

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2000 Proceedings

Americas Conference on Information Systems
(AMCIS)

2000

E-Commerce Systems: Architecture, Infrastructure, Model and Development Methodology

Akhilesh Bajaj

Carnegie Mellon University, akhilesh@andrew.cmu.edu

Keng Siau

University of Nebraska-Lincoln, siauk@mst.edu

Follow this and additional works at: <http://aisel.aisnet.org/amcis2000>

Recommended Citation

Bajaj, Akhilesh and Siau, Keng, "E-Commerce Systems: Architecture, Infrastructure, Model and Development Methodology" (2000).
AMCIS 2000 Proceedings. 66.

<http://aisel.aisnet.org/amcis2000/66>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

E-Commerce Systems: Architecture, Infrastructure, Model and Development Methodology

Akhilesh Bajaj
Carnegie Mellon University
Email: akhilesh@andrew.cmu.edu

Keng Siau
University of Nebraska, Lincoln
Email: ksiau@unl.edu

1. Introduction

The last decade has seen tremendous growth in the world wide web (WWW) (Berners-Lee, Caillau, Luotonen, Nielsen, & Secret, 1994) and e-commerce, which now offers organizations and consumers a unique channel to deliver and purchase goods and services. E-commerce is popularly classified as B2B (business to business), B2C (business to consumer) and C2C (consumer to consumer). B2B e-commerce, which is currently the fastest growing segment of e-commerce (Fellenstein & Wood, 2000), consists of making more efficient the flow of goods and services across organizations. This efficiency is typically increased using standards for electronic data exchange, automating certain processes across organizations and providing direct database access to external organizations, usually via the WWW. B2C e-commerce consists broadly of providing consumers uninterrupted access to an organization's goods and services, with minimal geographical boundaries. This is usually done by creating a virtual storefront, consisting of components such as shopping carts, catalogs, and electronic payment mechanisms. C2C e-commerce consists of providing distribution channels for consumers to sell goods and services to each other. Typical examples of this include auction sites, where users can auction items to each other, and consumer rating services, where consumers can share information about goods and services with each other.

The ubiquity of the WWW has also enabled changes in several aspects of traditional commerce. First, the barriers to entry for setting up a virtual business are considerably lower than for setting up a typical "brick-and-mortar" establishment. Second, geographical barriers are considerably reduced, providing sellers access to a larger customer base. Third, unique pricing mechanisms such as "name your own price," (e.g., priceline.com) and asynchronous auctions are now possible. Fourth, aggregation of demand by several individual buyers is now possible, leading to better negotiation of prices on behalf of these buyers. Fifth, search engines and rating sites provided aggregated information to the buyer about the different sellers of a particular good or service. It is widely accepted that all of these changes have enabled efficiencies and the creation of wealth.

The primary purpose of this work is to highlight areas of research in the construction of e-commerce applications. We do not look at lower level issues such as communication protocols or security mechanisms; though these are certainly large areas of research. Instead we focus on e-commerce application construction. While we do not claim to be exhaustive, our goal is to be as comprehensive as possible in covering different aspects of e-commerce application construction. The rest of this paper is organized as follows. In section 2, we list different aspects of e-commerce application construction, and indicate some research questions for each aspect. We present some general conclusions about trends in this area in section 3.

2. Different Aspects of E-commerce Application Construction

The aspects of e-commerce system construction we look at include the use of model-based methodologies, the use of component-based methodologies, the development of CASE tools for e-commerce construction, and the development of novel architectures of e-commerce systems.

2.1 Model-Based Methodologies

The motivation behind developing model-based approaches to hypermedia design has been to provide a defined set of primitives that are independent of the content of the actual application. Depending on the model, these primitives are useful in describing conceptual relationships in the content, specifying navigational links, and specifying the interface prior to implementation. Recently, there has been a thrust to incorporate formal design metrics as part of a model (Bajaj & Krishnan, 1999). We describe three model briefly to give a flavor for this area: the relationship management methodology (RMM) (Isakowitz, Stohr, & Balasubramanian, 1995), the object-oriented hypermedia design method (OOHDM) (Nanard & Nanard, 1995) and the Conceptual Model for Usable Web Applications (CMU-WEB) (Bajaj & Krishnan, 1999).

