To appear in *Proceedings of the First International Conference on Telecommunications and Electronic Commerce*, Nashville, TN, USA, November 20-22, 1998

Towards Electronic Commerce Management Systems: Concepts and Architecture

J. Leon Zhao, Hongjun Lu, Vojislav B. Misic
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{zhao, luhj, vmisic}@ust.hk

Abstract

Presently, Internet-based electronic commerce systems are developed through in-house programming and are plagued with uneven quality, low productivity, and little reusability. Furthermore, low interoperability between those systems is also a major obstacle to the deployment of open and flexible electronic markets. We investigate the concept of electronic commerce management systems (ECMS), which can help consolidate the electronic commerce technology by integrating object-oriented architectures, Web information systems, database systems, and workflow systems. In this paper, we develop the main concepts and propose a high level architecture for ECMS.

1. Introduction

Current *electronic commerce (EC)* applications are developed mostly through in-house software development, requiring intensive programming in HTML, CGI, and other script languages such as JavaScript. This invariably means uneven quality, low productivity, and little interoperability in EC systems from different organizations. This situation reminds us of the early days of data processing before the commercialization of database management systems (DBMS). Consequently, much needs to be done to advance the concepts and techniques in EC systems.

This situation has been recognized by many, and research is under way towards component-based EC system development (Bichler, Segev, and Zhao, 1998). New ways are being explored in the industry to reuse software components through techniques such as object frameworks or component-oriented programming. Under a component-based framework, the center of the system is a lightweight kernel to which new features can be added in the form of components. Distributed object standards as well as document-centric EC standards enable easier inter-operability of electronic commerce applications in a heterogeneous environment.

We believe that there is also a need to investigate the feasibility of separating the development of specific EC applications from the creation of generic facilities to support these application development activities. For this purpose, we propose the concept of *electronic commerce management system (ECMS)* in order to advance research and development in this area. We define ECMS as a software system that provides generic facilities to support the efficient development of EC applications. Such an integrated application software development environment will make the development of EC applications easier and faster, much like that DBMS makes the development of database applications more productive.

The creation of ECMS as generic systems to support application development of EC systems is technically necessary because packaging many components into an EC system can be a very complex task that require a five star guru. An ECMS will integrate the basic components into a homogeneous environment that enables end users to develop EC applications without coding. Additional components can be integrated loosely into EC applications, perhaps through some applications programming.

ECMS is also economically feasible. There will be a sufficiently large market to sustain the manufacturing of ECMS once EC markets permeate all sectors of the economy worldwide. There will be a third party market for EC components needed for special purposes, and there will be a market for a general purpose ECMS for developing EC applications.

In this paper, we develop the main concepts underlying ECMS and suggest an ECMS architecture based on proven techniques from database systems, workflow systems, object-oriented programming, the Web technology, and artificial intelligence. Next, we review the relevant concepts in Section 2. Section 3 suggests a conceptual framework that rests on a comparison of three types of activities and three generations of EC development. We then propose a high level architecture for an ECMS in Section 4. Section 5 highlights the main conclusions of the paper and outlines some research issues in ECMS.

2. Preliminary Concepts

In this section, we discuss several concepts that provide the technical basis for ECMS and are needed for the delineation of materials in this paper. These concepts include the electronic market model, the workflow paradigm, intelligent agents, and blackboard architecture. Their relevancy to ECMS is discussed as well.

2.1. The Electronic Market Model

Electronic market can be modeled by classifying commerce activities into several phases. Recently, there has been a trend towards a four-phase model. For instance, Selz and Schubert (1998) refer to the four phases in their model as information, agreement, settlement, and communication. In *the information phase*, customers search for potential sellers, look for the right products, and decide on alternative options. *The agreement phase* contains interactions between the buyer and the seller on the terms of purchase. This phase may involve negotiations between buyers and sellers. In the case of auctions, interactions can occur among buyers. In *the settlement phase*, monetary transactions occur between the buyer and the seller. The communication phase emphasizes the concept of community taking place among customers and between customers and the firm. This fourth phase entails the unique nature of electronic commerce.

Schmid and Lindemann (1998) called the fourth phase after-sales. By after-sales, the authors refer to customer support activities required to deal with product failures and to maintain customer loyalty. To unify the terminology, we refer the fourth phase as *the support phase*, which is a more general term than either after-sales or communication. This four phase EC model is the conceptual basis for analyzing the system requirements of EC and will be used throughout the paper.

