REPRESENTATION OF THIS DOCUMENT AS IF YOU WERE READING IT NATURALLY

1 INTRODUCTION

While there are many kinds of real-time systems in use, in production, or planned that have requirements for real-time database functionality, there is little common understanding of the underlying requirements that lead to such implementations. This situation extends both to the implementation community which must produce these systems as well as to the research community seeking to understand the fundamental principles that describe their behavior.

This paper will describe five application domains that exhibit some form of real-time database requirements, attempting to provide an understanding of their consequent database functions in terms of their performance characteristics. Before discussing these domains, however, it is necessary to define the term real-time database for the purpose of this exposition:

A Real-Time Database consists of a data store whose operations execute within a user-specified predictable response time, and with application-acceptable levels of:

External consistency The data being stored represents the state of the real-world artifacts at any point in time (e.g., if the database contains an aircraft velocity of 425 knots, the actual aircraft velocity is 425±.01 knots.)

Temporal consistency Each data element represents the accurate state of the real-world artifacts at a point in time acceptably close to the time at which other related data elements are known to be accurate (e.g., if the
target latitude was measured at a time of 215.06218 seconds, the longitude was measured within 20\mu s of that time.

**Logical consistency** No data elements are visible to an application unless all logically dependent data elements are also visible from the same transaction (e.g., if a data element shows a simulated aircraft destroyed, another data element does not describe a subsequent attack by that aircraft.)

**Permanence** When a data element is changed, that data change will continue to be visible until it is changed again, or until no possible references to it will be made in the context of the system mission.

**Atomicity** Each transaction will either be totally completed (i.e., committed), or will have no effect at all (i.e., aborted). No transaction effects are visible partially completed (see Logical consistency).

For each of these application domains, their database requirement descriptions will concentrate on the level of their need for the first four of these five attributes. The fifth attribute, atomicity, is assumed to be present for all five.

For applications such as these, few commercial products are generally used. While there are a number of experimental tools being constructed and evaluated for research, these are not available commercially. Instead, most existing systems, as well as those being currently developed, use custom-designed implementations in an attempt to meet the timing requirements. Although this paper describes, in general terms, the timing requirements for these types of systems, few such systems are described in these terms in their formal requirements documentation. Because of this, and because of the fact that few implementers are familiar with the real-time database literature, the implementation is likely to use ad-hoc techniques.

The information provided in this paper does not specifically refer to any particular project or implementation, but rather reflects a composite of such systems, based on the author’s experience with a large number of such systems. The remainder of the paper will consist of a brief description of five specific application domains (Air Traffic Control, Aircraft Mission Control, Spacecraft Control, Training Simulation, and Process Control) in Sections 2 through 6, followed by several brief comments in Section 7.