Reference Models for Electronic Commerce

Vojislav B. Misic and J. Leon Zhao
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
phone +852 2358 7632
fax +852 2358 2421
{vmisic,zhao}@ust.hk

Abstract

A number of reference models have been proposed to facilitate the development of electronic commerce systems and applications. A comparative analysis of existing models should be the first step in selecting the right foundation for the system being developed. This paper presents some results of an analysis of four among the well known reference models for electronic commerce, primarily from the viewpoint of their suitability to support development of flexible and interoperable electronic commerce applications.

Keywords

electronic commerce, reference modeling, conceptual model quality

Reference Models for Electronic Commerce

Abstract

A number of reference models have been proposed to facilitate the development of electronic commerce systems and applications. A comparative analysis of existing models should be the first step in selecting the right foundation for the system being developed. This paper presents some results of an analysis of four among the well known reference models for electronic commerce, primarily from the viewpoint of their suitability to support development of flexible and interoperable electronic commerce applications.

1 Introduction

Electronic commerce may be succinctly defined as the ability to perform exchanges of goods, services, content, assets, and money, using electronic tools and techniques. Although it promises the benefits of cost reduction, new ways of accessing the customers, and the ability to overcome geographical distance and other physical obstacles, the development and deployment of commercial electronic commerce systems is not without its problems, with the lack of a widely accepted conceptual foundation, in the form of a reference model. A reference model provides a conceptual framework that should facilitate the creation of domain-specific application models, or descriptions of specific electronic commerce application domains [11]. Such a model should, in turn, provide a foundation for the development of flexible, interoperable, and coherent electronic commerce applications. In other words, a reference model should facilitate application development in a top-down approach. Moreover, the existing electronic commerce solutions (most of which are proprietary and incompatible with one another), could also be assessed in the context of such a framework.

Although a number of such models have been proposed by individual companies and organizations and industrial consortia, they offer different perspectives and different sets of features, and selecting the most appropriate one within a given environment is not a trivial task. In particular, little has been done to analyze and compare these models in order to abstract their basic concepts, and arrive at some generalization that could serve as a common basis for further development. To the best of our knowledge, there have been only two attempts in that direction [1, 2]; in both cases, only a brief overview of the key concepts and underlying ideas has been given for each of the models mentioned, and no in-depth analysis or direct comparison of the models has been reported.

Consequently, a need exists for a detailed analysis and comparison of electronic commerce reference models and their pertinent properties. While recognizing that electronic commerce is definitely an area of 'applied technology', this paper concentrates on the system and, to some extent, technology-related issues, rather than the business-related ones. In this way, the analysis presented here may serve as the basis for improvements in some of the existing models, for merging of models in order to combine their individual strengths, or for creation of new and, hopefully, more useful reference models.

The paper has two major parts. First, we review some of the reference models for electronic commerce, and try to present their most important characteristics. Then, we try to assess their properties in a comparative framework, whilst paying special attention to those properties which will enable the model to fulfill its basic purpose. Finally, we present some directions for further research.

2 An Overview of Existing Models

As mentioned before, a number of models related to electronic commerce have been proposed so far, yet only a few possess sufficient potential for further development. Although the first notion of a reference model for electronic commerce dates back several years ago [6], most efforts were focused on specific components and/or services only (such as trading or payment [8,otp97bd,otp97ts}), and only recently comprehensive reference models have begun to emerge. The analysis presented in this paper concentrates on the following models:

- CommerceNet's eCo System [14],
- EBES/EWOS Building Blocks for Electronic Commerce [1],
- Java Electronic Commerce Framework [4]
- OMG Reference Model for electronic commerce [7], and
- Secure Electronic Market for Europe [12],

which constitute a sufficiently representative, albeit necessarily limited sample.

