Dependable Objects for Databases, Middleware and Methodologies: A Position Paper

Marion G. Ceruti,
Senior Member, IEEE
Space and Naval Warfare Systems Center,
San Diego, D4121, 53560 Hull Street
San Diego, CA 92152-5001, USA
Tel. (619) 553-4068, FAX (619) 553-4068
ceruti@spawar.navy.mil

Bhavani M. Thuraisingham,
Senior Member, IEEE
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730, USA
Tel. (781) 271-8873, FAX (781) 271-8752
thura@mitre.org

Abstract
This position paper describes work that needs to be done to integrate features of security, fault tolerance, and real-time computing into object-oriented technology to produce “dependable objects.” This dependable technology can be applied in the areas of database, middleware, including object-request brokers; and object-based design and analysis methodologies. The accomplishments to date of various groups and organizations are reviewed, and some preliminary ideas for new research directions are revealed.

Keywords - Command and control, dependable object-request broker, methodology, middleware, object-oriented database, real-time system

1. Overview
Object technology has become common practice for many. For example, The Department of Defense has conducted or sponsored extensive investigation on using object technologies for various applications, such as command and control systems [1, 16, 25 and 26], Airborne Warning and Control Systems (AWACS) [12], and real-time simulations [13]. At the MITRE Corporation, a major effort was undertaken between 1993 and 1999 to investigate evolvable and adaptable middleware and data management for command and control applications. This effort focused mainly on real-time aspects of objects. More details of this work are available in [3]. Moreover, recent progress has been made with scheduling in real-time, object-oriented systems that implement the Object Management Group’s (OMG) Common Object-Request Broker Architecture (CORBA). (See, for example, [7-10, 13-15, and 26]).

However, for systems to survive in the midst of failures and threats, real-time objects are not sufficient. One needs these object systems to handle threats and failure in a timely manner. Therefore, features like security and fault tolerant computing have to be incorporated into these objects [24]. Whereas real-time and fault-tolerant CORBA recently has been reported in a case study [15], to date, very little work has been done on integrating security, real-time and fault tolerance computing. We have named objects that incorporate all of these features, “dependable objects.”

Object technology includes programming languages; databases; middleware such as Object-Request Brokers (ORBs); design and analysis methodologies; and finally, components and frameworks. Various efforts are underway to develop real-time programming languages, secure databases, real-time database, secure middleware, real-time middleware, and design and analysis methodologies for real-time and secure systems. However, none of these efforts integrates all of the features. In this paper, we investigate dependable objects for three aspects of object technology: databases; middleware such as ORBs; and design and analysis methodologies. Section 2 addresses dependable object databases. Section 3 covers dependable middleware such as ORBs, and section 4 addresses dependable object-based design and analysis methodologies. We discuss initial ideas in this paper. Much research is needed before dependable object systems can be developed successfully. For each area we discuss the status and directions. The paper is summarized in section 5.

2. Dependable object databases
Object databases are now commercially available and in use throughout the Department of Defense. (See, for example, [1, 16 and 25]). These include object-oriented databases that make object-oriented programming languages persistent as well as extended relational databases that extended relations with objects. Object databases are based on object models. Object models include features such as
class hierarchy, object instances, inheritance hierarchy and method execution [2 and 4].

Extensive investigation has been carried out on secure objects. (See, for example, [21]). Some have extended objects with multilevel security properties. Others have extended objects with access control policies. (See, for example, [5]). For example, should access be granted at the instance level or instance variable level? Whereas the research on secure object databases was extensive during the late 1980s and early 1990s, work on real-time object databases began in the mid 1990s. These databases extend objects with properties for specifying time constraints on data as well as methods. For example, data are assigned time stamps. Methods also may have timing constraints associated with them. Furthermore, the concurrency control algorithms have integrated techniques such as semantic locking and priority ceiling (See, for example, the work at the University of Rhode Island and at MITRE [3]). Fault tolerance object databases have received little attention, although most commercial object databases have techniques for recovering from failures. For examples, techniques such as two-phase commit, backup and recovery have been developed for object databases. Note that standards groups such as the Object Database Management Group (ODMG) are developing specifications for security. However, real-time specifications are yet to be attempted although various research groups have carried out some work in this area.

The challenge now is to integrate security, real-time and fault tolerance features for object databases. For example, consider the authorization models developed by Bertino and others [5 and 6]. How can these models be extended to include the real-time features? (See, for example, [7 and 23]). Furthermore, when fault tolerance features have to be integrated, what extensions need to be made to the basic object model? Then how can these extensions be incorporated into models that have security and real-time features? What are the tradeoffs?

Presently, CORBA cannot support real-time applications because it lacks essential Quality-of-Service (QoS) features [9 and 10]. Feng recently has proposed an approach that provides QoS guarantees and run-time scheduling flexibility in a CORBA system [10]. However, the question remains, can a detailed QoS model be developed to specify tradeoffs between security, real-time and fault tolerance features? We need answers to these questions in a basic research program designed to investigate these issues.

3. Dependable middleware and object-request brokers

Next, we examine distributed objects and middleware. Middleware contributes an essential part of the infrastructure for large-scale systems. The OMG, which was founded in 1989, originally developed middleware specifications called ORBs. Since then, various additional specifications have been proposed to enhance ORBs. Since the early 1990s, the focus was on specifying various services, such as those for queries, transactions, security, and real-time systems.

