Abstract

"Avionics have grown in complexity and importance to the point where they account for up to 40 per cent of the total cost of a combat aircraft. In an effort to reduce this figure, while continuing to increase performance and reliability, air forces and industry are increasingly turning to integrated avionics. [1] Wright Laboratory's Functionally Integrated Resource Manager (FIRM) program is attacking the heart of this process with an integrated database management system. FIRM will provide real-time, secure data services for the integrated avionics systems that will be aboard future Air Force fighter aircraft, eliminating costly duplication of data and data access software, while providing a smooth integration path for future software and hardware upgrades. The key aspect of this system is its residence within and control of data within the integrated architecture. The advantages are faster turn-around time for data updates (i.e., threats, targets), reduced software development costs, and lower software life cycle costs. Wright Laboratory and its two prime contractors [2] adopted an object-oriented approach to defining and analyzing requirements and providing a development strategy for FIRM. We report on the first two phases of the FIRM program: the requirements phase where the contractors analyzed actual avionics missions to determine the drivers and stressors for the database system, and the functional design phase where the contractors performed further analysis on the pilot/vehicle interface and inferencing and correlation techniques, and developed the schema and architecture for the system. We identify several critical software technologies: real-time distributed database technology, multi-level secure database technology, and object-oriented technology, which FIRM will need to exploit to be successful.

1 The FIRM Program

Modern warfare's ever increasing demands for more information, more comprehensive information, more accurate information, more timely information, and more efficient use of information will revolutionize aviation systems. Information is data empowered with knowledge. The art is to impart this knowledge into the user in a manner easily assimilated. A major component of this revolution is the need to overhaul our approach to avionics computing and internal information transfer architectures for aircraft [2]. The avionics processing system shown in Figure 1 is representative of the type of architecture required in order for this revolution to take place. Modular avionics processing systems currently under development by the Air Force, Navy, and Army provide an opportunity to move away from current federated systems. These integrated systems will provide faster processors with large global memory and faster local memory to further improve access speed and data throughput. A large random access global memory will allow data downloaded from the mass memory to be accessed quickly by any processing element. A parallel data bus will allow fast access to information in the global memory created by other applications running on separate processing elements. Most importantly, an integrated architecture and its operating system will be essential to the implementation of a capable database management system (DBMS) within an avionics platform [3].

The FIRM program is designing a real-time database management system for the integrated avionics systems that will be aboard future fighter/attack aircraft. This concept can also be applied to current federated systems that will be updated with modular processing systems. The FIRM program is an Ada programming language software technology effort that will use an object-oriented database management model for avionics data management.

The goals of phase 1 of the FIRM program were to determine the data sizing and timing requirements for
fighter/attack aircraft operating in the 2005 time frame and to determine the DBMS requirements for these kinds of aircraft. The requirements were determined by analyzing mission profiles and identifying necessary functions and performance criteria. The phase I program has developed a system segment specification and will document DBMS requirements in a forthcoming software requirements specification. The documentation will be used as an input into a subsequent program that will develop a FIRM prototype. The prototype effort will begin in 1994 with software development and end with a laboratory demonstration and flight test in 1997.

In the rest of this paper, we summarize the requirements analysis that was done during phase I and identify the critical software technologies: real-time distributed database technology, multi-level secure database technology, and object-oriented technology, which FIRM will need to exploit to be successful.

2 Requirements Analysis

The aircraft missions analyzed to derive data requirements and DBMS functions were Battlefield Air Interdiction (BAI), Close Air Support (CAS) and Defensive Counter Air (DCA). The BAI mission puts the aircraft in a scenario attacking second echelon armor (typically tanks) moving towards the front lines. Targets are away from front in enemy territory, aircraft is loaded with appropriate weapons, and the area harbors threats such as surface to air missile (SAM) systems. The driving data size and access rate requirements for this scenario are for target identification, using synthetic aperture radar and infrared (IR) sensors, and a digital map. The CAS mission is carried out close to the front in support of friendly armor in contact with enemy forces. Tanks are the target again so the aircraft is appropriately loaded. Threats are small arms, anti-aircraft artillery, and SAMs. The driving data size and access requirements are derived from the same functions as those in the BAI mission. The DCA mission considered is to attack threat aircraft penetrating friendly airspace. The target is an aircraft penetrating friendly airspace and threats are the enemy escort aircraft for the target. The data size and timing drivers are for radar identification and IR search and track. The data requirements for the aircraft in these scenarios determine the amount of static and dynamic memory required. The access requirements determine the requirements for bus bandwidth and processor throughput.