2.1 The eCo System Framework

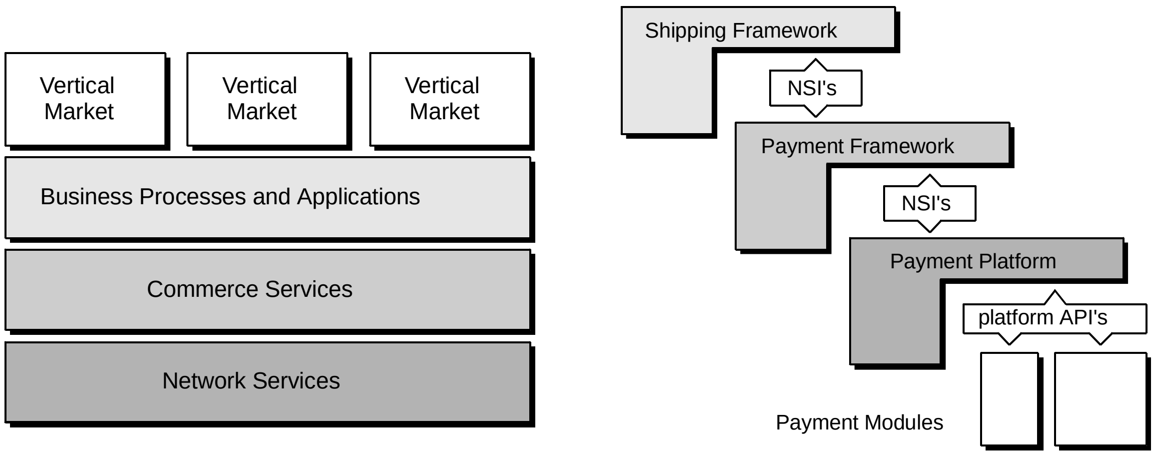
The eCo System is an object-oriented architectural framework for Internet commerce that promotes interoperability and reuse of applications and services, developed by CommerceNet [14]. The eCo project is focused on de facto interoperation, rather than on standards, in recognition of the fact that 'the IT industry is moving so fast that there's seldom time even for de facto standards to emerge' [14]. In other words, individual products, both those already in the marketplace and those yet to appear, should be able to communicate and cooperate in a seamless fashion.

In terms of architecture, the eCo System is intended to be a framework of frameworks, each of which models key business processes and services for building Internet markets. (The term 'framework' is used here to denote an almost complete application that can be customized or extended to address specific needs.) Different frameworks can be built on top of each other, and a shared services infrastructure is thus made available to all applications. The eCo system uses a layered structure with four general categories:

- Vertical markets layer contains services specific to particular Internet markets.
- Business Processes and Applications layer contains generic business services common to multiple Internet markets.

- Commerce Services allow individuals and companies to authenticate themselves and their partners, make payments, collaborate, and participate in Internet markets in any other way.
- Network Services layer contains services that enhance the performance, reliability, and security of the net, in order to accommodate different business needs.

The architecture of the Eco framework is shown in Fig. 1 (a).



(a) (b)

Figure 1. **The eCo System: a 'framework of frameworks' (a) and the hierarchical relationship of frameworks (b).**

Each of the eCo System 'frameworks' specifies the core services that all application objects from that layer must provide, a set of messages for requesting the core services, known as the network services interface (NSI), the business objects on which the services operate, and the application programming interface (API) for any software modules involved in delivering these services. The NSI messages, together with business objects and product taxonomies, constitute the Common Business Language (CBL), intended to be an alternative to the *ad hoc* text strings currently used in EDI systems.

Every application developed in the eCo System is represented with a set of network-accessible services. Commercial products are expected to provide the minimal set of generic services specified in the appropriate 'framework', although additional services can be provided as well.

Frameworks that define API's (also known as platforms) function as middleware, transforming standalone and legacy applications into eCo services accessed through standard NS requests. In this manner, many, if not all, of the solutions that already exist on the market can be integrated in the eCo System architecture.