2.2. The Workflow Paradigm

Most workflow management systems are based on the four perspectives of business process representation, i.e., functional, behavioral, organizational, and informational (Curtis et al., 1992; Bussler and Jablonski, 1994; Gruhn, 1995; Kwan and Balasubramanian, 1997).

The *functional perspective* indicates that workflow management systems need to specify the tasks and the underlying rationale of a workflow by decomposing high level functions into tasks that can be allocated to human or software agents. The *behavioral perspective* refers to the need for specifying when and how the tasks are performed; these can be specified using process logic in petri nets, or other process models (Kumar and Zhao, 1998). The *organizational perspective* seeks to answer the question of who performs what tasks and with which tools. In workflow management systems, the organizational perspective involves actors, roles, resources, and resource management rules that can be modeled with organization charts and object hierarchies. The *informational perspective* relates to the business data and documents that are the subjects of workflow activities. In workflow management systems, information is usually organized in object hierarchies or networks and stored in databases or file systems.

The essence of the workflow paradigm is (1) to separate roles from workers (agents) and (2) to separate the definition of process logic and execution of processes. By separating roles from agents, the workflow model can be defined using roles and the workers can later be bound to the roles. The result is that generic workflow models can be defined independently of organizations and workers, and that workflow models can be shared among different organizations. By separating the process logic from the execution of specific processes, each specific process can be bound to the process logic as defined in the workflow model and executed individually. These two types of separations enable the development of generic systems for managing workflow and for specifying applications in workflow, resulting in workflow management systems (WFMS).

Electronic commerce activities embed workflow processes, in particular, the production type of workflow[1]. Therefore, the recent conceptual and technical developments in workflow systems can be applied to the development of ECMS.

2.3. Intelligent Agents

In our view, intelligent agents will become an important technique in future EC technologies. Initial experiments have been reported to be successful in mediating consumer-to-consumer markets in a laboratory setting (Moukas, Guttman, and Maes, 1998). New frameworks have been proposed for applying intelligent agents to support electronic commerce (Liang and Huang, 1998).

According to Hedberg (1996), intelligent agents are autonomous software entities that can navigate heterogeneous computing environments and can, either alone or working with other agents, achieve some goal. Thus, they

[1] Other workflow types include administrative, knowledge intensive, ad hoc (Georgakopoulos et. al, 1995; Stohr and Zhao, 1997).

require on-board intelligence to achieve their task, such as planning, reasoning, and learning algorithms. Several examples of agent-based collaborative systems have been reported in the literature such as the followings:

- Edmonds et al. (1994) studied a collaborative design problem involving participants with heterogeneous skills and suggested that multiagent systems can be deployed to support design with a range of functions when considered from a number of perspectives.
- Mahling, Craven, and Croft (1995) presented an overview of their office work systems. The most recent generation, Polyflow, used intelligent agents to support knowledge-based workflow routing and task handling.
- A multiagent architecture was developed in the ADEPT (Advanced Decision Environment for Process Tasks) project (Jennings et al., 1996). In this system, the provision of services by one agency for another is initiated through negotiation of a service agreement. Each agent acting autonomously will constantly assess the situation and decide how to commit the resources of its agency, whether to call for services from other agents, or negotiate for new service agreements.

2.4. Blackboard Architecture

Blackboard architecture is important in the design of ECMS because it has been shown to hold promise for implementing open and flexible intelligent systems. We will investigate the feasibility of using a blackboard architecture as the basic communication structure in ECMS.

Blackboard architecture was first developed as an Artificial Intelligence technique to facilitate control in knowledge based systems (Hayes-Roth, 1985; Engelmore and Morgan, 1989). Recently, it has been proposed as a workflow management approach to support asynchronous cooperation (Yousfi et al., 1994). A blackboard architecture includes three main components: the knowledge source, the data structure, and the control process.

The workflow management system has three boards: board A contains the objects shared by actors and processes, board B contains control mechanisms to ensure the objects in board A are protected and coherent, and board C manages the asynchronous interactions between actors. The blackboard architecture is based on the Speech Act Theory of Austin and Searle (Searle, 1986) and allows message handling, information sharing, and workflow processing among cooperative agents in an open and flexible manner.