The DBMS must then ensure that data used by the application programs is delivered accurately, in time, and
that security is enforced. The accuracy of the data must be ensured by providing integrity control, concurrency control, and fault recovery methods. The DBMS is required to provide data to applications in time by pulling data from slower mass storage devices and transferring it to faster global or local memory. Security must be enforced for DBMS data objects through a secure operating system, bus manager, and access functions. Object-oriented systems offer a clean interface for DBMS security because all system actions are accomplished by messages. The DBMS will be required to have a standard interface for applications and operating systems (Figure 2). In order for the benefits of software re-use and data independence to be realized by using a DBMS, avionics application program developers must be provided with a standard DBMS interface. This is especially important given the many different operating systems planned for advanced avionics systems. A standard DBMS interface will minimize the amount of integration software required when FIRM is implemented on different avionics platforms.

3 Critical Software Technologies

For FIRM to succeed it will need to consider and exploit several critical software technologies. The avionics environment will place high demands on performance, reliability, security, and interoperability of FIRM. Of first importance is performance. FIRM must operate in a real-time, fault-tolerant environment and so real-time database technology issues, including main-memory databases, concurrency control and recovery, and distributed architectures are important. FIRM must provide secure operations on data at different classification levels, and so multi-level secure database technology issues are important. Finally, FIRM must be extensible and easily maintainable to deal with legacy systems it must interface with on the aircraft and future systems which may require new access methods or storage techniques. We believe object-oriented technology will provide the framework for this interoperability. In the following we highlight the challenges associated with each of these technologies and its relationship to the successful implementation of FIRM.

3.1 Real-Time Database Technology

As part of the avionics software suite, FIRM must support the real-time, fault-tolerant requirements of the avionics platform. There are currently no commercial, off-the-shelf (COTS) database management systems that provide the real-time "hooks" needed to analyze and predict performance [5]. The lack of real-time predictability is the primary reason that, up till now, database technology has not been considered for use in real-time environments. There has been a lot of recent research into this issue, however, and the time appears to be right to build the FIRM real-time data manager.

Most of the research in real-time data management has concentrated on improving the performance of the database system, so that it is less likely to miss required real-time deadlines. There are three major areas of importance to FIRM: concurrency control and recovery, multi-level storage, and distributed databases.

3.1.1 Concurrency Control and Recovery

The most studied area involves modified concurrency control protocols. The proposed protocols use additional semantics of the application to provide relief from the strict requirement of a general transaction processor which enforces serializability. Ideas include the use of priorities and/or deadlines to better schedule real-time transactions [6–15], the use of multiversion protocols to avoid delays for transactions [16, 17], and the use of application semantics to allow classification and partitioning of the transactions into mutually interleavable sets [18–21] or to soften the strict enforcement of serializability [22, 23]. New disk scheduling algorithms to support responsiveness to deadlines have also been investigated [24, 16]. FIRM should be able to take advantage of this research since the hard real-time requirements (such as track file fusion) are well defined, and the semantics of these applications can be built into the FIRM transaction processing system.

3.1.2 Multi-level Storage

The second major area of importance to FIRM real-time data management is the issue of multi-level storage [25]. Traditional database systems typically have one level of storage, the disk. Secondary systems, such as additional disks or tapes may constitute an additional storage level
used for backup in case of failure. FIRM will require at least dual redundancy of critical data and data processing functions to meet survivability goals. However, in order to achieve the desired performance a third level, main memory, will need to be used as a database store. Main memory databases (MMDB) have been studied for several years in anticipation of lower cost, higher densities memory becoming available [26]. It is now quite feasible to create a large main memory system capable of supporting a reasonably sized database or subset of a database. MMDBs, however, require a different approach to many parts of the database system software. Recovery needs are different since main memory is volatile [27-31]; different query processing algorithms and access methods are required since reducing disk accesses is no longer the target of optimization [32, 33]; concurrency control protocols should be modified since conflicts can be resolved quickly and memory space is more limited [34]; buffer management is not required since the data is already in memory [30]. FIRM levies the additional requirement that we have both a main memory database and a disk resident database. Thus issues of coordination and data migration are also important.