An example of the hierarchical relationship of eCo 'frameworks', showing the respective role of NSI's and API's, is depicted in Fig. 1 (b).

The eCo System documentation also describes a preliminary architectural design that implements the eCo System framework, and outlines further steps in the development of open interoperable electronic commerce applications [14].

2.2 EBES/EWOS Building Blocks for Electronic Commerce

The project on Building Blocks for Electronic Commerce was sponsored by the European Commission through EBES (European Board for EDI/EC Standardisation) and EWOS (European Workshop for Open Systems); both bodies are now superceded by the European Committee for Standardisation (CEN).

The concept of building blocks is developed as an analytical tool to identify the key technical components for electronic commerce and to position the example products and services currently on offer or planned [1]. The basic analysis technique for identification of such components is decomposition similar to the one used in Structured Analysis \cite{demarco}. Commerce activities are partitioned into five steps, or high-level commercial processes, of Marketing, Contracting, Logistics, Settlement, and Interface with Administrations. These processes are, in turn, partitioned into lower-level activities denoted as sub-processes, such as ‘consult product catalogues’, ‘request price quotation’, and the like.

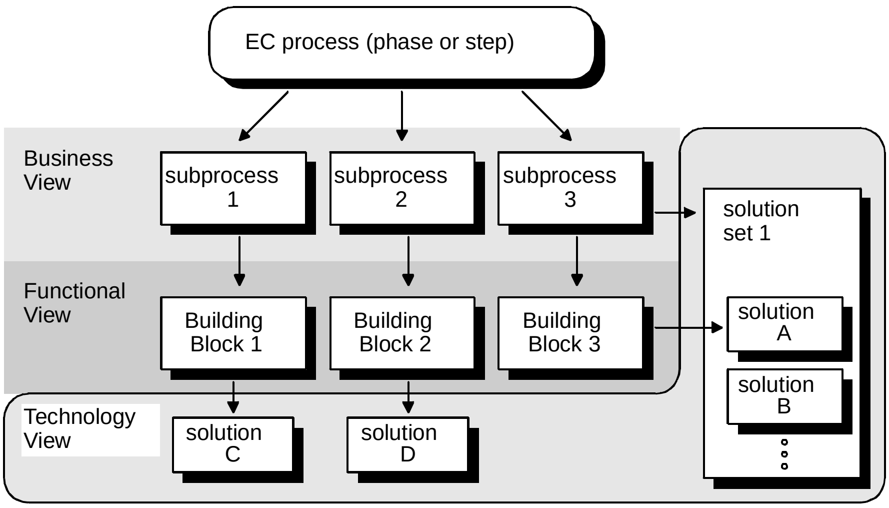


Figure 2. The EBES/EWOS Building Blocks architecture.

Further decomposition is not performed, as it would lead to small activity units which can not readily identified as commerce-specific, and therefore not very helpful in the context of the project. The two-level hierarchical decomposition is based on current electronic equivalents of traditional commerce activities, but it is believed to be universally applicable to any future electronic commerce environment.

The implementation of a building block is called a ‘solution’, which is to be integrated into appropriate applications or services. A set of integrated products or services used to provide support for one or several subprocesses is called a ‘solution set’.

The views and key concepts in the EBES/EWOS Building Blocks architecture are shown in Fig. 2. The topmost, business view of the system contains different commerce sub-processes, whereas the functional view contains the building blocks themselves. Finally, the so-called technology view of the system contains both the solutions and solution sets. Although the original documentation shows this view at the bottom, as if it is the most detailed level of abstraction within the model, we felt that it actually belongs to a different perspective of the model, which is why it is shown both below and to the side of two other views. A number of current and future service building blocks were identified, although no claim was made as to the completeness of the list.

Building blocks identified and defined in this manner can be classified into three basic groups. A local block belongs to the buyer or seller, and interfaces with another local block typically belonging to a trading partner. An access block is used to access a commercial service through a service block. Finally, a service block can only belong to a commercial service (provided by a third party); it interfaces with an access block from the seller or buyer, or from another commercial service. An example model (with two trading partners and two third parties providing additional commercial services) is shown in Fig. 3.

