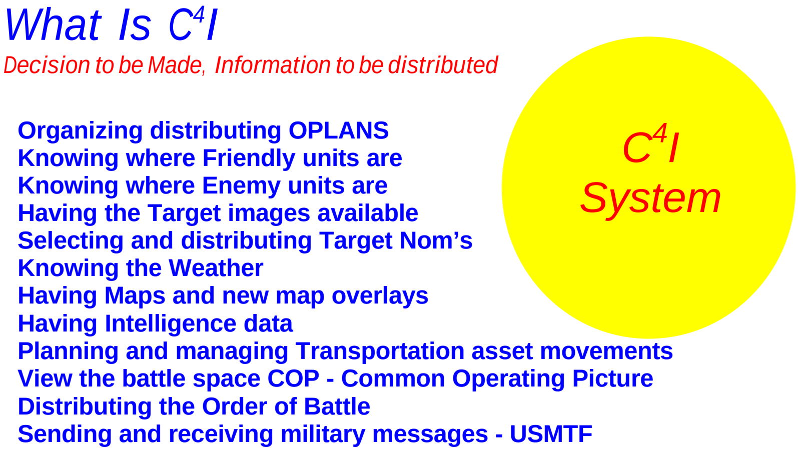
New Approaches C4I Data Warehousing for Crisis Action -- Synchronization, Replication, Sustainment Design Implications

Robert Babiskin
SAIC
8301 Greensboro Drive MS-5-2
McLean VA 22102
(703) 917-8415
robert.babiskin@saic.com

Abstract

C4IFTW (Command, Control, Communication and Intelligence For the Warrior) automated information systems (AIS) provide the planning and executing of military operations requiring mobilization, deployment, employment, sustainment, reconstitution, redeployment, and demobilization of U.S. Armed Forces. At the heart of C4IFTW AISs are highly distributed replicated decision support databases. These systems are often characterized as like a Walmart's data warehouse that goes to war, as the sensitivities to the decision support recognition of need-for-action and timeliness of decisions are acute and similar in approach.


What Is C⁴I
Decision to be Made, Information to be distributed
Organizing distributing OPLANS
Knowing where Friendly units are
Knowing where Enemy units are
Having the Target images available
Selecting and distributing Target Nom's
Knowing the Weather
Having Maps and new map overlays
Having Intelligence data
Planning and managing Transportation asset movements
View the battle space COP - Common Operating Picture
Distributing the Order of Battle
Sending and receiving military messages - USMTF
C⁴I System

This has lead to highly distributed replicating sustainable architecture to military data marts that dispense C4I *"information products"*. In fact, these may be among the largest replicating information system approach architected – pushing, pushing data to its evaluation point on secured networks. This paper will discuss design implications of the warehouse and mart building from the global virtual enterprise perspective of data manufacturer interfaces, database design movement implications, and from a user data accessibility perspective; all within the unique highly distributed data and communication architecture required from a warfighting perspective.

C4IFTW AISs creates a *shared* warfighting information apparatus that has a set of unique characteristics:

- Accurate global data awareness
- Autonomous operations and ability reattach and re-synchronize with the database (data docking)

- Specialized open-system architecture - Common Operating Environment (COE) – joint inter-operability
- Warfighting electronic commerce standards - military messaging

