Data Manager for Evolvable Real-Time Command and Control Systems

Eric Hughes, Roman Ginis, Bhavani Thuraisingham, Peter Krupp, John Maurer
The MITRE Corporation, 202 Burlington Rd., Bedford MA, 01730-1420
{hughes,ginis,thura,pck,johnm}@mitre.org

Abstract

We describe the design and implementation of real-time data management services. These services combine technologies developed in the context of real-time distributed object management, object DBMSs, and scheduling. This combination simplifies many of the services, and produces a result which is greater than the sum of its parts, because it can be used to improve the portability and flexibility of real-time applications.

1 Introduction

Many real-time command and control systems have been deployed which use custom-written real-time data management (RTDM) software that is highly tuned to particular requirements. This custom software is not only expensive to develop, but because it is entwined with the applications, must be rewritten when requirements change. We have developed an approach that makes real-time command and control software more evolvable by using shared, flexible RTDM services.

This paper surveys past and ongoing RTDM and distributed object systems work by the authors and several related groups. At MITRE, this work involves mission-oriented research as well as a prototyping lab, both funded by the US Air Force (USAF). These two groups participate in the transfer of RTDM technology from research to implementation. We also discuss current implementation efforts for the USAF's AWACS (Airborne Warning and Control System). Modernization of the AWACS mission computing system involves roughly a 40 staff-year effort over the next three years. As with many such efforts, there is not enough money available to replace the system (AWACS) in one block upgrade, so the approach is to use partial upgrades as an opportunity to migrate the system gradually.

We use the following terminology to properly characterize and give attribution for this work. Our RTDM *design* is a product of MITRE research in cooperation with Victor Fay-Wolfe at the University of Rhode Island (URI). A descriptive *white paper* on a real-time Common Object Request Broker Architecture (CORBA) [COR96] is being developed by the real-time PSIG of the Object Management Group (OMG); a standards effort. We have *prototyped* some of the key elements of our design. MITRE AWACS personnel, USAF, Lockheed-Martin, and Boeing have provided valuable *feedback* on the design and prototypes. Finally, we relate our work to the *implementation* being done by Lockheed-Martin and Boeing.

We began by examining the problem of building distributed, fault-tolerant, hard real-time command and control systems, in collaboration with the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU) [SLB+92]. We then identified and evaluated the flexibility of several architectures, with feedback from the AWACS program, and chose a design based on distributed objects [BFG+95]. We also worked with the URI group to extend CORBA to support real-time systems [TKH+96]. One of the results of this work was the identification of requirements for the use of real-time CORBA in command and control systems. We prototyped real-time CORBA by porting Xerox's Inter-Language Unification (ILU) Object Request Broker (ORB) [PAR97] to the Lynx real-time operating system [Lyn97] and by providing a distributed scheduler supporting rate-monotonic and

Proceedings of the 23rd VLDB Conference
Athens, Greece, 1997

deadline-monotonic techniques.

2 Approach

Here we outline our approach for RTDM services for command and control applications. Our design consists of services that form a framework for real-time systems, as shown in Figure 1. Our infrastructure insulates applications from the underlying real-time operating system. The ODBMS, Event Manager, and scheduler also use this infrastructure. It is relatively easy to migrate a legacy system to use our infrastructure [HR97]. As with ordinary ODBMS and CORBA systems, migrated applications can then evolve with minimal changes to RTDM services. Many technologies are important to this work but here we discuss only the critical subset.

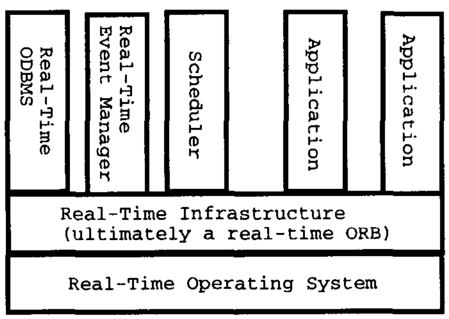

Figure 1: Evolvable Real-Time System.

We have chosen an object-oriented model for many of the same reasons given for non-real-time applications. We also have found objects to be convenient for the specification of real-time constraints (e.g., how often an object should be updated) and of semantic concurrency-control metadata (e.g., which methods of an object can be executed concurrently). This approach in effect extends the object model to support real-time constraints, as is done in RTSORAC [FWCPP94]. In most cases, this involves the creation of (new) object definitions for data in legacy systems, and these object definitions are then annotated with real-time metadata. For an existing object-oriented system, we believe that annotation with real-time information and enforcement of real-time constraints improves the system.