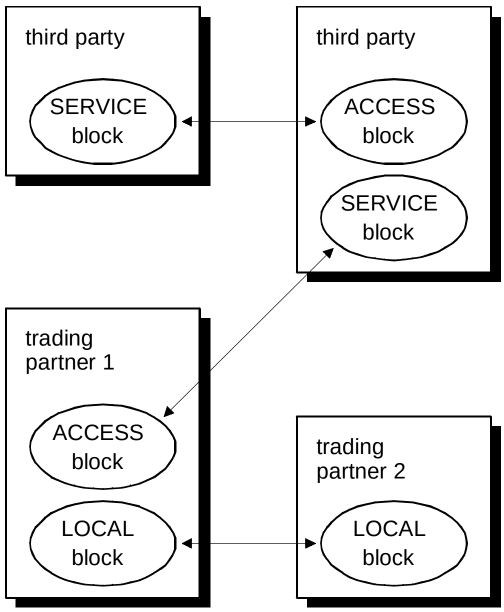


Figure 3. An example of building block relationships in the EBES/EWOS architecture.

Furthermore, the report provides a number of specific recommendations on policy setting and specific technical areas for action in Electronic Commerce standardisation. It seems that the analysis, rather than design, was the main goal of the project, and that can be clearly seen in most parts of the report.

It should be noted that the EBES/EWOS report contains a comparison of the Building Blocks approach with some other models, including some component standards that are beyond the scope of this paper [1].

2.3 Java Electronic Commerce Framework

Java Electronic Commerce Framework (JECF) is an extendible framework for conducting consumer-to-business transactions over the Internet or within corporate intranets [4] Its initial component is the Java Wallet, a client-side application to be distributed as a core component of the Java environment [5]. The Java Wallet will enable the users of any Java-enabled web browser to conduct commerce transactions with JECF-compliant merchant pages anywhere on the Internet.

The JECF consists of the following components:

- A database, to keep a permanent record of user's data (such as personal information, payment instruments, and transaction history).
- A number of payment and service cassettes – modules, to implement various payment protocols and value-added services (e.g., budgeting, financial analysis, and tax preparation), respectively.
- A number of administrative interfaces, used to maintain basic user information and customize certain aspects of the JECF's behavior and appearance.
- Additionally, standard applets are used to implement short-term customer relationships, such as shopping cart and content charging.

Both cassettes and applets may be downloaded from the merchant or financial institution into the consumer computer in order to enable the partners to execute business transactions. Unlike applets, a cassette and its constituent JavaBeans, once installed, are persistent on the client. In this manner, a 'customized' electronic commerce layer may be gradually built for a specific consumer. The JECF architecture is depicted in Fig. 4.

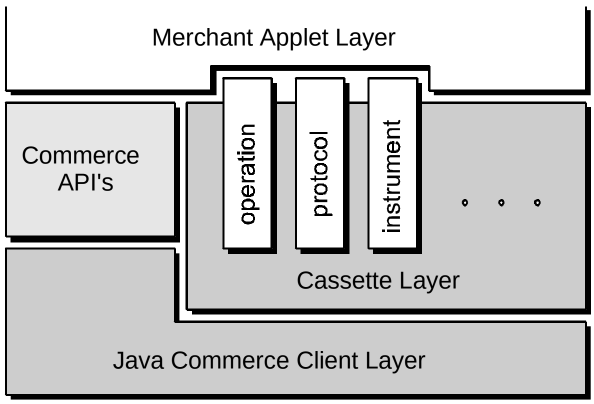


Figure 4. The Java Electronic Commerce Framework architecture.

Security is supported through the facilities already available in the Java kernel, as well as through a number of additional mechanisms.

