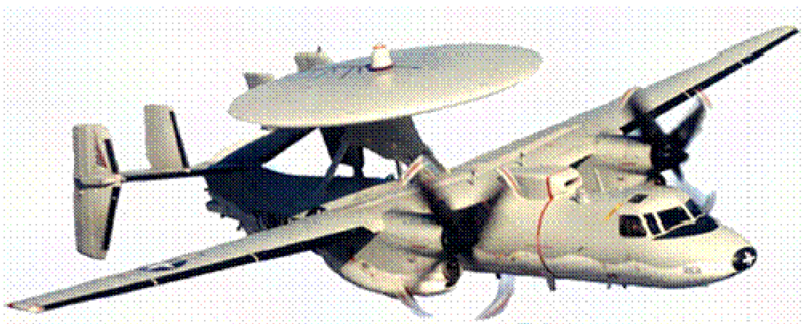
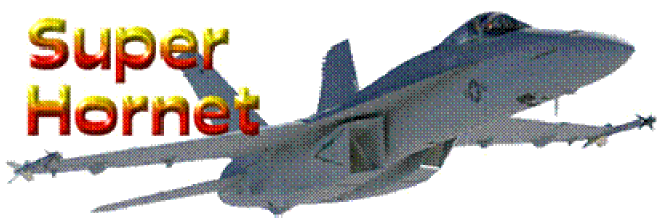
Document Number SBSI2005-12	Version: 4

Next Generation Fusion Architecture

Small Business Innovative Research (SBIR)

Final Report: Phase I and Option





Submitted by:

SILVER BULLET SOLUTIONS, INC.

30 April 2005

Executive Summary

This project describes the creation of a Next Generation Fusion Architecture, an open information architecture, for Command and Control (C2), and Weapons Control systems that require advanced sensor and data fusion. This Next Generation Fusion Architecture provides a foundation for advanced fusion algorithms including non-kinematic level 1 fusion, level 2 and 3 complex assessments, more broadly scoped Situation Awareness and Battle Management information analysis, and level 4 process adaptation. The architecture supports increased automation and higher quality data fusion through enforced integration and integrity of data – thus allowing advanced mechanisms, such as ontology-based inference, as well as the ability to execute multiple kinds of fusion algorithms that interoperate autonomously, yet synergistically.

In the body of this document, we describe an open architecture for Advanced Data Fusion that builds on top of the existing Navy Open Architecture. Such an architecture is now recognized as necessary for advancing Data Fusion. We are inserting a formal & rigorous approach to data modeling into the mission system that will broaden AHE interoperability, support plug-and-play fusion algorithms, and structure reference and intelligence databases so they can be used by fusion algorithms. We have matured, tested, and validated the technology - and we are ready to implement it now. We have already done experiments with heavy sensor loads and tracking, correlation, and ELINT algorithms. We've been building towards this for years, on major systems such as ACDS / SSDS, in work for OSD and SECNAV, in SBIR's, and in IRaD. The approach is open-ended, and provides the foundation for a future technology side we are also working on -- for knowledge-level semi-automated fusion and inference. The NGC AHE team agrees and is eager to work with us.

The motivation for open architecture fusion is that all current approaches to fusion have reached a "sound barrier" that prohibits further advances in fusion much beyond tracker and correlator technologies originally developed in the 1970's and 80's. Those advances were the result primarily of mathematical achievements (e.g., the Kalman filter) and signal processing (e.g., CFAR detectors). Advancements to non-kinematic level 1 fusion, such as multi-source identification, have not been as rapid or successful as would be expected (see, for example, [62]) given the growing fusion community, and the explosive growth of computing technology. Advances in non-kinematic fusion are <u>decreasingly</u> limited by processing speed and power, yet <u>increasingly</u> limited instead by scale, integration, and interoperability issues. For instance, the complexity of the input data for level 1 target ID processing is staggering -- a priori sources are difficult to manage, to groom for automated processing, and to account for in a mathematically rigorous algorithm set. This is even truer for higher levels of fusion. One solution to this is to make it easier to plug in new algorithms that access more information and more types of information. Our ontologic fusion architecture does just that, as the experiments conducted in this Phase 1 indicated.

SBSI staff experience with advanced fusion has convinced us that the problem is the lack of a comprehensive, consistent, and open architecture – especially the data layer. A fundamental difference between advanced fusion (such as Multi Source Identification or MSID) and kinematic fusion (such as tracking) is the number of inter-related data elements required. SBSI personnel worked on MSID problems as early as 1983, as part of the NAVSEASYSCOM Advanced Sensor Integration (ASI) and Tactical Distributed Processing (TDP) 62 research projects. Embedded computing and data management technology throughout the 1990's proved insufficient to this task causing many advanced fusion projects to be dramatically reduced or abandoned. At the same time, promising results from research have hit another

barrier – the integration barrier. Even a seemingly good result from a fusion experiment may be too difficult to integrate into the existing BMC2 or C4ISR system because the existing architecture so tightly weds the embedded data structures to the algorithms and using applications. A way to decouple the algorithms from the data, and to make the data "public" is needed.

With the clear understanding of what was stalling fusion progress in deployed systems, SBSI staff began looking for solutions to the fusion "sound barrier".

- At the International Information and Sensor Fusion (ISIF) conference in 2002, we presented the theory of ontology-based fusion as method for achieving interoperability in many independently developed "fuselets." Now, one can scarcely attend a fusion conference without a major emphasis on ontologies. Unfortunately, ontology is often confused with data modeling, and, even then, the fusion community is stumped on how to proceed. SBSI's continuing work is providing the only viable pathway to effective and efficient integration in fusion.
- The ontology-based fusion theory led to further interest in the possibility of implementation. SBSI performed experiments on the feasibility of an embedded DBMS supporting fusion ontology. We reported the results at the National Symposium on Sensors and Data Fusion (NSSDF) in 2003. Those experiments showed embedded DBMS and computing technology had advanced to the point that they could keep up with a very intense fusion data access load, one based on Navy Combat System stress tests. Those experiments are summarized in paragraph 3.1.
- Under Phase I of the SBIR, SBSI worked on formalizing the theory and performing the next round of feasibility experiments. Since we were satisfied with the loading results from our 2003 research, we wanted to know what was involved in conforming fusion software to the open architecture environment. We did this with limited experiments, using actual tracker and assignment algorithms from previous projects, a "1090" TPX-42 ATC tracker, Yakov Bar-Shalom's Dynaest library Kalman, Oliver Drummond's JVC, and the SSDS Mk-2 "Y-NOT" correlator. These experiments are described in paragraph 2.3 and in detail in Appendix B.

This project continues to advance both the theory and implementation of this exciting and promising new architecture. Now that we are satisfied with the loading of the implemented ontology, and have significant insight into rehosting actual fusion algorithms within the architecture, we need to experiment with integrating the AHE mission environment, and conforming existing and emerging AHE fusion software to the architecture (e.g., the MSI, CID, and ATO correlator software), Concurrently, we propose to continue developing the logic and mathematics of ontologic inference networks, so the architecture will be ready to support even more advanced fusion levels in the future.