3. Technological Foundation of ECMS

3.1. Data Transactions, Business Processes, and Commerce Functions

As discussed previously, numerous components must be included in an EC system to support the four phases of EC activities, namely information, agreement, settlement, and support. As a result, an EC system can be potentially very complex. However, from a system perspective, components in these four phases can be supported by the following three types of activities, i.e., data transactions, workflow processes, and commerce functions:

- *Data transactions* have to do with the storage, retrieval, search, and presentation of EC data elements as in the case of electronic yellow pages, product catalogues, etc. Data management functions for supporting data transactions include logical level functions such as data modeling, data structures, data queries, and physical level functions such as indexing, concurrency control, and crash recovery. These functions have been under continuous development during the last twenty years and have become commercially mature as found in database management systems.

- *Business processes* refer to the collaborative activities among specialized workers and managers that follow certain business protocols and regulations in order to achieve given business goals efficiently and effectively. Typical business processes include production, administrative, knowledge, and ad-hoc (Stohr and Zhao, 1997) that vary in process features such as complexity of the process model, frequency of occurrence of similar processes, and work functions required. Various workflow management systems are now available on the market, each may provide some of the following workflow management functions: process modeling, workflow routing, workflow monitoring, task automation, process memory, communication formalization, and system integration (Zhao et al., 1998).

- *Commerce functions* refer to the core activities that are unique to commercial transactions such as cataloging, brokering, payment, negotiation, auctioning, credit, security, etc. These activities are usually contained in various business processes and required data transactions. Therefore, some of the commerce activities can be supported by the data management functions in database management systems and by the process management functions in workflow management systems. Furthermore, the business processes in electronic commerce tend to be of production type, and data transactions are frequently well defined. Consequently, commerce activities can be supported by a subset of database and process management functions.

Table 1 illustrates the three types of functions, leading to a three-level framework that reflects the current state of software technologies.

Activity Type	System Functions	Commercial Systems
Data Transactions	Data modeling, data structures, data queries, concurrency control, crash recovery, etc.	Database management systems
Business Processes	Process modeling, workflow routing, workflow monitoring, task automation, workflow memory, etc.	Workflow management systems
Commercial Functions	Cataloging, brokering, payment, negotiation, auctioning, credit, security, etc.	Not yet available

Table 1. A three-level framework of technologies.

Generic data activities can be readily supported using database management systems (DBMS), and generic process activities can be supported with workflow management systems (WFMS). However, application development platforms for the commerce activities aforementioned are not yet available in off-the-shelf systems. We believe that ECMS are needed to support EC, which can integrate existing functions in DBMS and WFMS. Different ways of integrating commerce facilities with data and process facilities will lead to three dramatically different approaches of EC application development as we delineate next. In fact, these three approaches represent a natural progression of three generations of EC development methodologies.

3.2. Three Generations of Electronic Commerce Systems

We anticipate that there will be three generations of electronic commerce systems as illustrated in Figure 1 and elaborated next. Figure 1 indicates that the evolution of e-commerce development methods is expected to improve the productivity of applications development and the system interoperability, among other important metrics such as information sharability, system extensibility, and system reliability.

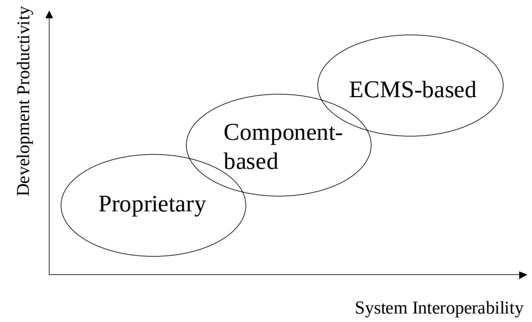

Figure 1. An illustration of three generations of EC technology.

3.2.1 Proprietary EC Development

The first generation of EC development has been done largely through proprietary programming or in-house programming as the corporations are doing it now. Under this approach, the Internet browser serves as the development platform. The programmers write EC systems using the combination of HTML and script languages such as HTX, CGI, IDC, and JavaScript. Under this approach, databases developed on top of a DBMS usually serve as the information repository to store EC data. There are many problems with proprietary EC development, which is expensive and of little or no reuse between systems.

While this first generation of EC development has accumulated valuable experience and achieved industrial momentum, the lack of reusability in current EC applications and the lack of interoperability standards are creating a great barrier to widespread EC application. As a result, the next generation of component-based EC development has been proposed and experimented.