The Java Commerce Package Layer implements the infrastructure needed by the merchant and the cassette layers. Features at this layer include a user interface, an application model, a data base, and access to strong cryptography. The underlying browser or operating system provides services that constitute the Java Environment Layer.

2.4 OMG Electronic Commerce Reference Model

The OMG (Object Management Group) Reference Model for electronic commerce [7, 9] has been developed by the OMG Electronic Commerce Domain Task Force (EC-DTF). It provides a high-level object-oriented framework for specification of requirements for electronic commerce systems, designed in accordance with OMG's Object Management Architecture – a generic set of components, interfaces, and protocols for distributed object-oriented applications [13].

The OMG architecture is broadly based on the eCo System framework, and it provides a similar multi-layer interoperability framework. Related functional requirements are grouped into a number of containers called facilities. Facilities are, in turn, categorized into market infrastructure services (catalogues, brokerage, and agencies), commerce facilities (such as contract, service management and related desktop facilities, and management and administration of intellectual property rights – IPR), and low level services (these include payment, semantic data facility, profile management, and selection/negotiation). A schematic view of the facilities and their respective categories is shown in Fig. 5. In addition, the object browser and navigator facility (available to all facilities, regardless of the layer they belong to) introduces an extendible framework for the inspection, presentation and execution of electronic commerce entities.

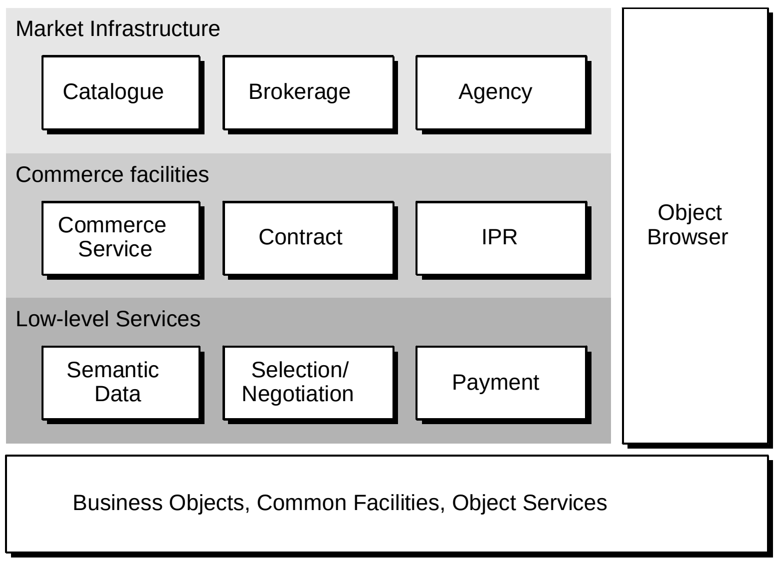


Figure 5. OMG Electronic Commerce Reference Model – principal facilities.

Since the model is based on an object-oriented systems architecture, each facility is handled as a real object offering interfaces to other objects. Detailed semantics for the facilities interfaces are provided, including (in some cases) high-level protocols of their usage [9].

In this manner, the reference model serves as a high level framework, allowing individual components to be developed on a per-market basis, and 'plugged in' in accordance with the requirements of the market. As in the eCo system framework, the emphasis is on interoperability of different solutions, instead of imposing a single unified solution—be it a protocol, payment method, product dictionary, or anything else—to all market participants. (Given the number of systems already present on the market, this would be an impossible task anyway.)

OMG has been active in promoting the framework: detailed CORBA IDL specifications have already been defined for interfaces of some of the facilities, and work is under way for defining others.

2.5 Secure Electronic Market Place for Europe (SEMPER)

Secure Electronic Marketplace for Europe (SEMPER) is a project sponsored by the European Commission, with the ultimate goal of developing and providing an open and comprehensive solution for secure commerce over the Internet and other open networks [12]. The first phase concentrated on the development of a framework and architecture for secure electronic commerce, and the provision of fundamental electronic commerce services (such as offering, ordering, payment and delivery for information services) within this framework. The subsequent phases concentrate on extending the architecture and developing more advanced services.