This project addresses several situational awareness and data fusion problems that have emerged:

- Fractured data: Data and information coming from diverse sensors and sources is difficult to combine because of the diversity of storage media, file formats, communication protocols, and technical interfaces. The first step in machine-supported fusion is getting information into a standard format on a common device.
- Stand-alone and non-interoperable algorithms: A bewildering array of filtering algorithms is available for addressing various classes of Level 1 tracking problems. Given a clearly defined multiple target tracking situation, the fusion community is relatively good at

identifying a small set of algorithms that should be tried, but currently we cannot enable users to easily swap in alternative algorithms for varying situations.

- Isolated and non-interoperable sub-specialties The fusion community currently fails to address fusion across technical sub-specialties: There are significant problems with combining increasingly useful ATR image data fusion products with 3-D data about sparse point-objects located in space.
- Inadequate prior knowledge utilization: Good analysts make use of enormous amounts of prior knowledge when trying to understand current information from diverse sensors and sources. In order for automated systems to support utilization of prior knowledge, that information must be available to the system in a standard, machine-readable format
- Information scope brittleness and shortfalls. Most fusion algorithms, and virtually all applied fusion algorithms, deal with a very narrow scope of information. For example, if an entity can know and reason about only a few kinematics parameters, many – if not most – kinds of logical inferences become impossible. Powerful inferential reasoning generally builds on widely divergent types of information that combine to generate a consistent understanding of the situation.
- Insufficient exploitation of weak and indirect evidence. In many cases, an analyst's assessment of a situation is not based on one single definitive piece of information, but on an overarching assessment of many smaller indicators, any one of which would be inconclusive – and many of which may be quite indirect. In order for automated systems to exploit weak and indirect evidence, they must recognize and accumulate relevant bits of information, and update assessed probabilities based on the evolving weight of evidence.

Though the operational consequences of system shortfalls have largely been overcome by improved human processes, this is not a long-term solution, particularly as more complex and unplanned missions come to fore, as is expected by National strategy and planning. Further, humans in the system now constitute bottlenecks, given that much more information is available than humans can input and process in a timely manner.

There are also technical constraints in the ability to create complete solutions to these problems, the greatest being the limits of intelligent computing. Despite the amazing advances in computing speed and capacities, and some strong examples of artificial intelligence applied in manufacturing and business, intelligent computing for military data fusion remains in its infancy.

Important developments that have enabled this project are processing speed and capacity, which now facilitate the ability to run real-time fusion algorithms against a managed database instead of flat track files. We recognize this advancement as an enormous development for fusion systems and consider it the primary contributor to addressing the problems described above. Real-time fusion algorithms allow ontology-based fusion to have an ontology foundation, just as in human reasoning. We conducted load experiments in 2003, followed by actual algorithm implants in 2004, and have found this technology to be mature for use in fusion systems.

In the area of intelligent computing, an essential development for this project has been the technique of inference networks. Inference network techniques provide a way to cope with massive amounts of interrelated variables via the Markov construction of the network that explicates probabilistic and causal dependencies. Between the class structure, properties, and inter-relationships of the ontology and the casual and correlated representations in the inference network, there is much opportunity for advances in intelligent computing. Under

Phase I of this SBIR, we have furthered inference execution formalization within an ontologic structure, and have advanced mathematical techniques for operations with large networks.

Although there is still much to do in advancing a general theory of ontology and inference, immediate benefits can be gained from the ontology running in a real-time DBMS, and providing an integrated open architecture for multi-algorithm interoperation. Our objectives for Phase II are to continue advancing the general theory while, in parallel, stepping from our real-time fusion DBMS test bed using limited fusion algorithms, to the NAVAIR test bed using a complete fusion algorithms suite and priors data sources. This is the next logical progression from our Phase I experiments and analyses, where we demonstrated that off-the-shelf fusion algorithms can be plugged-and-played against the ontology (given wrappers) and that the ontology could support real-time, uncertainty inherent in fusion processes, and interfacing with fielded data sources such as TADIL-J. The next step from the NAVAIR test bed will be testing in a Fleet and then Joint experiment, followed in Phase III with technology transition into the production systems. This will involve the rehosting of production system software to reside on top of the ontology, and the formalization of new software for certification and life-cycle support.

The mission importance of this project is that it will provide the path for evolution of fusion systems for decades, as algorithms are developed and inserted into the ontology and as the theory of massive, indirect, and weak inference becomes more mature. This will provide very near term benefits and well as long-term evolvability. It will support the rapid adaptation of fusion systems to emergent, and future, mission requirements.

It is universally recognized that a comprehensive, open architecture is necessary for advancing Data Fusion into the next generation, by inserting a formal & rigorous approach to data modeling into mission systems. We have the experience and capability to fully develop an open architecture for Advanced Data Fusion to support the full range of JDL fusion levels. In order to assure interoperability across the spectrum of potentially contributing systems, it will be important to build on top of – and maintain rigorous consistency with - the existing Navy Open Architecture. The Advanced Data Fusion architecture will broaden AHE interoperability, support plug-and-play fusion algorithms, and enable us to structure reference and intelligence databases so fusion algorithms can use them natively. In Phase 1 we conducted successful experiments with heavy sensor loads and tracking, correlation, and ELINT algorithms. These culminate years of theoretical work, and demonstrate that we have matured the required technology and we are ready to implement it now. We've been building towards this for years, on major systems such as ACDS / SSDS, in work for OSD and SECNAV, in SBIR's, and in IRaD. The architecture we envision will be open ended – and will lay the foundation for ongoing evolutionary change to fusion levels 0 and 1, as well as revolutionary contributions to fusion levels 2 through 4 – including case-based reasoning, ontology-based fusion, and knowledge-based inference. Near the end of our Phase 1 effort, we met for 8 hours with 12 key members of the NGC AHE technical team, and achieved unanimous agreement that our proposed Phase 2 approach can succeed and the products are necessary to achieve AHE goals and NGC plans.

The NGC AHE technical team agreed that our proposed Open Architecture for Advanced Data Fusion carries both (very low risk) short-term benefits and (medium risk) long-term benefits. The approach supports both inter-platform data interoperability and intra-platform plug-and-play for fusion algorithms (short term payoff). It enables native access to diverse reference databases that will immediately support human analysis and eventually support machine analysis (medium term payoff). It supports knowledge-based extensions (e.g., decision support, ontological inference, expert systems) and enables experimentation with these (long term payoff) so that these new and important fusion capabilities can be refined into practical realities. The approach builds on top of existing Navy Open Architecture, but constitutes a significant and necessary extension in the area of Data Fusion that is based on 25 years of experience in the both the research and operational Data Fusion communities.