3.1.3 Distributed Databases

The FIRM environment is a distributed, heterogeneous network of general and special purpose processors. The question for FIRM is: should FIRM be distributed as well or is a client-server approach more appropriate? A distributed database management system (DDDBMS) has several advantages including improved performance (since data is placed where it is needed), improved reliability/availability (since data can be replicated in a controlled manner), expandability (since new capabilities can be added to the network), and shareability (since existing systems may be otherwise unwilling to give up their data) [35]. Disadvantages include increased complexity especially when control is distributed, and increased security problems. Distribution of control is the key to our question. In the client-server approach control is centralized (Figure 3(a)). Requests for data go through the global data manager (GDM) or server, access plans are determined and passed to the appropriate local data managers (LDM) or clients for processing. In this approach LDM's have no local autonomy. In the fully distributed approach (Figure 3(b)), each site maintains local autonomy. A GDM is embedded at each site to coordinate global transactions, and an LDM runs all local and portions of global transactions running at its site. Full distribution has distinct advantages for FIRM considering the requirements for reliability and survivability, the performance requirements leading us to consider using multi-level storage, and coexistence with existing avionics systems.

3.2 Multi-Level Secure Database Technology

FIRM must operate in an environment where the operating system is multi-level secure and the stored data is multi-level. Several architectures for providing multi-level security (MLS) in a distributed DBMS are possible [4]: partitioned [36], replicated [37, 38], trusted computing base (TCB) subset [39], and trusted subject architectures. The partitioned and replicated architectures assume that untrusted back end DBMS's (usually COTS systems) are used. However, these approaches have additional overhead in the user front ends and/or the data manager which need to process queries and updates and enforce the database security policy. Under a TCB subset architecture the DBMS has some responsibility for enforcing the database security policy, however mandatory data separation and protection is provided by the operating system. Finally, with a trusted subject architecture even mandatory access controls are made part of what is now a DBMS TCB. This allows the DBMS to operate outside of the operating system security policy and permits it to enforce a distinct policy over objects at a different (usually finer) granularity than those protected under the operat-
ing system. The advantage of these later approaches is the increased performance that is possible. The disadvantage is the increased complexity of an MLS DBMS. The most extensive research on database security has been performed for the relational database model. Only recently has significant work concentrated on distributed databases and object-oriented models [40]. This research is important to FIRM since performance constraints lead us to the use of distributed database technology and the need for complex data types, reusability and maintainability lead us to the use of object-oriented technology.

3.3 Object-Oriented Technology

Object-oriented technology will be the glue with which the other complex technologies needed for the FIRM product will be put together. Object-orientation allows us to map real-world objects with their real-world constraints into a set of software objects which can be reliable, reusable, and maintainable. Ruberg, et al. [3] list the following advantages of object-oriented techniques in the design of database management functions for avionics systems:

- Ability to change mission specific data values or objects to the database without changing existing code in the operational flight program.
- Ability to add new functionality without duplicating needed data already in the database.
- Ability to add new data types.
- Reuse of data management software by multiple applications.

Object-orientation allows a consistent software development methodology to be used from analysis through implementation. Object-oriented analysis and design methodologies have been proposed for real-time systems [41-43]. These methodologies allow for the modeling of timing constraints within objects. The next Ada standard will include revisions that are intended to improve Ada's usefulness for real-time applications [44,45] and to move Ada closer to being a complete object-oriented language [46]. Object-oriented database (OODB) technology is now moving into the mainstream with several commercial implementations and efforts at standardization underway. Security issues for OODB models are being explored [47,48] with much of work for relational models carrying over. Object-oriented analysis, design, and implementation in an object-oriented language with an object-oriented database will provide substantial savings throughout the life cycle of the FIRM program.

4 Conclusion

The FIRM program is developing an avionics DBMS that will deal with the increasing information requirements on future military aircraft. DBMS software will be developed, integrated into laboratory equipment, and tested in real-time scenarios. FIRM will then be integrated into an avionics platform in support of an application much like the BAI scenario described earlier. Additionally, FIRM may be integrated into a variety of technology demonstration platforms in support of advanced data and information fusion, advanced pilot aiding, and future fighter/attack aircraft technology development.

A real-time distributed database management system with multi-level security, built using object-oriented technology, is a tall order for FIRM. Recent surveys of the current state of research and the open research issues in these areas [49-51] provide both reason to anticipate the successful near term application of these technologies but also the vast amount of work remaining to fully exploit them. FIRM will provide the impetus toward solving these problems for the avionics platform.

References