The SEMPER model of electronic commerce is based on the unified concept of ‘business items’: payments, credentials, and documents or statements. Furthermore, it is assumed that any business scenario consists of a number of standard business processes, which may be further decomposed into a sequence of unidirectional and/or bidirectional exchanges of business items. (SEMPER documentation uses the terms ‘transfers’ and ‘fair exchanges’, respectively.)

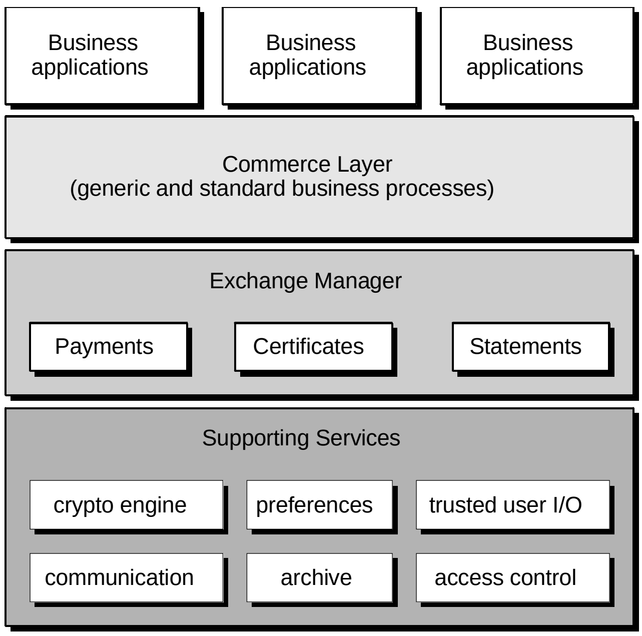


Figure 6. The SEMPER architecture.

These assumptions lead to the architecture shown in Fig. 6. Front-end business applications are built on top of the Commerce Layer containing services that directly implement business scenarios such as ‘mail-order retail’ or ‘on-line purchase’. Since the service set(s) cannot be fixed in advance, provisions are made for secure downloading of additional services to extend the functionality of the existing applications. (Entire applications can be downloaded, thus allowing customers to participate in business scenarios they never encountered before.)

Business scenarios are based on a sequence of exchanges of items of different types: payments, credentials, and/or documents, each of which is managed by a separate service in the Exchange Management Layer. Thus (multiple) existing implementations can be integrated into a unified service framework. For example, the payment manager can provide generic services for handling account- and cash-style payments, together with the negotiation of the means of payment. In this way, different payment systems may be simultaneously installed and any of them can be used in an actual transaction, while the appropriate negotiation may be entirely transparent to user.

Exchange managers use a set of low-level basic services provided by the Supporting Services Layer: communication, cryptography services, user profile management, access control, and others. Different services are specified in the SEMPER architecture, although the services dealing with security requirements are given a prominent role. Again, future extensions are deemed inevitable, hence the standardization activity has focused on the interfaces between individual service blocks, rather than on the details of the blocks themselves.

The SEMPER architecture is deliberately kept as open and extendible as possible, which is why its specification is independent of any particular distributed systems architecture. However, most of the pilot implementations use the Java language.

3 Comparison of Models

The models presented in the previous section will now be compared, mostly on the basis of their suitability for practical implementation, because a reference model has a definite practical purpose.

Among the models reviewed, JECF is the most technology dependent one, since it requires the Java computing environment in order to function properly. The reference framework, then, serves to define the operational process and the facilities for electronic commerce systems within the Java environment. The OMG model exhibits a certain technology bias as well, due to the use of CORBA as the underlying standard for distributed computing. Yet this bias is much less pronounced than in JECF, since CORBA is actually an interoperability framework, rather than a specific technology, and it can be used to integrate systems that use different technologies (Java included). Moreover, the choice of CORBA is somewhat arbitrary, since the OMG model itself does not prohibit the use of some other, functionally equivalent architecture for distributed systems.