3.2.2 Component-based EC Development

The 2nd generation of EC System development is based on object frameworks for EC applications (Bichler, Segev, and Zhao, 1998). An object framework is a collection of cooperating objects that provide an integrated solution. The basic feature of such an object framework is the bi-directional flow of control between the application and the class library, achieved through dynamic binding. The object framework enables significantly more functionality than a traditional library, and also provides large-scale reuse.

Another bedrock of component-based approach is distributed object standards such as OMG's CORBA or Microsoft's DCOM. The Interface Definition Language (IDL) and the Application Programming Interface (API) by Object Management Group (OMG) enable client/server object identification within a specific implementation of an Object Request Broker. The OMG's Electronic Commerce Domain Task Force (ECDTF) defined a reference model including ten facilities needed in EC applications, such as object browser, catalogue, brokerage, agency, service, contract, payment (OMG/CommerceNet, 1997).

At the core of this component-based approach lies a system kernel that contains the basic functions needed by most EC applications such as the ECDTF facilities. Various other more specialized components can be requested from the class library and plugged into the system as dictated by the EC system developer. As said previously, this approach starts at a higher level and eliminates a lot of lower level programming due to better component reuse.

Component-based EC development is a great step forward towards better productivity, reusability, interoperability, and efficiency in EC applications as compared with the first generation of EC development that mainly relies on proprietary programming. It is expected that this second generation of EC development will become popular as the standard class libraries and EC architectures become widely available. The component-based approach will lead to the third generation of EC development that emphasizes the separation of EC applications and EC development facilities.

3.2.3 ECMS-based EC Development

Component-based approach entails a loose integration between the EC system kernel and various EC components. Much customization of system architectures and components interfaces may still be needed during application development. Based on past experience with extensible DBMS such as the contrast between Postgres and Exodus approaches, a loose integration approach in Exodus requires five star gurus to complete the integration and to tune the system; conversely, Postgres took a tighter integration approach and therefore is easier to use.

ECMS as previously discussed in this paper will be the third generation of EC development technology, and contains various generic EC functions to serve the needs of customers and sellers. The main difference between the component-based approach and ECMS approach is that ECMS integrates the EC facilities tightly and provides development facilities on top of the EC facilities indicated in the ECDTF reference model. Such development facilities include Policy manager, Process modeler, Application designer, etc. as discussed in more details next.

EC applications can be developed on top of an ECMS using the facilities provided in the system. The critical difference between an ECMS and a component-based approach is that an ECMS contains the basic components needed for developing EC applications and the user needs not load components from class libraries. Further, the paradigms commonly seen in DBMS and WFMS will be utilized in ECMS in a unified framework.

4. Architectural Issues for ECMS

In this section, the basic architecture of an open ECMS is described. Note that we focus on the business-level functions in this paper. Lower-level functions such as networking facilities and security functions, although very important, are not discussed here since they are the underlying system infrastructure.

4.1. A Workflow Analysis of EC Applications

Let's take a look at how workflow analysis can be applied to EC applications. To start, we analyze how a customer buys something at a department store in a shopping mall, assuming that a fixed pricing policy is in effect. We then discuss the similar process of purchasing at an electronic store on the Internet.

In general, a customer first browses the shelf or searches the electronic product catalogue to find desired merchandises. The customer may consult the store employee. Once the customer decides to buy something, s/he would pick up the goods and take them to the cashier to pay either with cash or credit card. After payment, the purchase is complete.

What happens when the same purchasing process is done on the Internet? The customer would go through a similar process for the purchase. However, a few things are different in an electronic store:

- The physical store shelves will be replaced by electronic catalogues

- The real products will be replaced by pictures, videos, and animations
- The human salespersons will be replaced with software agents for consultation, payment handling, and product delivery
- The shopping cart will be replaced by a virtual basket
- The purchase can not be paid by cash, but with credit card or electronic money.
- In a workflow model, the customer is a human agent. It interacts in the electronic shopping process with a search agent, a consulting agent, a basket agent, a payment agent, and a delivery agent to complete the purchase as shown in Figure 2.

In Figure 2, the customer interacts with various agents in the shopping process for finding and selecting merchandise, making decisions, and paying for the purchase. Note that there are also some interactions among the software agents to complete the purchase.