The time is ripe for next generation data fusion. SBSI has a great deal of experience in this area, and we have initiated development of a data fusion open architecture that builds on and extends the Navy Open Architecture to facilitate and enable next generation data fusion. Our data fusion open architecture will allow plug-and-play of fusion algorithms - for development, experimentation, and in practice – to support all JDL fusion levels. Current algorithms at levels 0 and 1 (based on signal processing, statistical, and associative methods) are relatively mature but can be further matured, could benefit from a wider diversity of source data, and can be made easier to mix and match. Current algorithms at JDL levels 2 through 4 (based on decision support, case based, and knowledge-based methods) are less mature. Our architecture supports all these kinds of methods, and allows them to interact with each other across fusion levels. To support the higher levels, systems will need access to various reference databases. We know what many of these reference databases are, we know what must be done to prepare them for machine processing, and we have done this before. We understand the requirements for the Advanced Hawkeye, we understand where we fit in to the plans, and we can work well with the system integrators at NGC. We would very much appreciate the opportunity to be a part of the team that brings it to the AHE community. We understand the requirements for the Advanced Hawkeye, we understand where we fit in to the plans, and we can work well with the system integrators at NGC

Table of Contents

List of Tables

List of Figures

1 UNDERSTANDING OF THE REQUIREMENTS FOR ADVANCED HAWKEYE NEXT GENERATION FUSION ARCHITECTURE

The system and the software to which this document applies is ATDS software, build 9, and beyond, with IOC projected in GFY010. The Advanced Hawkeye will give provide greater threat detection capabilities over land and water, with greater range and precision than current systems. It is also intended to be the foundation for the Navy's theater air missile defense function. New communications systems are designed to make it a major node in the Navy's FORCEnet information/decisions grid, enabling it to provide and integrate key information and surveillance data, fuse decision data and provide forward control and communications capabilities. The system will provide the enhanced airborne C2 and expanded surveillance umbrella that are required for SeaPower 21. The new platform is central to the Navy's role in future military strategy.

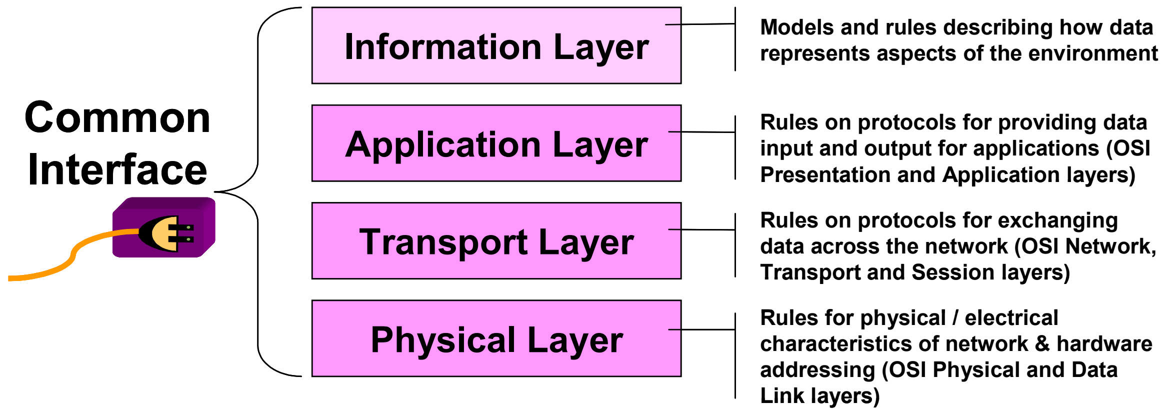


Figure 1. JBMC2 Common Interface Layers

The scope of this architecture, in the context of Navy Open Architecture, is what Joint Battle Management Command and Control (JBMC2) calls the “information layer,” as illustrated by Figure 1, taken from [39].

Navy Open Architecture specifications and guidance deal with the physical, transport, and application layers [57], in particular with respect to data, and with OMG middleware such as DDS. The scope of this project is creation of an information layer, rigorously structured into an ontology layer and, beyond that, additional structure, and mechanisms for that ontology layer to provide a foundation as an inference network.

1.1 Missions and Mission Environments

The overriding imperative in designing new, more robust methods of data fusion is driven by significant changes in the composition, command and control, and speed of events in the modern battlefield. The ‘traditional’ role of the navy has changed as ‘Joint’ operations have matured. No longer does the Navy ‘go to war’ on its own. Instead, Joint Operations, Joint Task Forces, and Joint Planning & Execution decide the eventual mix in any given scenario, with the Naval Forces providing execution of their assigned roles, often within a broader context that includes other service elements in similar or complementary roles.

Additionally, operations are now stated in terms of hours, or even minutes of preparation depending on the assigned target of opportunity. That target may be identified by one service element, passed along in the intelligence chain by another, and executed by yet another branch. Thus, data—clear, unambiguous data—become critical at all levels, and especially in real-time operations. There is no time for extensive translation and analysis; the data must be pushed toward those subscribers who have need for it, even in the circumstance that the data is not 'perfect' or that the data has not been pre-cleansed to remove unneeded artifacts.

In this age of joint application of force to given situations, data fusion at all levels becomes critical to success. Missions and mission environments are expressed first in terms of the capabilities that must be brought to bear to ensure the enemy does not succeed. Capabilities, in turn are executed through assets available on the battlefield that, given appropriate mission data, can contribute to mission execution. Mission data is provided through interfaces that can receive data, or channel data to recipients in a manner that is both acceptable and understandable.

Many of the legacy systems that exist in the Navy, and throughout DOD and the Coalition partners in given actions was built to serve specific purposes, some of which have been subsumed into larger capabilities, such as Time-sensitive Targeting (TST), Joint Forcible Entry Operations (JFEO), Common Operating Picture (COP) or Integrated Logistics (IL). These joint capabilities, as they are defined in the Joint Battle Management Command and Control Roadmap, are sets of assets, present and future, clustered in development and execution, to provide the level of response desired. Assets can be from any Military Service, and may vary over time in the intensity of use, but rely on the ability to use and understand data from disparate uses to execute the mission. Importantly, it is expected in this changing environment that there will be needs for individual applications and systems, closely integrated systems of systems (SOS), and more loosely associated families of systems that share some interoperability, but not all—relying on translation for validation of data to be used.

This study suggests that there are varying levels of data fusion that will be required. These are discussed below.

1.2 Fusion

Data fusion is a process for associating, correlating and combining data, information, and knowledge from multiple sources to achieve refined position and identity estimates of entities in a battlespace; and complete and timely assessments of the significance of those entities in terms of the overall tactical or strategic situation and specific threat relations among entities at varying levels of aggregation. The process should be characterized by continuous refinements of estimates and assessments, and by ongoing evaluation of the need for additional sources or modification of the process itself to achieve improved results.

Fusion Level	Association Process	Estimation	Entity Estimation
L.0 Sub-Object Assessment L.1 Object Assessment	Assignment	Detection Attribution	Signal Physical Object
L.2 Situation Assessment L.3 Impact Assessment	Aggregation	Relation Plan Interaction	Aggregation Effect (Situation given Plans)
L.4 Process Refinement	Planning	(Control)	(Action)

Figure 2. JDL Fusion Levels and Main Related Processes and Estimates

- Fusion Level 0: Sub-Object Data Assessment. Sub-Object Data Assessment is the estimation and prediction of signal/feature states on the basis of pixel/signal level data association and characterization.
- Fusion Level 1: Object Assessment. Object Assessment is the estimation and prediction of the states of different entities based on observation-to-track association, continuous state estimation (e.g., kinematics) and discrete state estimation (e.g., target type and ID).
- Fusion Level 2: Situation Assessment. Situation Assessment is the estimation, inference, and prediction of relations among entities to include force structure and cross force relations, communications and perceptual influences, and physical context.
- Fusion Level 3: Impact Assessment. Impact Assessment is the estimation and prediction of the effects on situations of planned or predicted actions by the participants; to include interactions between action plans of multiple players, and assessed susceptibilities and vulnerabilities to possible threat actions given one's own planned actions.
- Fusion Level 4: Process Refinement. Process Refinement, related to resource management, is the adaptation of data acquisition systems, methods, and processing to support mission requirements based on recognition of evolving mission needs.