On the other extreme, the EBES/EWOS Building Blocks, the eCo System, and the SEMPER model are essentially technology-neutral, and the actual implementation is free to use whatever technology may seem fit for the purpose.

In terms of openness, the eCo, EBES/EWOS, and SEMPER models are certainly more open than OMG or JECF. The eCo model is also the most generic one, which means that the actual implementation of an eCo-based system would certainly force the designer to make a number of choices, many of which have the potential for reducing openness. This holds for the SEMPER model as well. The EBES/EWOS approach rates rather favorably in terms of openness, as service blocks such as payment and certification may be based on other accepted standards (e.g., SET or ecash), and even externally implemented solution sets (e.g., Microsoft Merchant Server) could be incorporated in a system designed according to this approach. As for the OMG model, its use of CORBA means that interfacing to other component services should not be difficult - if these services provide a CORBA-compliant interface.

JECF, on the other hand, is limited by reliance on Java-specific technology, such as the cassette interface and the facilities provided by the Java environment, and interfacing to systems not based on Java is likely to be a nontrivial task. Given the (relative) simplicity of Java technology concepts, the JECF model may also be considered to be open – but only to component services implemented in Java. This does not necessarily mean that the JECF is inferior to other models - but it does mean that the choices for the application designer will be limited in certain respects.

In terms of stratification and integration, all models (with the possible exception of EBES/EWOS Building Blocks) are essentially layered, with components from different layers collaborate to provide the required functionality. The EBES/EWOS approach is layered as well, however it results in a hierarchically structured system, in which the essential relationship between components from different layers is aggregation, rather than cooperation. Some additional comments are in order. First, in the SEMPER model a separate layer is devoted to the management of exchanges of business items; this activity is part of the commerce services layer in both eCo and OMG models. Second, communication services, together with security, authentication, and identification services (and all of these are network-related - to some extent), are just components within the supporting services layer of the SEMPER model. On the other hand, the eCo System lumps all of these into the Network Services layer, while the OMG model delegates most of them to the so-called Common Services layer. Which of these approaches is better, and under what circumstances, remains an interesting topic for further research, both in theory and practice. Moreover, one might ask what is the optimal number of layers in a reference model? These appear to be interesting topics for further research, and experiences obtained fro the implementation of other reference models (such as the ANSI/SPARC database syste architecture [3] and the ISO/OSI reference model [10]) may well be worth revisiting.

Integration of different components and layers in the OMG and JECF model is implemented through the use of object-oriented techniques, such as the Java cassette interface in JECF or CORBA IDL specifications in the OMG model. The eCo and SEMPER models do not address this issue, although adherence to the object technology seems almost inevitable. Integration in the EBES/EWOS Building Blocks is essentially of the hierarchical, caller-callee type, and therefore reasonably independent of the technology used, but at the same time underdeveloped, and the missing detail certainly carries a potential for misinterpretation. The underlying infrastructure (corresponding to the eCo System network services) is assumed present and is not discussed in the architecture.

4 Further Research Issues

Each of the reference models reviewed presents a rather unique perspective of electronic commerce systems. The eCo System concentrates on high-level requirements of interoperability between electronic commerce application, and its treatment of requirements for such applications remains at a high level of abstraction. Therefore, it appears suitable as the foundation for creating other, more specific and hence more technology-dependent reference models.

EBES/EWOS Building Blocks present a more traditional, structured approach to the development of electronic commerce applications. It certainly contains some interesting notions, however it may not be best suited to modern, object-oriented development.