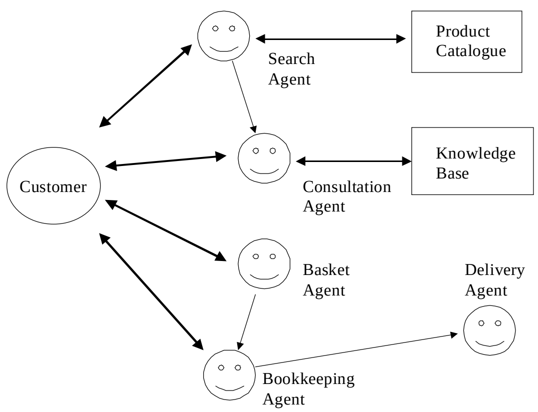

Figure 2. An agent-based eletronic shopping process.

This shopping example is rather primitive; however, it illustrates the basic requirements of electronic commerce. Real world EC applications are of many different kinds and can be much more complex than this example. The complexity may arise in terms of pricing structure and purchasing process. For instance, instead of fixed prices, negotiation may occur between the buyer and the seller. There may be many buyers for the same items such that a bidding process may be needed. Furthermore, the pricing system can be more complex in user authentication, negotiation, and auction methods.

4.2. A System Architecture for ECMS

Figure 3 illustrates a potential architecture for an open ECMS. In this architecture, there are three types of user, the end user, the system administrator, and the application developer. The end user is usually a customer, the system administrator is an EC manager, and the application developer is a programmer responsible for developing applications and integrating system resources. This architecture is based on the client/server model where the client contains facilities for supporting the customer's purchasing activities, and the server contains facilities that support the design and execution of EC processes.

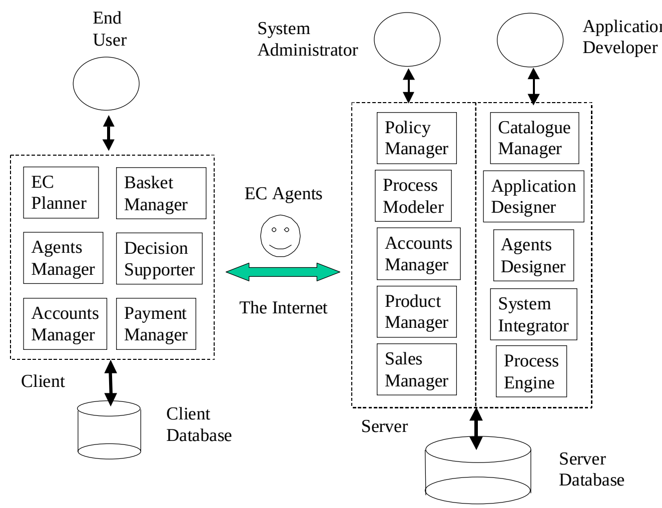

Figure 3. The architecture of an open ECMS.

The major end user[2] facilities include:

- The EC planner helps the customer plan and record buying transactions.
- The agents manager enables the customer to hire software agents to oversee routine purchasing activities and to trigger planned actions. It should include tools for the customer to customize the agents to suit any

[2] Note that another type of end user is the worker of the EC system that is not discussed in this paper.

special needs.

- The basket manager maintains the items of each purchase, both past and present.
- The decision support component helps the customer make purchasing decisions. This tool is especially useful in the case of complex pricing policies such as those require negotiation and bidding.
- The accounts manager enables the customer to manage various EC accounts with different suppliers.
- The payment manager helps the customer keep track of payments made to the sellers.

Some of the facilities for the system administrator include:

- The policy manager allows the administrator to maintain policies, guidelines, and constraints related to the EC activities.
- The process modeler enables the administrator to model workflow processes on all four dimensions, namely roles, routes, rules, and organizations.
- The accounts manager helps the administrator create and maintain customer accounts.
- The product manager for the purpose of maintaining information on merchandise.
- The sales manager provides tools for the administrator to manage sales activities.

The facilities for the application developer include:

- The catalogue manager enables the maintenance of product advertisement.
- The application designer allows the developer to develop EC applications such as payment and messaging tools.
- The agents designer helps the developer create and modify EC agents and make them available to the ECMS.
- The system integrator can facilitate the integration of the ECMS and other system resources such as other Web systems.
- The process engine manages the enactment of workflow processes as defined by the system administrator.