The RMM model consists of entities with attributes and 1:1 and 1:n associative relationships. The subset of the attribute set of a single entity that is shown on a single HTML (hypertext markup language) page is termed a slice. Navigation between slices of the same entity is done using uni- and bi-directional links. Navigation across entities is done using indices, guided tours or groupings. An index is a table of contents to a list of entity instances of the same type. A guided tour implements a linear path through a collection of items. A grouping is a high level menu that provides access to other features such as indices and guided tours. Using RMM involves drawing the entity-relationship diagram first. Next, different slices are constructed from the diagram, determining how information from within entities will be grouped for display purposes. These slices are organized into pages. Next, navigation paths are constructed between these slices. Relationships are used to determine these. So, if faculty teaches courses, then the “teaches” relationship may indicate that all courses taught by a faculty member must be accessible, as also all faculty members who teach a particular course. (Isakowitz et al., 1995) offer heuristic guidelines on how to construct RMM schema.

The RMM model focuses on interface construction and offers a data centric view (derived somewhat from traditional conceptual database modeling) of the interface, prior to the actual implementation.

The OOHDM model comprises of four different activities: conceptual design, navigational design, abstract interface design and implementation. These four activities may be done iteratively, with prototype-based development. In the conceptual design phase, the application is modeled using well-known object-oriented principles (Rumbaugh, Blaha, Premerlani, Eddy, & Lorensen, 1991) with additional primitives such as attribute primitives and sub-systems. The primary end product of this phase is a class diagram, with relationships.

The navigation phase recognizes that navigation patterns for the same conceptual schema are dependent on the users accessing the application. The primitives used in this phase are nodes, links, and access structures (indices, groupings, and guided tours). Nodes represent views of class attributes, so that a node can have one or more attributes listed from one or more classes. Links are defined from relationships in the conceptual schema, though more links can be added, depending on the user profile. The navigational class schema lists all the navigational nodes, with anchors designating links between nodes. The navigational context schema lists context classes, which describes what elements of a node are visible within a particular context.

The abstract interface design phase involves creating interface level classes and objects. Essentially, the method of representation of the different objects, and the user’s interaction (e.g., ‘mouse-clicked’) are listed here. The implementation phase involves converting the navigation and abstract interface schemas into concrete objects, using available tools.

The OOHDM model borrows heavily from object oriented concepts, though many of its concepts are essentially similar to RMM. The navigation concept and abstract interface design are attempts to address the actual usability of the application. Both RMM and OOHDM offer the ability to document the design, however, they do not address the key issue of what makes for a better design of the user interface. The recently proposed CMU-WEB model attempts to do this (Bajaj & Krishnan, 1999).

The primitives in the CMU-WEB model are information chunks, which are related using relationships, and hyperlinks, which are within and outside applications. An application consists of canvas views, each of which contains information chunks and hyperlinks. CMU-WEB has considerably fewer primitives than other methods, thus possibly making it easier to produce schemas of real world applications. The main advantage of CMU-WEB is that it offers quantitative metrics that can be evaluated for a particular CMU-WEB schema. These metrics measure usability requirements such as coherence of a canvas view, global coherence of the entire application, the cohesion of the information on each canvas view, and the coupling of information across canvas views. Usage of the CMU-WEB model involves coming up with a user interface plan, and then developing a CMU-WEB schema for the interface. The metrics for the schema can then be calculated. Interfaces that have already been developed can also be evaluated by creating a CMU-WEB schema for each interface and then evaluating the metrics for each schema.

Based on our survey of model-based methodologies, the key research areas appear to be the development of models that will a) be scalable so that they can be used for real world applications, b) support new systems development such as Web, E-Business, knowledge management, and ERP, and c) provide feedback on the goodness of the application, along dimensions such as usability and efficiency of the application. The emergence and standardization of Unified Modeling Language (UML) will have a profound impact on the subsequent development of model-based methodologies.