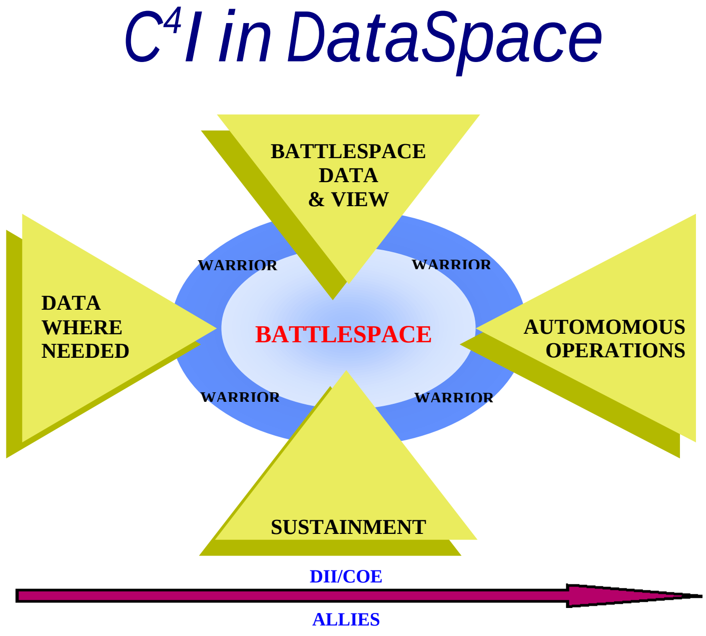
1. Overview

This Paper addresses database design and topology issue as applied to the C^4I automated Information Systems (AISs). These issues gravitate around database synchronization and application performance. There are potential significant improvements in the following areas:

- Database Synchronization: solutions resolve N-way replication degradations.
- Application Performance: system response times in a high data streams environment.
- Data Integrity: Consistent data timing – decision with the SAME data

2. Database Topology

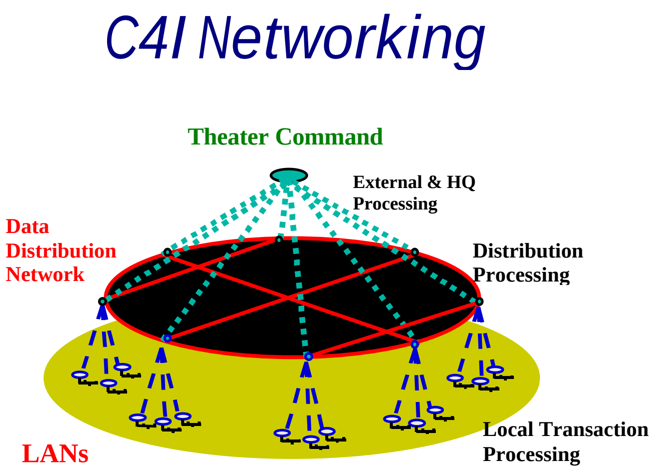
Problems with C4IFTW AISs and their associated databases have existed at least since the mid-1980. While the systems architecture has been improved achieving large financial savings for DoD; legacy issues in the mapping of business functions and the technical requirements have persisted in the database.



Most C4IFTW AISs have implemented asynchronous database updating, using a mesh topology for its data distribution and replication architecture. In a mesh topology, all sites are peers; data can be distributed to and owned by any of the sites. The topology is difficult to manage because there is little centralization of data. Network traffic is unorganized and may traverse unnecessary node hops to reach it destination. Mesh topologies are useful in uncommon cases where data has to be replicated in different domains. The current N-Way replication topology with *n* master sites and asynchronous database updates was put in place as the alternative to the two-phase commit to a remote single master. The two-phase commit ensures that every site in a distributed database always has the same data but relies more heavily on a network infrastructure that can support remote client-server environment. There is additional the sustainability issues related to the single master vs. the *n* master approach, where more replication masters are more sustainable. In the current asynchronous replicating update mode updates operate as a *push* to other sites with *no co*ordination.

This gives rise to:

(1) Geometric Replication Problems--- with *n* masters every local update requires a peer-to-peer connection to "n-1" receiving site creating "n-1" replications per update.
(2) Current database technology for database replication beyond a small number masters that are update masters is immature. Synchronization issues can be summarized as:



- Asynchronous Updates: This process essentially still allows out-of-sync databases due to delays, recoveries and locks
- Lost Updates: These problems occur when the owner of the data locks data cells at his/her site while users at other sites continue to update the data that was locked. This can happen because the lock notice has not arrived at the other sites yet. With the asynchronous replication capability, the users' updates are accepted at their database sites. However, the same transaction will be rejected at the site where the data had been locked. The data will now propagate unpredictably to other sites with copies of the data. This problem cannot be resolved technically as long as some transactions can bypass DBMS. It can only be resolved with policies and enforced procedures.
- Conflicting Business Rules: Situations occur that apply different rules to the same conditions. Data reengineering is essential to identify and resolve these.