In some respects, an EC system is a workflow system that requires special commerce functions and could be developed using the paradigm of component-based workflow systems. In this paradigm:

- All EC participants are treated as human agents, each of whom assumes some given roles.
- The various EC facilities are treated as software agents that interact with one another and with human agents as well.
- Some of the software agents can be treated as intelligent agents that can perform special assignments based on the instructions of their owners.
- The intelligent agents can communicate with one another using a standard agent language, a part of which can be spoken by human agents as well.
- The intelligent agents can also serve as intermediaries between human agents and information resources such as databases, data warehouses, and yellow pages.
- All software agents and information resources can be considered as EC components.
- Under this environment, EC components can be added into, removed from, or modified in the EC system without interrupting on-going services.
- Various EC components will be purchased from the software vendors for application design, customization, and implementation.

4.3. Main Component Types in ECMS

Several key component types are included in the ECMS, including blackboards, information repositories, software agents, and human agents. The blackboard technique as discussed in the literature review section is very important to support the open and flexible architecture of the ECMS. It allows the exchange of information among human and software agents in an asynchronous manner. The main advantage of the blackboard technique is that integrating new users and agents into the system is straightforward. For instance, new users can be admitted into the system so long as they satisfy some preliminary conditions such as credit rating and identity check.

Information repositories are a major type of components because the ECMS is to process EC related information. The ECMS must be able to communicate with the information repositories registered with it. Another main point is that some of the information repositories may be foreign; that is, the ECMS has no control over them with respect to their management operations and policies.

Human agents include customers, workers, system administrator, and application developers. Each of these human agents uses the ECMS for some particular purposes and is given different sets of tools as discussed in the architecture.

There are many types of software agents within the ECMS that assist the operations of customers and workers. The design and maintenance of software agents can be done by application developers and end users (customers and workers). The types of software agents may vary in different ECMS, but the basic types can be found in (Liang and Huang, 1998).

It is therefore desirable to have an ECMS, which functions as a generic development platform similar to the case of DBMS and WFMS for developing database and workflow applications. With ECMS, we will be able to separate the development of ECMS and that of EC applications. This separation will also bring in the separation of ECMS manufacturers, EC application developers, and EC users.

While we believe that the development of ECMS is feasible, we also think that the nature of ECMS should be different from that of DBMS and WFMS because of the following characteristics of EC:

- Any EC system must be inter-organizational since EC invariably occurs between customers and a seller organization such as in electronic shopping, among many customers such as in used car sales, or among several corporations such as in EDI.
- Any EC system must be globally distributed since the inter-organizational characteristic determines that all parties of EC are rarely located in the same place. That is why Internet has become the uniquely appropriate media for EC.
- The EC processes are very heterogeneous due to variations in pricing and other business structures; on the other hand, database processing in DBMS is relatively homogeneous across organizational boundaries in terms of the data manipulation processes involved.
- The EC system must be a fully open system in order to accommodate new customers and organizations. This is because EC involves market processes and market efficiency requires simple and low cost protocols for incorporating new buyers and sellers.

The logical consequence of these characteristics of EC is that the ECMS must be a system more open than either DBMS or WFMS. This includes the possibility of incorporating different pricing systems from fixed pricing to discounted prices and to negotiated prices, different buying processes such as various auctioning types. In order to increase the economy of scale, ECMS developers should allow the option of bundling different EC components in their ECMS. In other words, the ECMS will be component-based, flexible, and open.

5. Concluding Remarks

In this paper, we argued that there is a need for electronic commerce management systems (ECMS). We conceptualized an ECMS framework that consists of (1) three levels of activities, namely data transactions, business processes, and commerce functions and (2) three generations of EC development approaches, namely, the proprietary, the component-based, and the ECMS-based.

We further envisaged an ECMS architecture that consists of various human and software agents and information resources. ECMS can benefit from the application of intelligent agent techniques in workflow systems since EC requires the support of workflow. However, EC applications usually contain workflows with a larger scope than conventional workflow applications since the former typically involve many organizations and individuals in market processes.

ECMS-based EC development will grow out of component-based EC development. As the industry becomes more mature and the component-based architecture becomes more standardized, ECMS will be the next step in the software technology for electronic commerce. However, academic research should take a lead in the research and development for EC development. In this regard, the topics discussed in this paper are timely. While our study is at the very early stage, this paper has laid a conceptual basis for further and more in-depth investigations. We hope that what is described in this paper can stimulate debates and follow-up studies towards the design and implementation of ECMS.