The key to successful fusion is to move from large amounts of initially uncorrelated and uncertain data - to a smaller set of non-redundant information with known probability parameters - to a cogent body of knowledge allowing commanders to make decisions. In trying to achieve this seamless flow to the commander, intelligence stovepipes and JDL Fusion levels can both become artificial restrictions to the flow. While there is clearly a "fusion vector" up and down which information flows, the categories of activities along the vector are not clearly demarcated. The lines between intelligence analysis and decision support, or between situation assessment and threat analysis, are neither clear nor useful. The key to our approach is the implementation of a powerful and general information architecture that will allow individuals and applications to apply varying technologies at various places along the fusion vector – and then generate results that can be easily shared with other individuals and applications.

As one moves along the fusion vector from object analysis toward impact analysis the applicable techniques predictable range from more mathematical/statistically-based to more cognitive/knowledge-based. At the object analysis end we apply signal processing techniques and then estimation techniques such as Bayesian Nets, Maximum A Posteriori Probability (e.g., Kalman Filters, Bayesian), and Evidential Reasoning). Higher levels of inference require decision level techniques such as Neural Nets, Cluster Algorithms, or Fuzzy Logic. Even higher levels of inference require knowledge-based techniques such as Expert System, Scripts, Frames, Templates, Case-based Reasoning, or Genetic Algorithms. Clearly as we move up to higher levels of inference we encounter less mature technologies. The only way we can efficiently experiment with and mature these technologies is by implementing an architecture that makes it easy to share data across fusion levels and to swap in various algorithms for analysis.

1.3 Tracking Database Operations

Tracking databases typically must perform:

a. New. Occurs when new objects enter the surveillance region, when objects that are in the region are newly detected, or when objects are created in the region such as a missile launch. An aberrancy is to create a new track that is false. New track operations involve assigning reference numbers and initializing object attributes. Lack of DBMS services, such as identifier assignment and management - along with business rules preventing such problems as identifier collision - have caused BMC2 errors.

b. Drop. Occurs when objects leave the region, when objects are no longer detected, or when they cease to exist (e.g., are destroyed), or join back up with a main object. An aberrancy is to drop a track that is still in the region and still have BMC2 interest. Drop operations must follow certain business rules (e.g., STANAG 6016 prohibits dropping engaged tracks) and must be complete. Many BMC2 errors are caused by incomplete dropping of tracks due to the unavailability of DBMS services such as cascaded deletes and synchronization.

c. Update. Occurs when new measurements or other information about the object is made available (e.g., published).

d. Correlate and / or Merge. Two tracks are now realized to be the same object. This can be the result of an aberrancy correction or can be the result of multiple reporting nodes. In the latter case, aberrancies are to correlate two tracks that are actually difference objects (false correlation) or to believe that two tracks represent two distinct objects when in fact there is only one (dual designation). The database operation is complicated because it involves a drop of one track and merging of its data into the kept track. Which data to keep can be an issue and in some cases may involve complex merging (e.g., LMS). In some cases, a track cannot be dropped and a link has to be made. Lack of DBMS services to manage links has been a source of many BMC2 errors.

e. Associate. Can mean a measurement is caused by an object, so the measurement is associated with the object. This involves either maintenance of a link or incorporation of the measurement data into the track.

f. Pair. A mission linkage, e.g., weapon-target pairing, flight leader / wingman relationship, etc. Again a linkage is maintained that DBMS services could do well.

1.4 Track and Reference Files

In many Command and Control systems, the track file is a flat file (or set of flat files) that references the intelligence files to infer identification (class, type, allegiance, nationality) and associations (e.g., 3rd party targeting). In the case of GCCS, the full MIDB, EWIR, and other S&TI databases reside in a COTS DBMS. The architectures, in simplified form, are as shown in Figure 3. The problem with the architecture was noted by GCCS developers [34][24] and by the proposed associate investigator during research into Bayesian networks for ESM/ELINT fusion [46], namely, that the track file and the intelligence database often refer to the same objects and their attributes. It is simple and natural for a human fusion expert to see corresponding elements from Intel and track databases as imperfect reflections of a single entity – but viewed from two perspectives. It is not straightforward for traditional fusion software to make this sort of abstraction-derived inference. The lack of an abstraction layer causes data integrity degradation and a convoluted fusion architecture. In [[46]], a remedy was researched where massive RAM was used to create a set of track file structures large enough to hold all of a theater OOB and C&P (around 100,000 tracks), basically treating OOB as massive priors that had been surveilled some time in the past and that were awaiting discovery by current sensors. While this approach was somewhat effective, a more elegant solution is now possible - using ontologies to support machine abstraction.

1.5 Track and Reference File Load Levels and Access Times

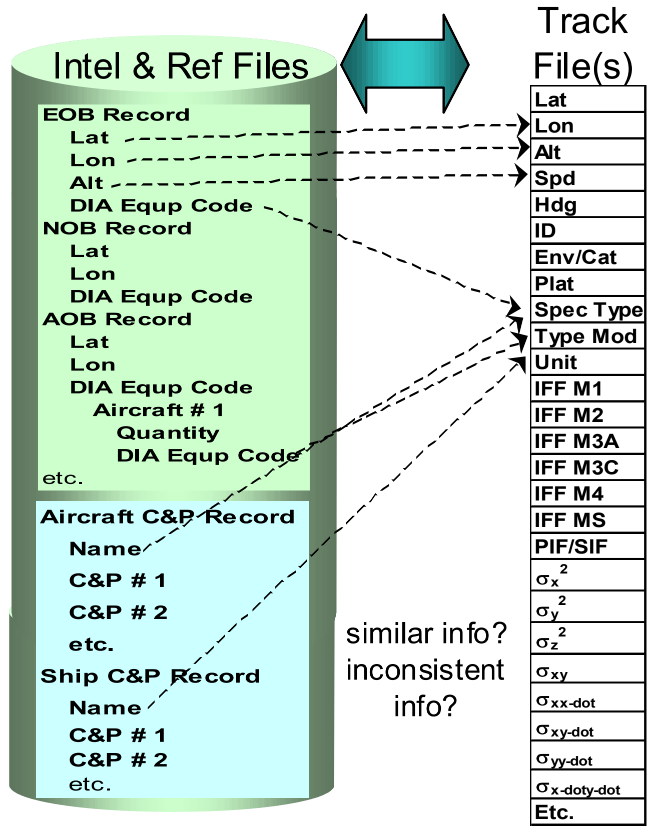


Figure 3. Conventional C2 System Data Architecture

Data access demands can be very high for fusion processes, particularly those defined as Levels 1-3 fusion. For purposes of this proposal, “high-volume data access” means that the volume of data access is large enough that the data access time could cause the input transaction job or processing queue to exceed operational requirements and that special design consideration must be given to either special data access techniques (e.g., hashing) or managing the job queue (e.g., pruning).