At the heart of our RTDM services is a real-time ORB. The use of an ORB greatly simplifies communication between distributed objects, resulting in a more natural and flexible integration of systems. We have used the Lynx operating system environment and ILU to prototype a real-time ORB (as described in the real-time CORBA white paper). We have also ported an experimental surveillance and tracking application of AWACS to the prototype ORB while leaving the remaining AWACS applications in the legacy environment. The ported surveillance and tracking application uses our prototype data manager to store, update, and retrieve information on tracks, which represent sensor information about aircraft.

ORBs are not optimized for efficient object transfer and caching, which are better handled by an ODBMS [CB97] that provides additional RTDM services. In our design, the ODBMS uses the ORB for communication that does not involve transfer of objects, such as messages describing triggered events. In the Lockheed implementation, the ODBMS uses the ORB to allow a local thread to connect to a remote database. In addition, in our design, the ORB uses the ODBMS for transactions and persistence of objects. For example, the Naming Service of the ORB could use the ODBMS to store its mapping from object references to implementations.

Other approaches have been taken to integrate ORB and ODBMS. The OMG Persistent Object Service (POS) allows CORBA objects to be stored and retrieved using many possible means (files, databases, etc.) [OMG94]. However, this is limited as a means for integration of real-time ORBs and ODBMSs. For example, the POS ignores the role of the ODBMS in efficient transfer and caching of fine-grained objects. Reverbel has investigated ORB and ODBMS integration in the context of medical information systems, but his efforts are geared to commercial ORB/ODBMS adapters [Rev96]. It remains to be seen whether these adapters could be modified for use in real-time systems.

In our design, both ORB and ODBMS are further simplified by the use of a scheduler, which completes the set of RTDM services. This approach is detailed in the real-time CORBA white paper and is being implemented by Lockheed. The scheduler allows threads to request resources for computation based on deadlines and priorities assigned. Many of these computations may involve ODBMS transactions. The concurrency control service of the ODBMS may need to change the priority of a computation (as described below), which in our design can be achieved by requests to the scheduler. Our design also isolates applications from the operating system by providing scheduling, communication, and memory management services, which improves the portability of applications (even beyond the portability afforded by POSIX-compliant operating systems). This approach is being carried out in the current implementation.

In our design, the ODBMS provides constraint enforcement, concurrency control, querying, and persistence services. We prototyped an event manager for constraint enforcement, and found that this approach is viable for command and control applications. However, we used a real-time relational DBMS (RDBMS) [THK+96] with this prototype, and found that the poor performance of the RDBMS and the impedance mismatch between the event manager and underlying databases makes the RDBMS a poor choice. Our current design adapts the event manager to an ODBMS [THK+96]. Lockheed's implementation supports the CORBA Event Service, which in turn supports the specification and delivery of events. Our constraint enforcement design uses this Event Service to create events caused by updates to ODBMS objects and to deliver these events to the appropriate handlers, which can enforce constraints. For example, a tracking application might specify a constraint on how often a track must be updated, and violation of this constraint should cause an event which eventually results in an update to the track. This approach improves the evolvability of applications by making such constraints explicit and easier to change as the system requirements change.

We are pursuing several possibilities for concurrency control. We designed the *affected set priority ceiling* (ASPC) protocol which integrates priority ceiling with semantic locking, and we have implemented a prototype data manager with this protocol [TKH+96]. In ASPC, the concurrency control service may need to change the priority of a transaction, and this can be done by a request to the scheduler. Lockheed is also investigating a multi-version concurrency control approach. Soparkar et al. discuss other options, including optimistic concurrency control [SKS96]. Our design allows for a choice of concurrency control protocol based on the needs of the application. We also prototyped a metadata manager which stores and controls access to metadata (e.g., real-time constraints) for transactions and database objects. In the prototype, object definitions and transactions are pre-analyzed, and the resulting metadata is used to determine potential conflicts between executing transactions.

A Query Service was not part of our original design, but feedback from AWACS groups and current implementation suggest that a real-time ODBMS should support an object query language (OQL). Simple real-time queries can be compiled into efficient ODBMS accesses, which allow end users to interact with an object database in a familiar fashion. Queries are also useful for non-real-time access to databases (e.g., maintenance and browsing). Lockheed is implementing a simple OQL for AWACS based on ODMG [CB97].

In our design, the ODBMS is primarily an in-memory database, which agrees with the consensus of the real-time DBMS community [Sop96]. However, objects still must be made persistent, either to support recovery of ODBMS transactions or other application needs. Our data manager prototype used shared memory as the primary store and our design includes persistence to store database objects. The Lockheed implementation uses in-memory databases but will provide persistence for logging, checkpointing, and application needs.

