and Carey and McLeod. Another pertinent finding was that in 1992 only 30% of respondents were using CASE tools, despite the promises made for this development technology. Jones and Arnett noted that this was consistent with other studies.

Clearly the diffusion of system development methods and technologies into mainstream practice is an area worth investigating further.

Adoption of New Practices or Technologies

The focus of the studies reported here is the extent to which a range of practices and tools have been adopted in the two countries, and the benefits, if any that are accruing from these. New practices or tools are not always adopted at the same pace, and some promising new tools never move out of a niche market into general practice (Fichman and Kemerer, 1993). A development innovation is always competing with alternative practices and technologies, in the same way that a product competes with alternative products that meet the same needs.

A new IS innovation will normally exhibit the characteristic S curve of the product life cycle, usually referred to as the adoption curve (shown in Figure 1 over page). In the early stages the benefits or values of a practice or technology are still unknown, and there is a high risk involved in adopting it. However early adoption can often reap major strategic benefits, so there are usually pioneering organizations willing to take this risk. As pioneers and early adopters gain experience with a new practice, evidence accumulates about its benefits. Where the benefits are positive, the adoption rate for a successful practice or technology will increase, until it has been adopted by the majority of organizations. It then becomes a predominant practice (or technology), and the rate of adoption slows until all but the most resisted have embraced it. The difficulty for many organizations is to know when the benefits of a new practice are real, and when they are illusory. Arnott and O'Donnell (1994) warn that in the early stages of the IS method life cycle curve there is considerable proselytising and hype, with many promises made that will often never be realized. This was an issue also canvassed by Brooks in his classic 'no silver bullet' paper (Brooks, 1987).

Whether a practice or technology makes it through all the phase of the adoption curve to become institutionalized will depend largely on the extent to which it delivers benefits over and above those delivered by its competitors: known as its *relative advantage* (Utterback, 1994). This will also affect the rate of adoption, demonstrated by the steepness of the adoption curve. But there are other factors besides relative advantage involved in the extent to which a practice or technology is adopted (Rogers, 1983), including its *compatibility*, *complexity*, *trialability* and the extent to which its benefits can be seen (its *observability*). Relative advantage is, however, the most powerful factor, and practices that provide relative advantage, and that have been around a long time will have the greatest penetration amongst organizations. Practices that offer very high levels of relative advantage will also be adopted much more quickly, and hence will show a steeper adoption curve.

Several adoption curves are illustrated in Figure 1 (over page), taken from Everett Rogers' work on diffusion of innovation. From this figure we can see that Innovation I is being adopted at a much faster rate than Innovations II and III.

Survey Method

The Australian study involved a self-report questionnaire developed from the one used in the US study. A pilot survey had been undertaken in 1992 amongst US companies asking them to report on the practices and technologies they considered to be leading ones. "Leading" was defined as those practices or technologies likely to result in higher levels of system quality or

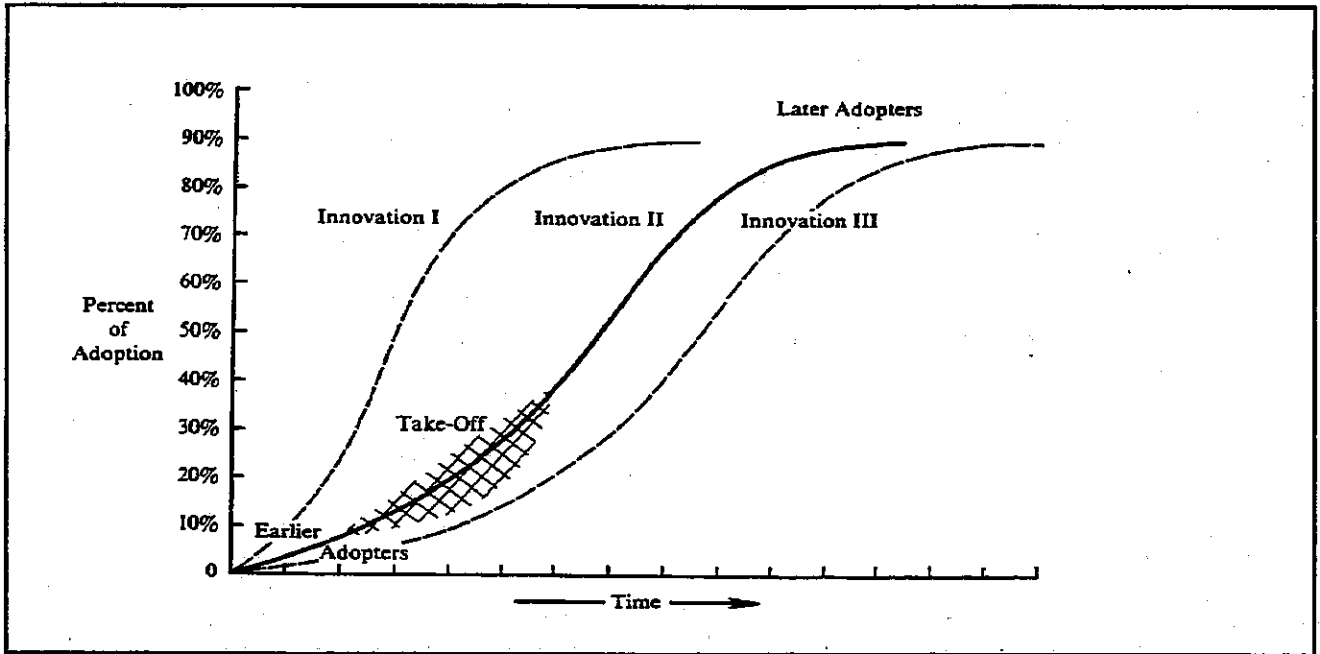


Figure 1: Innovation Adoption Curves (after Rogers, 1983)

productivity, that also had a high level of adoption. This might be adoption within the systems development community generally; within the respondent's industry; or within his or her own organization.

Using this list as a starting point, a survey was developed which asked respondents to provide details about their use of 22 leading practices or technologies, and the significant benefits or improvements, if any, that the practice had provided. Although the survey actually had eighteen questions, the wording of the one most related to the results reported here is shown in Figure 2 below. Respondents were also given the opportunity to add additional practices or technologies not included in the original list. To help increase the reliability of the survey, the questionnaires included a glossary of terms defining what was meant by the various practices or technologies.

Researchers interested in the complete survey can obtain a copy from the authors.

The survey also included questions about the relationship between respondents' Information Systems division or department, and the organization it served, as well as a range of demographic questions. Although the Australian questionnaire was based on the US one, there were some minor changes between the two countries, so results for some questions cannot be reported for the US organizations

Respondents

A total of 76 organizations responded to the surveys, 38 in the US and 34 in Australia. The person completing the survey was normally the most senior IS officer within the organization (ie the Chief Information Officer). In

A number of Systems Delivery practices and tools are listed below (see attached glossary for definitions). For each item, please indicate the length of time it has been in use, the percent of current projects that are using it, and the contribution it has made to the performance improvement goals indicated in the columns to the right. Check the "Value/Improvements" boxes only if the practice has been of significant benefit to this business unit in that particular area.

Practice or Tool	Number of Years In Use	Percentage of Projects Using It	← nature of improvements or benefits → (check those that apply)					
			Fewer Defects	Improved Customer Satisfaction	Reduced Development Time	Reduced Development Cost	Reduced Rework/Waste	Other *see Note
JAD			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Client Server			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
etc			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2: Survey Question on Use of IS Practices and Technologies (Edited)