At level 1, it is often necessary to access many association or correlation candidates for goodness-of-fit testing. In a system with 2000 track file capacity, hundreds may fall within an input track report in dense areas. This happens often because for applications such as air traffic control, tracks are clustered in airways and around cities and airports; they are not uniformly distributed across the surveillance area. Even in currently deployed systems, the design must account for thousands of accesses per second [47]. At level 1, it is also often necessary to access many archetypes and currently known instances for target identification processing. For example, in a theater-level system, hundreds archetypes and instances can be required to be accessed per second [46]. Level 2 and 3 fusion processes can also have high access demands as reference and track files are referenced to discern patterns that could lead to level 2 and 3 knowledge.

Because of the high-volume data access for these types of fusion processes, the data must be maintained in computer RAM in applications-dependent data structures and accessed with special hash and search algorithms. Early radar trackers used hash by target range and bearing. As the fusion systems evolve to higher level fusion and the need to input and reference more types of data in ever greater quantities, the use of Data Base Management Systems (DBMS) such as Oracle or Microsoft’s SQL Server offers many benefits. However, these have been out of the question for fusion applications primarily because a single disk access can greatly exceed the entire timing budget for accessing all candidates. Because of this performance limitation, fusion system developers must build their own data access and handling software.

"Real-time" is often used synonymously with what might be called "fast-time", meaning, the processing is done quickly, or more precisely, within some allotted time period [81]. This differs from a more purist definition of real-time computing that would address determinism, that is, that ability of the computing system to respond assuredly at a specified time, to a specific level of precision less than a millisecond. For Levels 1-4 fusion there is rarely a requirement for this type of real-time; rather, 'fast-time' is usually what is required. An example of several-thousand track fusion systems running in non-real-time, or "general purpose", operating systems are the US Navy's Multi-input Tracking and Control System. It receives radar contact reports from 30 radars and track reports from whomever is participating in various tactical networks, tracking the contact reports in a adaptive tracker, correlating all the independent input sources, and estimating track kinematics.

In many respects, real-time DBMS' often are fast-time DBMS' by the deterministic criterion. Since this is the case, we propose to use fast-time, or 'embedded' DBMS that operate interactively with the disk drive for data access.

The features of an embedded DBMS are simply that, (a) data accesses do not require disk operations, (b) typical DBMS functions are available such as responsive to the Entity-Relationship model, access via SQL and extensions such as Procedural SQL or "record sets", background archiving and backup, and so on. In addition, it is beneficial if the SQL and other data operations software have been optimized for RAM operation rather than disk data access; ordinary DBMS' have been optimized for disk data access.

1.6 Software Open Architecture

OSD is driving the Open systems Joint Task Force (OSJTF) towards establishing in DoD an open systems approach as the foundation for all weapon systems acquisitions in order to lower life cycle costs and improve weapons systems performance. An open system is one that implements sufficient open standards for interfaces, services, and supporting formats to enable properly engineered components to be utilized across a wide range of systems with minimal changes, to interoperate with other components on local and remote systems, and to interact with users in a style that facilitates portability

The Navy Open Architecture (OA) is a systems design approach, supported by verifiable governmental testing, that implements open specifications for interfaces, services and supporting formats. The OA enables software components to work across a range of systems and interoperate with other software components on local and remote nodes. The Open Architecture Computing Environment (OACE) guidelines specify middleware technologies and products including CORBA, of which SQL is a query language type, and DDS.

One of the most overlooked aspects of software reuse and open architecture is data. For many applications, this makes sense since the data is local to the processing object. However, for command and control, fusion, and intelligent computing applications, just the opposite is the case – data is shared across many, if many cases, most objects. In command and control and fusion applications, the situation awareness files, consisting of track files, mission files, intelligence files, etc., are shared across almost all applications. Therefore, it is essential for software reuse and open architecture for the processing elements to be able to interoperate at the data level. This requires either standard data objects or middleware translators.

This project, however, requires an open architecture not just for software reuse but also for:

a. Process Synergy
b. Operational Adaptation
c. Evolutionary Adaptation

2 TECHNICAL APPROACH

The following subparagraphs will describe the core of the technical approach, an ontology open architecture layer.

2.1 Information Scope

In accomplishing this project expeditiously, it is important to capitalize on prior work in developing an Information Elements taxonomy. One widely cited study, entitled "Multi-INT Fusion Measures of Performance," exhaustively analyzed warfighter Essential Elements of Information (EEI) in order to quantitatively measure the benefits of improved multi-intelligence fusion. In this study, an information requirements model was developed to answer the question, *"What are the information needs of the warfighters that might be improved by advanced fusion architectures?"* Thousands of EEIs [15] were analyzed and categorized as to the required information. The object types were categorized into object taxonomies. The high-level object types are shown in Table 2-1 along with summary narratives of the subtype objects. The information needed about those objects categorized into the information types shown at a high-level in Table 2-2. The information types had five levels of decomposition and constituted what in the study was called a "knowledge matrix". The study was very well received and briefed throughout the DoD.

The object types with their levels of composition (Table 2-1) and the information types with their levels of precision and accuracy (Table 2-2), can be viewed as basis vectors of a situation awareness information space, as suggested by Figure 4.

This was noted by both the study director and the scientist leading level 2 fusion for the Army's All-Source Analysis System (ASAS) [30]. They also noted similarities to what is now known as the C2IEDM, a robust, comprehensive, and rigorous model of the battlespace information domain [11] that will be discussed in greater detail in our technical approach (paragraph 3.1.1).

Table 2-2. Fusion Types of Information

Kinematics	Location, velocity, and trajectory (past and predicted), from detection to accuracy sufficient for PGMs
Identification	Broad type to specific unit and with varying certainty
Activity	General to specific plan and with varying certainty
Status	General to specific and with varying certainty
Intent	General to specific and with varying certainty

Table 2-1. Fusion Object Types

Platforms and Facilities	Ships, aircraft, missiles, vehicles, SOF units, SAM sites, TELs, etc. from Company level up to Corps level.
Infrastructure	Communications networks, electrical networks/grids, transportation networks, etc.
Political	National organization, intent, internal conflicts, economic triggers and indicators, etc.