We believe that a temporal object model is extremely important for real-time applications, and this model should be supported by RTDM services. In our design, objects can be defined to have temporal constraints (as described above), timestamps, temporal checkpoint requirements, etc. As with concurrency control, our design leaves the choice of temporal object model to the needs of the system.

3 Future Work

Commercial support is emerging for several key RTDM services. We expect that commercial implementations of real-time ORBs (including the Lockheed implementation) will be available within the next two years. One of the important challenges for these products is the ability to adapt to the needs of the system, such as choice of concurrency control protocol or temporal object model. We suspect that real-time ORBs will require other forms of adaptability, such as choice of communication protocol, selectable based on the system state and quality of service requirements. We are currently investigating ways to achieve this adaptability [GHK+97].

There are many possible applications of the RTDM services we have described. We have discussed AWACS, a system that combines the avionics and command and control domains. Another domain is information survivability, in which a user could bring down a system by over-using its resources [KTM97]. Real-time ORBs have application in many domains [COR96], and real-time ODBMSs have similar potential [Sop96].

4 Acknowledgments

The MITRE mission-oriented research was performed under contracts with the US Air Force via Rome Labs. We wish to thank the members of the cited groups for their consideration of and feedback on new technologies. MITRE AWACS personnel, including Jonathan Bernhardt, have been willing users of new technology and new project management approaches. We wish to thank Maj. Michael Milligan and the AWACS project leadership of the US Air Force for their support. Our work has benefited from collaboration with Victor Fay-Wolfe at URI and with the real-time CORBA PSIG. We thank Lockheed-Martin for their willingness to exchange implementation ideas and Boeing for their support.

References

[BFG+95] Ed Bensley, Larry Fisher, Michael Gates, James Houchens, Arkady Kanevsky, S. Kim, Peter Krupp, Alice Schafer, and Bhavani Thuraisingham. Evolvable Systems Initiative for Real-Time C3: Infrastructure Requirements. In *ICECCS*, Nov. 1995.

[CB97] R. G. G. Cattell and Douglas K. Barry. *The Object Database Standard: ODMG 2.0.* Morgan-Kaufmann, 1997.

[COR96] CORBA/RT PSIG. *CORBA/RT White Paper.* OMG, 1996.

[FWCPP94] Victor Fay-Wolfe, Lisa Cingiser, Joan Peckham, and Janet Prichard. A Model for Real-Time Object-Oriented Databases. In *Conference on Database and Expert Systems Applicaitons*, 1994.

[GHK+97] Roman Ginis, Eric Hughes, Peter Krupp, John Maurer, and Bhavani Thuraisingham. *Object-Oriented Data Management and Distribution for Real-Time Command and Control Systems.* To appear in RTDB, 1997.

[HR97] Eric Hughes and Arnon Rosenthal. *Migration of Legacy Applications to Use Enterprise Services.* In preparation, 1997.

[KTM97] Peter Krupp, Bhavani Thuraisingham, and John Maurer. Survivability Issues for Evolvable Real-Time Command and Control Systems. In *DARPA Information Survivability Workshop*, 1997.

[Lyn97] Lynx. The Lynx Operating System. In *http://www.lynx.com/products/lynxos.html*, 1997.

[OMG94] OMG. *Common Object Services Specification, Vol. I.* Doc. No. 94-1-1, March 1994.

[PAR97] Xerox PARC. Inter-Language Unification (ILU). In *ftp://parcftp.parc.xerox.com/pub/ilu/ilu.html*, 1997.

[Rev96] Francisco Reverbel. *Persistence in Distributed Object Systems: ORB/ODBMS Integration.* PhD thesis, University of New Mexico, April 1996.

[SKS96] Nandit Soparkar, Henry Korth, and Abraham Silberschatz. *Time-Constrained Transaction Management.* Kluwer Academic, 1996.

[SLB+92] L. Sha, J. P. Lehoczky, M. Bodson, P. Krupp, and C. Nowacki. Position Paper: Responsive Airborne Radar Systems. In *IEEE International Workshop on Responsive Systems*, 1992.

[Sop96] Nandit Soparkar. Post-Workshop Report for Workshop on Databases: Active and Real-Time (DART). In *http://www.eecs.umich.edu/~soparkar/Dart96/report.ps*, 1996.

[THK+96] Bhavani Thuraisingham, Eric Hughes, Peter Krupp, Gary Gengo, Alice Schafer, and Mike Squadrito. Active Real-Time Database Management for Command and Control Applications. In *Databases: Active and Real-Time (DART)*, 1996.

[TKH+96] Bhavani Thuraisingham, Peter Krupp, Eric Hughes, Mike Squadrito, and Victor Fay-Wolfe. Transaction Model and Concurrency Control for Evolvable Real-Time Command and Control Systems. In *ATMA 96*, 1996.