2.2 Component-Based Methodologies

Over the last decade, the usage of library-based components has evolved into CBSD (Component Based Software Development) (Clements, 1995). CBSD embodies the “buy, don’t build” philosophy proposed by (Brooks, 1987) and emphasizes composing software systems, rather than programming. The development of standards such as CORBA (Common Object Request Broker Architecture), Java Beans and COM (Component Object Model) means that several skeleton infrastructures are now available, into which pre-built components can be plugged. For example, it is possible for an organization to adopt a CORBA architecture, and to plug in CORBA components into an enterprise wide information system, including e-commerce components. The development of software components is now a growing industry.

The main research areas in CBSD from an application construction perspective include the construction of increasingly complex components, and attempts to standardize components so that a system can be composed of components purchased from different vendors.

Research on the construction of complex components is important, because well-designed components need to have strong internal cohesion, weak coupling with other components, and need to provide as generic an interface as possible, for maximum reuse. Lower level infrastructural components are relatively straightforward to define, but application level components need to follow different semantics for different applications. Thus, a low level communication protocol component will offer the same interface to all applications, but an application level component like a shopping cart will work differently in different e-commerce applications. One open research question is the definition of application level components so that they can be used in diverse applications.

The second research question related to reusability is the establishment of standards for application level components. Standard interfaces for components such as catalogs and shopping carts will allow the creation of a market for components, so that organizations can compose systems by purchasing components from different vendors. The requirements for a standard are that it provide a generic interface, and still provide flexibility so that components from different vendors can differentiate on criteria such as efficiency, cost and feature sets. We next briefly describe some representative standards that are evolving.

Several frameworks or models for e-commerce are evolving. Examples include the building blocks model (CEN & ISSS), the EC-DTF reference model ((OMG) & Net), the ecoFramework project (Net & Systems) and the Open Trading Protocol (OTP) specification (Consortium). The *building blocks model*, sponsored by the Information Society Standardization System (ISSS) and the European

Commission DG III/B2, is an attempt to identify key components (blocks) in business processes and to identify technical solutions to each of these blocks. The *EC-DTF reference model*, sponsored by Object Management Group (OMG) and CommerceNet, attempts to identify objects at different levels for enabling e-commerce. At the lowest level EC-DTF covers infrastructure services like payments. At a higher level, facilities like catalogs, brokerage and agencies are also covered. *EcoFramework*, sponsored by Veo Systems and Commerce Net, is meant to be a framework of frameworks. The four frameworks it addresses are: applications & services that model business processes; a common business language using messages and objects; an extensive set of interface specifications, class libraries and network services; and a layer of middleware that insulates applications from each other and platform dependencies. Recently, *ecoFramework* has expanded its scope to cover a variety of specifications such as Catalog Interchange Specification using the extensible markup language (XML) (Khare & Rifkin, 1997). The *OTP specification* is an attempt by the OTP consortium to specify a unified framework for trading over the Internet. The OTP architecture aims at identifying all the parties that play roles in e-commerce, and the sequences of key steps in electronic commerce such as delivery and payment. Examples include the consumer, the merchant, the deliverer and the consumer care provider. OTP also seeks to identify all the transactions that can occur between these parties and then uses XML to specify the content and format of messages that constitute these transactions.

All of these frameworks or models are still evolving and in time may be submitted for evaluation as standards. As new technologies such as XML continue to emerge, several of these frameworks incorporate the facilities offered by these technologies. Thus, document interchange specifications using XML are now being added into many frameworks. All of these frameworks are being created by committees or consortiums, and under sponsorship. A **top-down approach** is being followed to create them, and in their final form they are all likely to be very large and comprehensive, encompassing both B2B B2C e-commerce. Since all these frameworks have the same goals of identifying commerce objects and documents, it is doubtful if the business community will embrace any **one** of these standards for actually doing e-commerce, over the others. Hence, the two issues of good interfaces to components as well as the definition and adoption of standards that satisfy the requirements listed earlier are likely to be open questions in research.

2.3 CASE tools in e-commerce construction

While there has been some move towards developing CASE tools for e-commerce application construction, primarily from model based methodologies (e.g., OOADM, RMM), there is a paucity of literature on the subject, and we believe this is still a relatively open area of research. Research

questions in this area include a) how to define the primitives that need to be stored in repositories for e-commerce application construction, and b) how to provide rapid application development capability for e-commerce systems.

