

December 1998

# An Object-Oriented Modeling Framework for an Enterprise-wide Distributed Computing System

S. Bhattacharjee

*State University of New York at Buffalo*

Ram Ramesh

*State University of New York at Buffalo*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1998>

---

## Recommended Citation

Bhattacharjee, S. and Ramesh, Ram, "An Object-Oriented Modeling Framework for an Enterprise-wide Distributed Computing System" (1998). *AMCIS 1998 Proceedings*. 316.

<http://aisel.aisnet.org/amcis1998/316>

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 1998 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# An Object-Oriented Modeling Framework for an Enterprise-wide Distributed Computing System

Sudip Bhattacharjee

R. Ramesh

Department of Management Science & Systems

School of Management

State University of New York at Buffalo

## Abstract

*This research focuses on developing an object-oriented framework for the design of an enterprise-wide distributed computing system. We define the objects in the enterprise, develop the overall framework and model the inter-relationships between objects, and outline an approach for implementation of the framework.*

## Introduction

Recent advances in information technology have enabled the widespread use of distributed computing systems, leading to decentralized management of information. This has been aided by and has in turn fueled intense competition in business with faster and more accurate data storage and retrieval and information processing (Barua, Ravindran and Whinston 1997). Several organizations have achieved high efficiency, including ease of use and lower costs in operations by adopting a client/server computing framework (King 1996). Moreover, system integration and interoperability issues are being intensified (Manola 1995) as organizations are moving from mainframe based processes towards an open, distributed computing environment, and is compelling firms into an accelerated construction of large-scale distributed systems for operational use.

The focus of this research is to develop an object oriented (OO) framework for design and development of an enterprise-wide distributed computing system. A framework can be defined as a reusable, high-level abstraction of a system that can be tailored to meet specific scenarios (Fayad and Schmidt 1997). OO frameworks provide a modular structure, are portable and reusable from one system scenario to another (with little modification), and provide an inversion of control to system designers. Hence an OO framework for an enterprise-wide distributed computing system will aid in rapid analysis and design of distributed systems, and will reduce the need to develop correct, portable and efficient designs of large-scale distributed systems for an enterprise from scratch every time.

## Object Oriented Distributed Computing Framework Design

An enterprise-wide computing system consists of *users* scattered throughout different geographical locations of the firm, who access *application systems* across locations which may be hosted on diverse platforms. These applications in turn access *databases* across the enterprise and return the information to the users. Fig. 1 depicts a high-level view of the objects and their interactions in an enterprise. The *user objects* in this framework, denoted by  $U_i$ , comprise of the types of users in the organization, eg. manager, clerk, support staff, developer, etc. The user objects in the organization are grouped into various business domains, like customer service, billing service, accounting services, etc. *Application system objects*, denoted by  $A_i$ , are hosted across the organization in various locations, and access *database objects*  $D_i$  which are likewise distributed across the organization. An instance of a user object may exist in more than one business domain; similarly, an application or database object may be used across multiple business domains. User and application objects may have a many-many relationship (denoted by  $n:m$ ), and applications and databases likewise may have a many-many relationship between them.

The *business system functionality* is defined at the object-object interface. An *association matrix* of user and application objects defines the relationships and functional interface between the user-application objects. This approach helps in separating the objects from functions, hence the interface architecture can be easily reused, extended or modified without any change in the respective object properties. For example, a user type  $U_i$  belonging to a certain business domain may access certain functions of an application  $A_k$ , while the same user type belonging to another business domain may access other functions of the same application  $A_k$ . Regardless of the functional change, the properties of neither objects require any change in the framework – only the interface functionality definition is modified depending on the business domain the user belongs to. This can be easily implemented in the design and development of the user interface framework (Fig. 2). Similarly, an application-database object association matrix developed for the enterprise defines the interface functionality between those types of objects.

## System Development and Implementation Issues

Physical implementation of the framework to develop a distributed computing system includes decomposing the framework into related subsystems, and defining the subsystem interfaces. Also, a physical determination must be made on the location of applications and databases, including defining their accesses by the various user objects, with the aim of improving performance of the system with respect to system utilization, reliability, security, availability and query response times. For implementation across multiple platforms, middleware services must be included in the design decisions.