In MITRE’s effort on developing evolvable, real-time command and control systems, ORBs were selected for the middleware. Since 1995, ORBs have come a long way. (See, for example, [3 and 18]). For dependable middleware, we are particularly interested in the services for security, real-time implementations and fault tolerance.

Around 1996, the security services were specified to include policies for authorization and access control for ORBs. Today we are beginning to see commercial products emerge that satisfy the specifications. Investigation of real-time services began in early 1996 and the specifications started coming out in 1998. These include specifications for static as well as dynamic scheduling systems. The real-time, special-interest group at OMG also have developed specifications for fault tolerant ORBs.

Whereas some researchers have discussed integrating real-time, security, and fault-tolerance features for ORBs, no one has performed this work. We need to develop a research program in this area. Moreover, the various special interests groups at OMG should develop combined specifications. Due to the potential benefit in this area, we can expect to see ORB specifications for integrating various features. That is, dependable ORBs could become a reality in a few years. However, much still needs to be done. We also need to explore the notion of QoS for these ORBs.

4. Dependable object–based methodologies

The last object technology we investigate here is object-oriented design and analysis methodologies. These include various approaches such as those of Rumbaugh, Booch and Jacobson. However, recently they have combined their approaches and developed a joint methodology called Uniform Modeling Language (UML) [19]. Moreover, OMG adopted UML as a standard in 1997 [20]. Therefore, any effort for developing dependable objects should focus on UML [11].

Much effort has been focused on extending UML for real-time systems. (See, for example [20]). As a result, RTUML has been specified. Some early work was carried out on extending Object Modeling Technique (OMT) for real-time systems [22]. Essentially, the idea is to extend the various phases of object modeling to incorporate time constraints. With respect to security, initial work on extending OMT for secure applications was proposed in [17]. The focus of this effort was on multilevel security. However, much remains to be done on extending UML and RTUML for secure applications.

To develop dependable object-based methodologies, we need to integrate RTUML with a methodology that includes secure UML and fault tolerant UML. However, to our knowledge, no one has accomplished this. Another area to explore is the QoS aspect of design and analysis methodologies. For example, we want to answer ques-
tions such as, “What type of QoS primitives can be specified for these methodologies?” We need to conduct a research program to investigate these issues.

5. Summary and directions

A critical requirement for a command and control system is that its middleware and its various other components work well together. For example, often security and real-time processing are conflicting goals. Therefore, one needs to determine adaptable security policies that simultaneously meet the real-time constraints and maintain security. Therefore, integration of the various requirements is important. Furthermore, graceful degradation of the system in the presence of faults, malicious processes, and intrusions is essential. To accomplish this goal, research is needed on integrating fault tolerance, security, and real-time processing to develop evolvable; survivable; and adaptable command, control, and communication. We believe that an open implementation of the infrastructure is essential to developing survivable systems that can adapt to changing environments.

In this paper, we first discussed the need for integrating security, real-time and fault tolerance features for object-based systems and then introduced the notion of dependable objects. We identified five different types of object technologies and explored the notion of dependable objects for three of these technologies. In particular, we discussed dependable object databases, dependable middleware and dependable object-based methodologies. Other dependable objects that we did not discuss in detail include object-based programming languages and object-based components and frameworks.

We have provided only some preliminary ideas on dependable objects. These ideas have to be explored further and much research has to be completed before dependable objects can be implemented successfully. We have provided the initial direction for this important research area.

Acknowledgments

This work was produced by a U.S. government employee as part of official duties and is not subject to copyright. It is approved for public release with an unlimited distribution.

References


**Dr. Marion G. Ceruti** is a scientist in the Advanced Concepts and Engineering Division of the Command and Control Department at the Space and Naval Warfare Systems Center, San Diego. She received the Ph.D. in 1979 from the University of California at Los Angeles. Dr. Ceruti’s professional activities include information systems research and analysis for command and control decision-support systems. She has served on the program committees and as a session chair for workshops, colloquia, and conferences since 1987. She is the author of numerous publications on various topics in science and engineering. Dr. Ceruti is a senior member of the IEEE and a member of the IEEE Computer Society, the Armed Forces Communications and Electronics Association (AFCEA), and the New York Academy of Sciences.

**Dr. Bhavani M. Thuraisingham** is a senior principal engineer at the MITRE Corporation, Bedford Massachusetts. She received her Ph.D. degree from the University of Wales, Swansea, UK. Dr. Thuraisingham’s professional activities include data management, data mining, database security, real-time and object-oriented, distributed systems. She has served on IEEE program committees, often as chair, and on numerous panels for workshops, colloquia, symposia and conferences. She has been featured as an invited speaker for many conferences, including the AFCEA Federal Database Colloquia from 1994 to 1999. Dr. Thuraisingham is the author of over 300 technical publications and reports, including over 40 journal articles and several books and book chapters. She holds three US. patents for MITRE on database inference control. Dr. Thuraisingham is a senior member of the IEEE and a member of the IEEE Computer Society, the ACM, AFCEA, and the British Computer Society.