2.4 The Development of Novel Architectures For E-commerce Applications

For this work, we define the architecture of an e-commerce application to be the application level components, and the relationships between them. An example of a traditional B2C architecture would be an information page, a catalog, a shopping cart, and order form and a payment component, with obvious relationships between them. Over the last years, traditional architectures have evolved in all categories of e-commerce applications. An open research question is the proposal and testing of novel architectures. An example of a novel application level architecture is the security system offered by First Virtual Holdings (Borenstein et al., 1996) that offers acceptable security using non-encryption based techniques. The security is rooted at the application level, and stems from the sequence of simple steps that parties involved in a transaction must follow. The advantage of this architecture is that it only requires e-mail access, and hence can be used to collect payments in low trust, less developed economies. Another example of a novel architecture is the proposal of intelligent agents that can shop and negotiate on behalf of one party in a potential e-commerce transaction (see (Corp., 2000) for a listing of agents that can be used for e-commerce). As new technologies, such as XML, continue to emerge, we anticipate that they will enable new application level architectures.

3. Conclusion

In this work, we highlight some of the research areas at the application level, in the construction of e-commerce applications. The area of model based methodologies, in section 2.1, attempts to bring discipline to the construction of primarily customized applications. The area of component based methodologies, in section 2.2, aims to standardize the construction of applications. Applications that use components would usually not use model based methodologies. Thus there is a tension between the two areas, with one offering the advantage of disciplined customized applications, and the other offering easier to build components. The former also appears to be more mature than the latter, since components and component standards are only recently being proposed, and, to the best of our knowledge, none have enjoyed wide spread acceptance yet. The development of CASE tools, in section 2.3, stems mainly from model based methodologies, though there is no reason why CASE tools may not be proposed as repositories for components, with front-ends for composing e-commerce applications from components, as component standards emerge.

References

- (CEN), E. C. F. S., & (ISSS), I. S. S. S. *Building Blocks of Electronic Commerce* (<<http://www.cenorm.be/iss/Workshop/ec/Prb.zip>>).
- (OMG), O. M. G., & Net, C. *EC-DTF Reference Model* (<<http://www.omg.org/pub/docs/ec/97-06-09.pdf>>).
- Bajaj, A., & Krishnan, R. (1999). CMU-WEB: A Conceptual Model for Usable Web Applications. *Journal Of Database Management*, 10(4), 33-43.
- Berners-Lee, T., Caillau, R., Luotonen, A., Nielsen, H., & Secret, A. (1994). The World-Wide Web. *Communicatiuons of the ACM*, 37(8), 76-82.
- Borenstein, N. S., Ferguson, J., Hall, J., Lowery, C., Mintz, R., Morris, J., New, D., Parenti, B., Rose, M., Stefferud, E., Stein, L., Storm, C., Vielmetti, E., Weiser, M., & Wolff, P.-R. (1996). Perils and Pitfalls of Practical E-commerce. *Communications of the ACM*, 39(6), 36-44.
- Brooks, F. P. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer*, 20(4), 10-19.
- Clements, P. C. (1995). From Subroutines to Subsystems: Component Based Software Development. *The American Programmer*, 8(11).
- Consortium, O. T. P. O. *Open Trading Protocol* (<<http://www2.echo.lu/oii/en/commerce.html#OTP>>).
- Corp., I. c. (2000). *The Spot For All Bots On The Net* (<<http://www.botspot.com>> Accessed March 16, 2000.).
- Fellenstein, C., & Wood, R. (2000). *Exploring E-commerce, Global E-business and E-Societies*. Saddle River, NJ: Prentice Hall, PTR.
- Isakowitz, T., Stohr, E. A., & Balasubramanian, P. (1995). RMM: A Methodology for Structured Hypermedia design. *Communications of the ACM*, 38(8), 34-44.
- Khare, R., & Rifkin, A. (1997). XML: A door to automated web applications. *IEEE Internet Computing*, 1(4), 78-87.
- Nanard, J., & Nanard, M. (1995). Hypertext Design Environments and the Hypertext Design Process. *Communications of the ACM*, 38(8).
- Net, C., & Systems, V. *ecoFramework Project* (<<http://www.commerce.net/projects/currentprojects/eco>>).
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorenzen, W. (1991). *Object Oriented Modeling And Design*: Prentice Hall.