3. Requirements

The design of a C^{4}I system has several overarching requirements that are the core design parameters. These are in the areas of *sustainability* and *replication*. *Sustainability* is where the system and the database must be available in the event of a system or communication failure, which includes autonomous operation on less than current data. *Replication* is where the database is duplicated at other sites and updates are coordinated and synchronized to each of the sites.

Sustainability Core Requirements:
- The system shall support continued autonomous operations in the event of a communications failure on prior-to-failure data.
- The system shall have a hot-standby database.
- The system shall support fault-tolerant raid devices.
- The system shall support fibre channel disk arrays

Replication Core Requirements:

- The system shall support n-Master synchronized databases with distributed network updating.
- The system shall support up to of 70 Master synchronized databases.
- The system shall support partitioning of the database rows among the Masters with individual replicating schemes.
- The system shall support conflict resolution of competing updates.
- The system shall support notification of the owner of a backed-out update in a conflict resolution.
- The system shall support a high transaction replication stream of 40-100 TPS.
- The system shall be manageable from a robust system management remote base.
- The system shall support local database subsets of the master database. Local database are updated only locally and not replicated.
- The system shall allow background updates.
- The system shall allow updates to be held for update until authorized.

While performance improved in certain areas (e.g. the transaction queue processed faster), those improvements often created problems in other areas (data got out-of-synchronization quicker). For example, it took three days to clear the transaction queues in WWMCCS. GCCS can clear them in hours. But, as no changes were made to the transaction posting process the result is that the same out-of-sync condition happens more rapidly. Where new requirements were implemented during the legacy port, such as the provision for asynchronous database updates, they complicated the "out-of-synchronization quicker" problem as a by-product.

4. Approach

4.1 Data is the Core

All data is not equal. Some data is fresher and more accurate. The process of long going systems is keep data that is required for events and to ensure it sourcing is from the most precise and consistent AIS available:

- Identify and eliminate redundant data.
- Remove data no longer required.
- Identify required data and develop automated links to better data.

Four areas that require design attention in order to evolve the C^4I systems and databases for more rapid mobilization to *crisis actions* include:

- Design Parameter 1:
  Database Topology -- the number sites related to geometrically synchronization issues of replicating updates to other Masters.
- Design Parameter 2:
  Logical Database -- Start with the Data Model reduce data holdings to the minimum essential. Identify targets of opportunity for deletion/consolidation of obsolete and/or redundant Data:
    - *Data Redundancy and Data Integrity*
    - *"Best" Data Sources*
    - *Lower level detailed data mapping*
    - *Obsolete Objects*
    - Object Coalescence

- Design Parameter 3:
  Physical Database – Star-schema vs. normalization, de-normalized table validity; Model Database Changes and Determine the Impact on Data Integrity and Data Consistency.
- Design Parameter 4:
  Housecleaning -- issues associated access permissions, the number of active Rows (holdings), and the restructuring for more efficiencies of huge resource consumers (batch changes) that run concurrently with the online transactions.

4.2 Alternatives

The three topologies are:

(1). N-master. This topology consists of the current multiple masters, and asynchronous updates with the addition policies and operation parameters that enable more precise control of the remaining causes of synchronization issues.

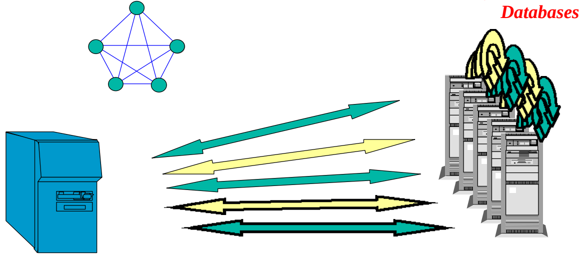
(2). Two Master Topology. This topology consists of only two masters for any given C^4I database:

- A master copy of each database is maintained at the data 'owners' database site, usually the supported CINCS.
- One other master will be replicated at one other database site. Snapshots of the master will be sent to non-master sites to support report generation and query needs.
- Replicating would be employed to maintain up-to-date and accurate at the second master site.

(3). Reduced N Master: This topology consolidates to a relatively small number of database reducing the amount of the synchronization traffic.