		Information Categories and Examples					
Object Categories	**Examples**	**Location**	**Movement**	**Identify**	**Status**	**Activity**	**Intent**
Platforms and Facilities	Units, vehicles, sites, facilities, aircraft, ships, satellites	lat/long	spd/hdg	country / alliance, type/class	readiness	targeting, reconitering	COA
Infrastructure	Comm, power, transportation, water/sewer	network, grid	throughput, flow rates, amps	name, part-of relationships	BDA, op levels	repair, broadcasts	expansion plans
Sociological	Culture, religion, economic, ethnic, government, history, languages	temples, historic structures	relocations	names and associations	stability, vulnerabilities	political or religious activities	religious or political plans
Geophysical	Terrain, weather, climatology, oceanography, astrometry	feature lat/long, alt/dpth	flowraters, tides	names	sea and river levels, temperature	storms, volcanos	forecasts

Figure 4. Span of Information for BMC2

2.2 Ontology Open Architecture Layer

Despite substantial developments in individual algorithms (e.g., for estimation and assignment), progress in developing fusion systems that can operate over the large and diverse range of required knowledge has been limited. Advances are needed for broader level 1 (object refinement), 2 (situation refinement), and 3 (threat refinement) fusion, one issue involves the lack of a tractable way to address a large-scale domain that allows modular and collaborative evolution of multiple fusion algorithms. This is required, as no single algorithm can implement all of the techniques required for differing objects and relationships. Advanced fusion requires modular interoperating components. Compounding this for military data fusion is the DoD's federated acquisition process. This process makes inter-acquisition coordination difficult-leading to inefficiencies and degrading system-of-systems coordination, interoperability, and effectiveness. Achieving evolution towards efficiency requires common portable, or at least interoperable, software components. Another impediment to large-scale fusion has been the tight binding of fusion designs to the expected operating environment. This makes performance brittle when an unexpected environment occurs in operations. Level 4 (process refinement) fusion requires adaptable and composeable capabilities.

The potential dis-interoperabilities and inefficiencies of current fusion architectures can be mitigated in a number of ways including reduced, coordinated, and standardized coupling of components. The goal is an ontology-based fusion architecture that will enable fusion agents to operate in independent, yet coordinated, manners. The approach is based on recent research in the area of ontology-based fusion in the international fusion community [77]. The approach is also well grounded in actual DoD fusion systems and algorithms such as CEC, GCCS, MSI, SIAP, and the Navy Open Architecture. The principal features of the ontologic OA layer approach are:

- Coupled Composability
- Semantic Coherence

Each of these is described in the following subparagraphs

2.2.1 Coupled Composability

One of the desired properties of good software architecture is composability, the capability for creating new applications by combining existing and/or new software components. Composability is a function of whether the software components are compatible or not. Ideally, we should be able to use existing components in a new system where a similar function is required. Code reuse has been a goal for computer science for decades now. One of the intended benefits of object-oriented (OO) programming has always been code reuse. OO promotes a homogeneous view of types, where every data type in the system, including primitive types, is an object derived from a base object type. Every data element in a program is an object and has known properties.

In the Microsoft .NET framework, *managed* code has rich support for object-oriented constructs such as interfaces, properties, enumerated types, and classes. All of these code elements are collectively referred to as types. Managed code introduces new object oriented constructs including custom attributes, advanced accessibility, and static constructors (which allow you to initialize types, rather than instances of types) thus extending other object oriented environments. Managed code can make use of pre-built libraries of reusable components. These libraries of components are called *managed assemblies* and provide the basic building block of binary composability. Reusable components are typically packaged in files called assemblies, but even a managed executable is a managed assembly. Binary composability allows your code to use other objects seamlessly without the necessity to have or compile source code from the third party code. This is largely possible due to the rich descriptions of code maintained in the metadata. The .NET framework inherently provides very strong versioning ability, through the Common Language Runtime (CLR). Since applications may be composed of many objects published in different assemblies, it is necessary to manage versioning issues as new versions of the various pieces are installed on a system. The CLR knows enough about an object to know exactly which version of an object is needed by a particular application.

It is important to note that object technology (OT) is not sufficient to enable composability, even though most available technologies for component-based development are object-oriented. JavaBeans and Enterprise JavaBeans exemplify component-based technology. The Object Management Group's Unified Modeling Language—itself an outgrowth of object-oriented analysis and object-oriented design—actively addresses component concepts. OO programmers generally agree that OT was a useful and convenient starting point for Component Based Software engineering (CBSE), but that by itself, OT does not express the full range of abstractions needed for CBSE; and it is possible to realize CBSE without employing OT. Thus, OT is neither necessary nor sufficient. Moreover, CBSE might induce substantial changes in approach to system design, project management, and organizational style—changes that go well beyond those implied by a large and growing base of industry experience with OT.

OT alone is insufficient for CBSE when the component's role as replacement unit is considered. The definitions above each address at least one characteristic related to replaceability: explicitly specifying context. Concretely, this might be implemented via a "uses" clause on a specification, that is, a declaration of required system resources. This suggestion causes some contention because a "uses" clause implies that the interface describes an implementation rather than an abstraction of possible implementations. OT does not typically support this concept—and there are strong arguments why it should not. However, these lose force when applied to design-level abstractions, especially when attempting to compose using existing components.

Some programmers are seeking ways to insulate their approaches to CBSE from OT, because the OT technology market (especially distributed OT like Java, Corba, and ActiveX) is unstable and contentious. Many programmers treat distributed OT as infrastructure "plumbing" and components as larger-grained abstractions and implementations applicable to diverse infrastructures.

The degree to which we are able to "plug in" components relates directly to the degree to which components adhere to some set of predefined constraints or conventions. Prominent component technologies - Enterprise JavaBeans, ActiveX, and Corba (assuming OMG adopts a component model) - impose constraints on components. For example, the ability of a component infrastructure to inquire into a component's interfaces requires that the component implement some service or obey some convention as defined by its underlying component infrastructure. Participants at a recent conference on CBSE argued that components should implement two interface types: a functional one that reflects the component's role in the system, and an extra functional one that reflects the component model imposed by some underlying component framework. The latter interface expresses the architectural constraints that enable composability and other desirable properties of component-based systems. Our understanding of what makes a component a component is inextricably linked to our understanding of the architectural constraints imposed on components by a component framework-cum-object model.

Although components and architecture clearly go hand in hand, the "two interface" suggestion unduly emphasizes the role of the component framework in software architecture. It may be useful to maintain a clean separation between the software architecture and component framework. A more general definition avoids this problem but still preserves a component—architecture duality by recognizing three different views of architecture:

- Runtime. This includes frameworks and models that provide runtime services for component-based systems.
- Design-time. This includes the application-specific view of components, such as functional interfaces and component dependencies.
- Compose-time. This includes all the elements needed to assemble a system from components, including generators and other build-time services; a component framework may provide some of these services.

2.2.2 Semantic Coherence

What is needed is a semantic knowledge processing capability to request and accept input from different fusion engines and thus combines information from diverse sources in an intelligent manner to derive an operational picture that captures knowledge, uncertainties, and probabilities about the state of forces as well as commanders intent (both enemy and friendly). Intelligent agents will be able to explain their reasoning, describing what evidence they are basing conclusions and also supporting what-if analysis under varying hypotheses. The super-ordinate objective of the project is to broaden and increase the cognitive capabilities of the computing infrastructure to provide advantages to the warfighter in an information intensive, network centric environment for filtering, detecting, tracking, fusion, and situation assessment.