The OMG model is less generic than the eCo System because more details have been spelled out (and more design decisions have been made in the process), and because it uses the CORBA distributed object-oriented computing architecture. The SEMPER model results from a similar approach, although different solutions were adopted to the definition of requirements for electronic commerce applications. It may be interesting to note that the SEMPER consortiu seems to play roughly the same role within the European business community as does the OMG in the United States. As the volume of business transactions between US and Europe continues to grow, interoperability between OMG- and SEMPER-based applications may easily become an important requirement in not so distant future. As both models use a similar architecture, the interoperation of such applications should pose no insurmountable problems.

The eCo System, OMG, and SEMPER models provide a generic foundation for electronic commerce development, whereas the JECF offers a distinct implementation-oriented perspective at a lower level of abstraction. (The JECF framework could be considered as a Java specific software engineering framework for electronic commerce.) Consequently, it seems possible to design a system based on either eCo, OMG, or SEMPER model, and later implement it using the JECF framework. This hints that multiple models, differing in conceptual focus and technology content, may be needed in the process of design and development of electronic commerce systems; this topic certainly deserves more attention.

Future research can proceed in two directions. First, a more comprehensive analysis of both reference models and component service models and protocols could be undertaken, in order to gain deeper understanding of their respective strengths and weaknesses. Compatibility between reference models and component service models and protocols could also be investigated. In this way, application developers will be able to choose the optimal reference model according to the characteristics of the electronic commerce system they are about to design and implement. Second, the insights from the comparative analysis of existing reference models could lead to the development of a better reference model or models, either as a modification or generalization of an existing model, or (more likely) as a combination of two or more of them. In both cases, the final goal is to use the reference models to help develop new, powerful, flexible and interoperable electronic commerce applications and systems.

5 References

[1] Building blocks for electronic commerce. Final report (version 5.1) deliverable to the European Commission DG III/B2, European Board for EDI/Electronic Commerce Standardization, 1997.

[2] Business Team on Electronic Commerce. Report to JTC1: Work on electronic commerce standardization to be initiated, ISO/IEC JTC1 N 5296, May 1998.

[3] R. ElMasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, Redwood City, CA, 2nd edition, 1995.

[4] White paper: The Java Electronic Commerce Framework. Technical report, Sun Microsystems, Inc., Mountain View, CA, 1998.

[5] The Java Wallet architecture white paper. Technical report, Sun Microsystems, Inc., Mountain View, CA, 1998.

[6] K. Y. Jo, J. J. Pottmyer, and E. A. Fetzner. DoD Electronic Commerce/Electronic Data Iinterchange systems modeling and simulation. I *Proceedings of MILCOM 95*, volume 2, pages 479-483, San Diego, CA, Nov. 1995.

[7] S. McConnell, M. Merz, L. Maesano, and M. Witthaut. An open architecture for electronic commerce. OSM response to OMG Electronic Commerce Domain Task Force RFP-2, OSM, 24th Feb. 1997.

[8] The OBI Standard. V 1.1, OBI Consortium, June 1998.

[9] The OMG/CommerceNet Joint Electronic Commerce Whitepaper. Technical report, OMG, 27th Jul 1997.

[10] ISO 7498. Basic Reference Model for Open Systems Interconnection, International Standard, 1984.

[11] B. F. Schmid and M. A. Lindemann. Elements of a reference model for electronic markets. In *Proceedings 31st Annual Hawaii International Conference on System Sciences*, volume 4, pages 193-201, 1998.

[12] M. Schunter and M. Waidner. Architecture and design of a secure electronic marketplace. In *Proceedings 8th Joint European Networking Conference (JENC8)*, Edinburgh, UK, June 97.

[13] R. M. Soley, editor. *Object Management Architecture Guide*, Revision 3.0. John Wiley and Sons, New York, 3rd edition, June 1995.

[14] J. M. Tenenbaum, T. S. Chowdhry, and K. Hughes. eCo System: CommerceNet's architectural framework for Internet commerce. White Paper & Prospectus, version 1.0, CommerceNet, Inc., Mar. 1997.