Many research issues must be resolved concerning the conceptual and execution modeling facilities for electronic commerce. Our hypothesis is that existing modeling languages in database and workflow management systems may prove to be inadequate for modeling electronic commerce applications, and new modeling paradigms might be needed for this purpose. Future ECMS must provide these facilities for the development and enactment of EC applications. We are investigating these issues both from a theoretical perspective and from an engineering perspective.

References

1. Bichler, M., Segev, A. and Zhao, J. L., “Component-Based E-Commerce: Assessment of Current Practices and Future Directions”, *ACM Sigmod Record: Special Section on Electronic Commerce,* December 1998 (forthcoming).
2. Bussler, C. and Jablonski, S. "An approach to integrate workflow modeling and organization modeling in an enterprise." *Proceedings of the Third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (1994) p81-95.
3. Curtis, B., Kellner, M.I. and Over, J. "Process modeling." *Communications of the ACM*, vol.35, no.9 (Sept. 1992) p75-90.
4. Edmonds, E.A., et al., "Support for collaborative design: agents and emergence", *Communications of the ACM*, vol.37, no.7 (July 1994) p41-7.
5. Engelmore, R. and Morgan, T. (eds). *Blackboard Systems*. Addison-Wesley Publishing, 1989.
6. Georgakopoulos, D., Hornick, M., and Sheth, A. "An overview of workflow management: from process modelling to workflow automation infrastructure", *Distributed and Parallel Databases*, Vol. 3, No. 2, pp. 119-153, 1995.
7. Gruhn, V. "Business process modeling and workflow management." *International Journal of Cooperative Information Systems* vol.4, no.2-3 (June-Sept. 1995) p145-64.
8. Hayes-Roth, B. “A blackboard architecture for control”, *Artificial Intelligence*, Vol. 26, 1985, p 252-321.
9. Hedberg, S. "Agents for sale: first wave of intelligent agents go commercial", *IEEE Expert* vol.11, no.6 (Dec. 1996), p16-19.
10. Jennings, N.R. et al., "Agent-based business process management". *International Journal of Cooperative Information Systems* vol.5, no.2-3 (June-Sept. 1996) p105-30.
11. Kwan, M.M. and Balasubramanian, P.R., "Dynamic workflow management: a framework for modeling workflows." *Proceedings of HICSS97.*
12. Kumar, A. and Zhao, J. L., “Dynamic Routing and Operational Controls in Workflow Management Systems”, *Management Science* (Accepted, 1998).
13. Liang, T.-P. And Huang, J.-S. “A framework for applying intelligent agents to support electronic commerce”, *International Conference on Electronic Commerce ’98*, April 6-9, 1998, Seoul, Korea.
14. Mahling, D.E., Craven, N., and Croft, W.B. "From office automation to intelligent workflow systems", *IEEE Expert*, vol.10, no.3 (1995) p41-7.
15. Moukas, A., Guttman, R., and Maes, P. “Agent-mediated electronic commerce: an MIT media la boratory perspective”, *International Conference on Electronic Commerce ’98*, April 6-9, 1998, Seoul, Korea.
16. OMG/CommerceNet, *The OMG/CommerceNet Joint Electronic Commerce White Paper*, Stephen McConnell (Ed.), July 27, 1997.
17. Schmid, B.F. and Lindemann, M.A. "Elements of a reference model for electronic markets." *Proceedings of the Thirty-First Hawaii International Conference on System Sciences* (1998) p193-201 vol.4.
18. Searle, J. B. *Expression and meaning - study in the Theory of Speech Acts*. Cambridge University Press, New York, 1986.
19. Selz, D. and Schubert, P. "Web assessment-a model for the evaluation and the assessment of successful electronic commerce applications." *Proceedings of the Thirty-First Hawaii International Conference on System Sciences* (1998) p222-31 vol.4.
20. Stohr, E. A. and Zhao, J. L. “A technology adaptation model for business process automation”, *Proceedings of the 30th Hawaii International Conference on System Sciences*, January 1997.
21. Yousfi, F., Geib, J.M., Beuscart, R., and Hamza, M.H., “A blackboard-based architecture for managing workflow in health care domain”, *Proceedings of IASTED/ISMM Symposium*. Applied Informatics. Annecy, France, 18-20 May 1994.
22. Zhao, J. L., Ho, J. K. K., Sheng, O. R. L. and Tam, K. Y., “Supporting Global Operations via Internet Workflow Technologies”, *Proceedings of the 1998 Asia Pacific Decision Sciences Institute Conference,* Taipei, Taiwan, June 7-11, 1998.