What is sought is an expert based, battle space information ontology that can be the basis for an inference net with distributed cooperating inference engines. This would lead to many benefits for automated and massive inference such as:

- Allowing the use of large E-R models developed by DoD that cover all form of defense activities and objects, allowing connectionist access to prior knowledge from implementing databases, and enabling the massiveness required for weak evidence accumulation. For example, MIDB has over 100,000 worldwide radar sites – SAM, AEW, ATC, etc. – along with their composition, 20,000 worldwide military airfields along with their typical aircraft types and numbers, etc. A single observation will typically have many hypotheses regarding, for example, the emitter type, the specific emitting site, site radar, and particularly the emitting platform for mobile platforms. However, additional information on the platform counts and basing locations, un-refueled radii (e.g., for aircraft or small boats), speed over terrain, etc. that is deeply inter-related can influence inference beliefs and accumulate evidence to reduce ambiguities.

- Bringing the mathematical rigor of E-R modeling to bear along with the proven ability to collaborate and converge to large-scale ontologic concurrence.
- Allowing direct implementation in a COTS Data Base Management Systems (DBMS), which has many software reliability benefits, as described in [44]
- Providing a new architecture in which the objects of interest and the prior knowledgebase are integrated and consistent, with a higher fidelity representation of the battle space information domain. For example, new ESM/ELINT observations can activate the belief level of EOB information from "believed-to-exist-last-RECON-mission" to "does-currently-exist".
- Providing open architecture backplane data, triggering, inference, and conditional independence for the fusion components that reverses the conventional software architecture of process elements that are the primary objects of interest to one where the domain objects are the primary objects of interest, with process elements embedded to provide belief updating and networking in an information-based open architecture, as suggested by Figure 5.

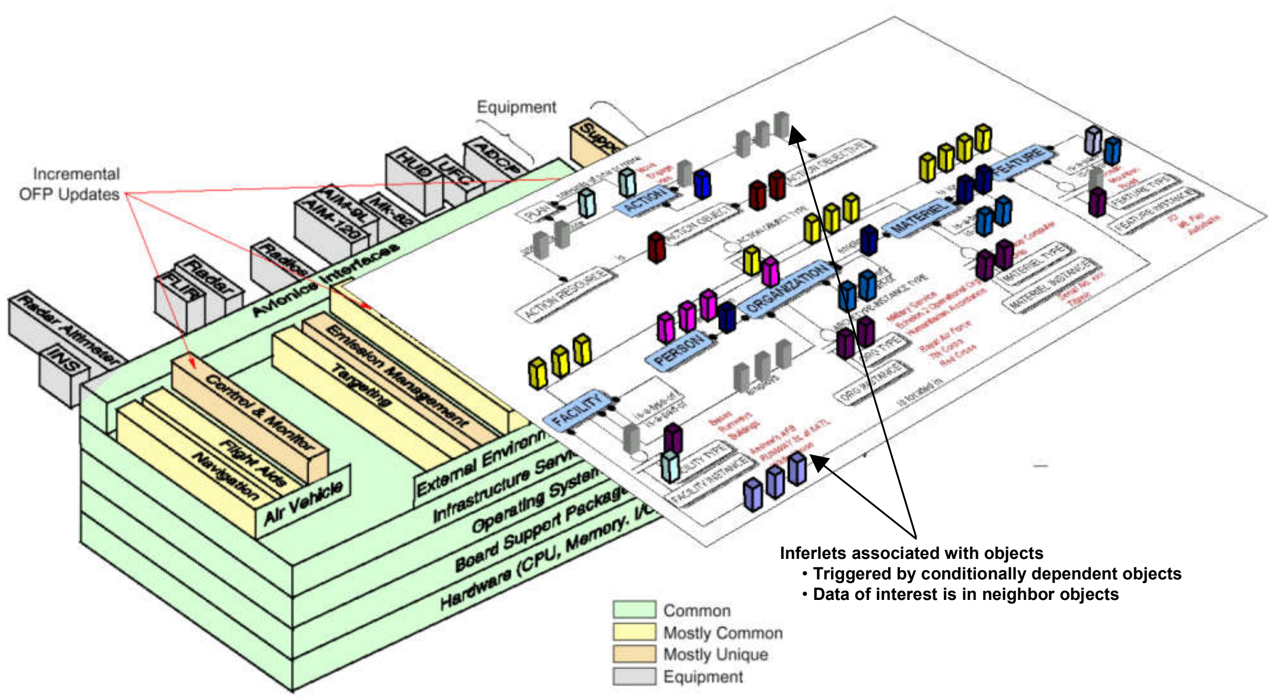


Figure 5. Open Architecture Data / Ontology Layer

2.2.3 Ontology Data Services

Our goal is to implement an open fusion architecture that allows multiple independently developed fusion algorithms to interoperate in a coordinated and collaborative manner, allows for run-time exploration of alternative algorithms as a result of performance monitoring results, layers on top of on-going open systems models, and is portable. In this project, SBSI suggests the following characteristics and requirements that can be achieved through our approach:

1. System Reliability. System crashes and hangs caused by data corruption can be reduced
2. Presentation Accuracy. Output data is consistent.
3. Object Fidelity. The objects of fusion can be treated with more accurate and complete object semantics.
4. Inference Execution Accuracy. Inference can operate with greater completeness and faithful object inter-relationships.
5. Backtracking and Auditing. Input, state, and output history can be maintained to a very large degree.
6. Inference Design Accuracy. Inference design can be made more logically coherent and complete.
7. Tractable Data Management. Reference, track file, sensor file, etc. can be managed in a systematic manner.

2.2.3.1 DBMS Services

In the prototype effort, we used the Times Ten Real-Time Database software for real-time event processing, but any other real time database with the same characteristics would work just as well. TimesTen provides database software for real-time event processing – a fundamental requirement of time-critical applications used in command and control systems. TimesTen is a re-write of ANSI SQL to optimize for RAM versus disk-based virtual memory access. It supports open architecture with ANSI SQL compliance, binding to common programming tools, and an interface to offline DBMS's such as Oracle.

2.2.3.2 Publish / Subscribe Triggering

In the "publish-subscribe" (PubSub) design approach: a person or application publishes information, and an event notification or the data itself is broadcasted to all authorized subscribers. The relationship between the publisher and subscriber is typically mediated by a service that receives publication requests, broadcasts event notices and/or data to subscribers, and enables privileged entities to manage lists of people or applications that are authorized to publish or subscribe. In most PubSub services, the focal point is a "topic" or "node" to which publishers send data and from which subscribers receive notifications and/or data. Additionally, some nodes may also maintain a history of events and provide other services that supplement the pure PubSub model. Event-driven network communications is important for real-time applications, because real-time systems cannot afford the data delays and wasted network bandwidth associated with polling for data. PubSub automatically distributes event data to interested subscribers.

Our approach is to use industry-standard CORBA [16] protocols such as the SQL query language for system management while relying on optimized protocols for time-critical data transfers. The approach allows for transparent, automated data conversion between different machines. An important consideration in designing a PubSub approach is reliability. Mission-critical applications must be able to maintain communications in the face of network delays, network failure, or computer failures. Thus it is important to support routing around network failures, real-time data integrity checking, data retransmission as required, and optional network reconnection and fail-over.

One of the strengths of the PubSub method is it has the potential to produce very high performance. It is currently possible for publishers to achieve transmission rates of hundreds of thousands of updates per second. PubSub is also easily tailored for real-time requirements. All events can be time-tagged, and subscribers can determine whether event data arrives in time. Distributed time synchronization keeps publishers' and subscribers' clocks accurate. PubSub can also handle data rate mismatch between publishers and subscribers. If a publisher is generating data too quickly for a subscriber, there can be selectable options for buffering data or else applying intelligent data quenching to discard excessive data. Different subscribers can be individually tailored as to their data buffering behavior.