- This topology would have 3-5 sites
- Some users would operate across the network in a client server approach
- Replicated copies of the database would be at each master site

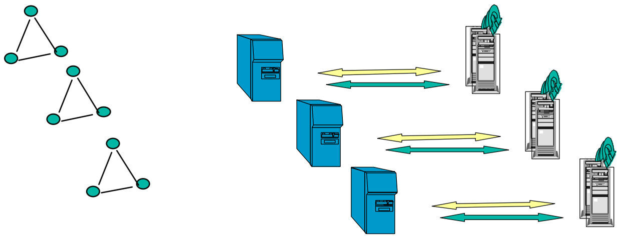
N-Way Replication



Replicate all updates everywhere
There can be 70 databases

Synchronized Databases

Fragmented Replication



Replicate all updates within Fragments
There can be 3-4 Databases
A lesser Geometric Problem

Synchronized Databases

The alternative are broken down to the following 6 technical schemes.

Technique	**Description**
Asynchronous	n-master where n = all nodes
Master-Slave—star or Less than n-masters Multi-branch	Update to a single or a few nodes (multi-branch) and replicated dynamically to remaining (1-way replication, <n-masters. Replicates are read only. Topology based fragmented. *Objective – reduce traffic, reduce out of sync conditions.*
Publish/sync point synchronization vs. dynamic synchronization	Similar to master-slave but with a global sync point at a specified interval publish time for replicated batches to occur. *Objective – reduce traffic, further reduce out of sync condition*
Hierarchical Replication	Non-geometric dynamic replication (where n-way is 1 replication sent 1 time). *Objective – further reduce traffic, reduce out of sync conditions*
Single Master with Functional Fragmentation	Variation of Multi-branch functionally fragmented. *Objective – reduce traffic, reduce out of sync conditions.*

4.3 Design Ramification

Design ramifications of a replication schema permeated through out the system architecture and is limited by design constraints hardware/software, military standards for interoperability/infrastructure. All of which must be evaluated in design alternatives and include:

(a) Technical Performance Analysis: replication alternatives in terms of synchronization measures; throughput, data integrity, and, data consistency
(b) Implementation Analysis: GOTS/COTS/COE products.
(c) Operational Analysis: Conduct network sizing and system sizing engineering on system components in terms of:

- Hardware implications
- Bandwidth implications
- Operational readiness implications
- Autonomous operations implementations
- Optimal number of sites
- System Management implementations
- Cutover implications
- Sustainability implications
- Schedule

4.4 Sample Analysis – Case Study

A 16 node 2-tier client server system was examined. Replication was nearly n-ways averaging approximately 12 way replication. The design architecture was optimized at the application processing level with replicated database on the local LAN. An application server was also on the LAN and it would cash specific data that was being requested.. This architecture provided good response time after significant load and setup time. An alternate architecture was examined and is being build that focused on a 3-tier WEB-base architecture.

As the replication traffic volume caused the congestion in the prior architecture a 4-node database solution was put into place with application server co-resident with the database server. This eliminated the database traffic and minimized the replication traffic. The 16 client nodes operated as WEB client shipping and receiving WEB pages. The results where:

Overall Traffic from the original 16-database nodes (2-tier) to 4 database nodes (3-tiers) was reduced by 55%.

Migrating to 4-database node (2-tier) required 2300% more bandwidth than the 4-database nodes (3-tier).

In both cases there remained 16 client hubs on a LAN with a users attached. In the hub reduction, sufficient bandwidth was programmed for.

5. Conclusions

The current n-way, n-master database replication scheme using asynchronous updating was the best solution available at the time. However, database synchronization issues continue to plague various C^4I AISs. The redesign approaches presented in this Paper provide the engineering process to resolve these long-standing problems database synchronization issues. They form a system-level approach for analyzing the complexities in C^4I databases.

Biography

Mr. Babiskin has been a noted technologist for over 25 years and have been involved in the design of warehouses and very large data systems at DoD and Civil Agencies that have been of a terabyte size or with wide – distributing patterns. Mr. Babiskin holds degrees in computer science, an MBA and completed in doctoral work in computer science and operations research. Mr. Babiskin has been senior executive or senior technologist at such firms as Oracle, Computer Science Corporation, GTE, and GE.