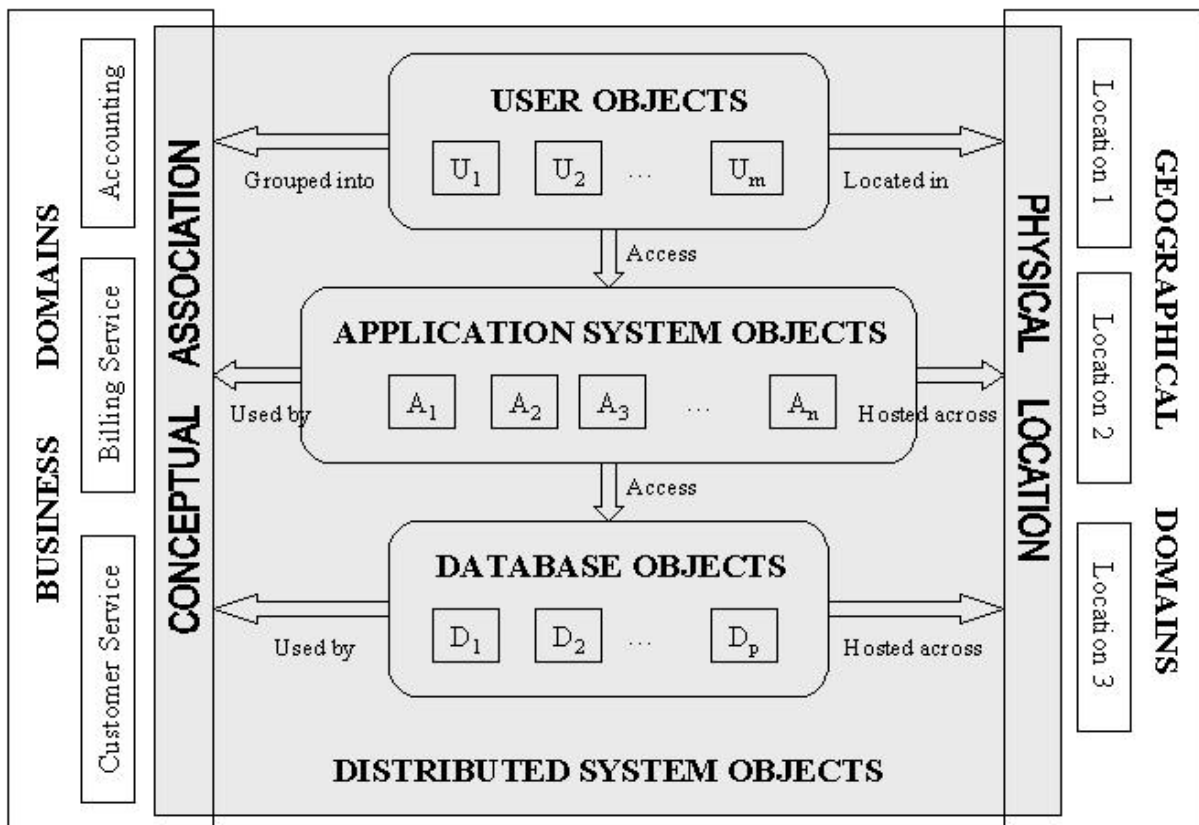


Figure 1. Conceptual Model of Entities and Interactions

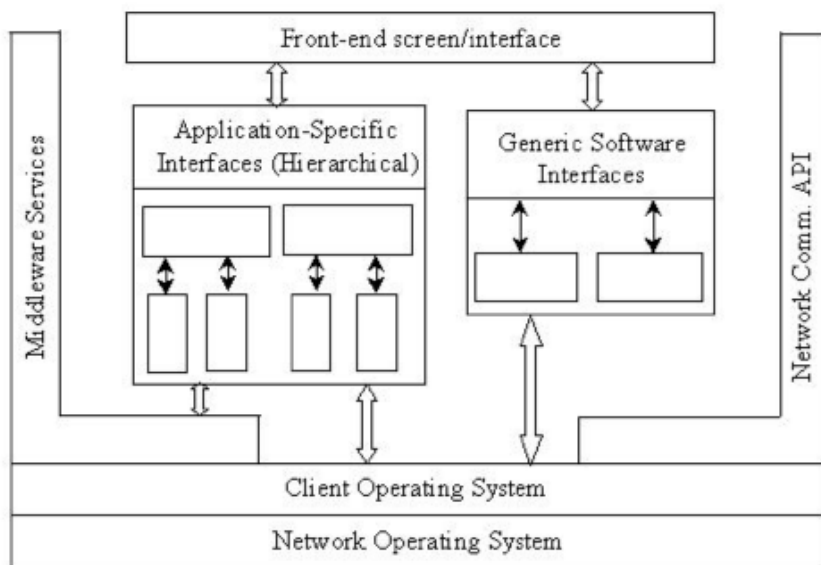


Figure 2. User Interface Architectural Framework

Research exists on various subparts of the distributed computing systems design problem. Studies include OO application framework development (Baumer et. al. 1997; Fayad et. al. 1997), and distribution of OO applications among heterogeneous platforms (Purao 1995). An ongoing research (Bhattacharjee and Ramesh 1998) examines an integrated model for infrastructure and resource planning for an enterprise-wide distributed computing system.

### Conclusion

In this study, we develop a modeling approach for OO framework development for an enterprise, which provides modularity, reusability and easy extensibility of the frameworks to other related scenarios. The objects in an enterprise are modeled and their conceptual associations with the business domains are utilized to define object-object interface functionality. It is shown that these

interface architectures are reusable and portable, and can be easily changed without major modifications of the entire system. This approach presents an elegant and efficient method to model an enterprise-wide distributed computing system. An extension of this work discusses the implementation issues of the system, including location decisions of applications and databases.

### References

References available from the authors upon request (bhats@acsu.buffalo.edu; ramesh@acsu.buffalo.edu).