2.2.3.3 DDS Specification

Real-time fusion applications have a requirement to model some of their communication patterns as a pure data-centric exchange, where applications publish "data" which is then available to subscribers that are interested in it. More generally, any application requiring selective information dissemination is a candidate for a data-driven network architecture. Predictable distribution of data is of primary concern to real-time applications. It is important to be able to specify resource availability and provide policies that align resources to critical requirements. The capability to scale to hundreds or thousands of publishers and is also important. This is actually not only a requirement of scalability but also a requirement of flexibility: on many of these systems, applications are added with no need/possibility to reconstruct the whole system. Data-centric communications decouples senders from receivers; the less coupled the publishers and the subscribers are, the easier these extensions become. Distributed shared memory is a classic model that provides data-centric exchanges. However, this model is difficult to implement efficiently over a network and does not offer the required scalability and flexibility. Therefore, another model, the Data-Centric Publish-Subscribe (DCPS) model, has become popular in many real-time applications.

DCPS relies on a "global data space" that is accessible to all interested applications. Applications that want to contribute information to this data space declare their intent to become "Publishers." Similarly, applications that want to access portions of this data space declare their intent to become "Subscribers." Each time a Publisher posts new data into this "global data space," the middleware propagates the information to all interested Subscribers. Underlying any data-centric publish subscribe system is a data model. This model defines the "global data space" and specifies how Publishers and Subscribers refer to portions of this space. The data-model can be as simple as a set of unrelated data-structures, each identified by a topic and a type. The topic provides an identifier that uniquely identifies some data items within the global data space. The type provides structural information needed to tell the middleware how to manipulate the data and also allows the middleware to provide a level of type safety. However, the target applications often require a higher-level data model that allows expression of aggregation and coherence relationships among data elements. Another common need is a Data Local Reconstruction Layer (DLRL) that automatically reconstructs the data locally from the updates and allows the application to access the data 'as if' it were local. In that case, the middleware not only propagates the information to all interested subscribers but also updates a local copy of the information.

There are commercially available products that implement DCPS fully and the DLRL partially. However, these products are proprietary and do not offer standardized interfaces and behavior that would allow portability of the applications built upon them. The purpose of The OMG DDS specification is to offer standardized interfaces and behavior.

The OMG DDS specification (Data Distribution Service for Real-Time Systems) describes two levels of interfaces. The first is a lower DCPS (Data-Centric Publish-Subscribe) level that is targeted towards the efficient delivery of the proper information to the proper recipients. The second is an optional higher DLRL (Data Local Reconstruction Layer) level, which allows for a simple integration of the Service into the application layer.

2.3 Multi-Node Distributed Fusion Architecture

While the focus of the next generation architecture is the intra-platform computing architecture, it also opens up interesting options for future fusion distributed processing inter-node architectures, in the sense described by the DoD Architecture Framework. Ever since NTDS and ATDS first went to sea, there has been recognition of the need for platforms to inter-operate in the development of their track pictures. At the very least, this has been viewed as data sharing and engagement coordination. There has also been a vision that the platforms could work together as a distributed processing system-of-systems. Both the ACDS Block 1,

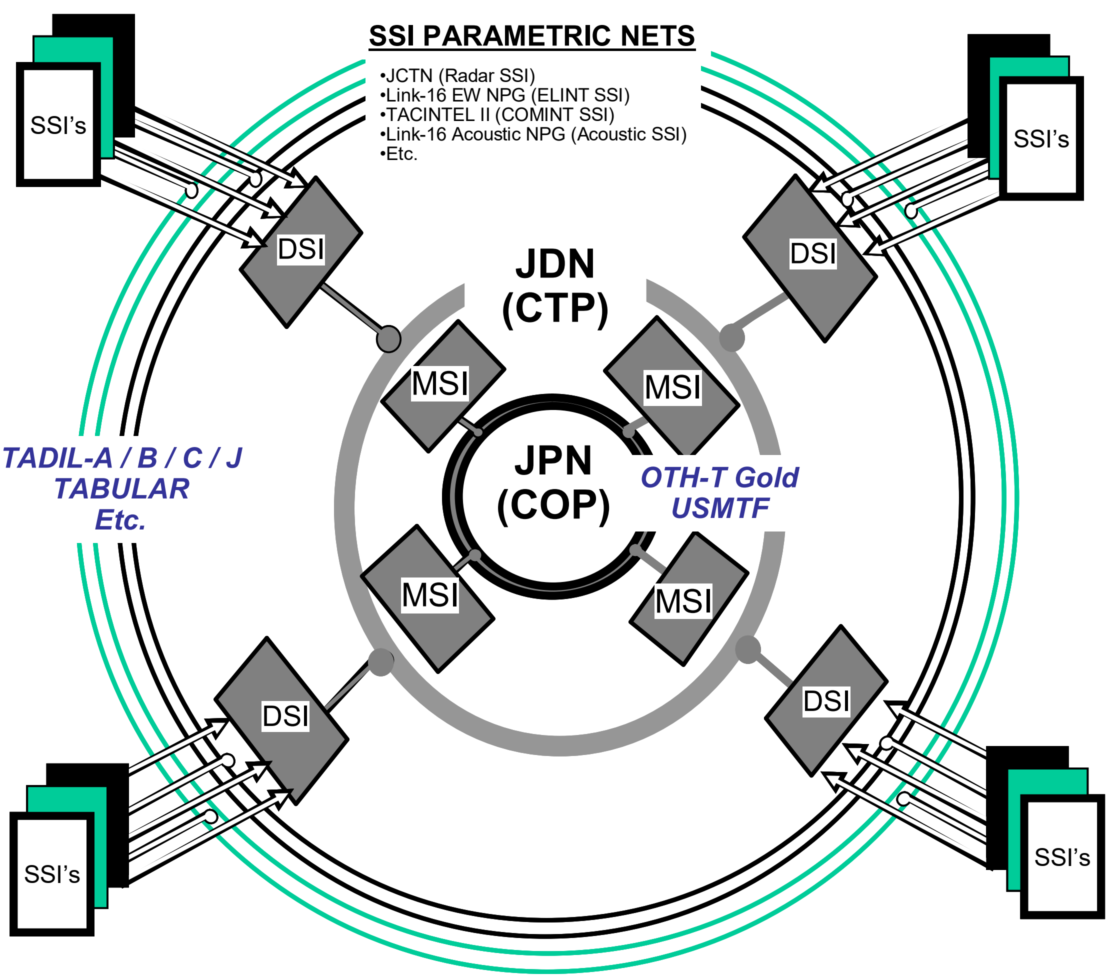


Figure 6. Current MSI Multi-Node Fusion Architecture Overview

with its original plan for a 3-ship distributed system, and AEGIS, with its Battle Group AAW Coordinator initiatives, sought this SoS concept. CEC is a result of these initiatives.

A limiting factor has been the awkward and costly data sharing, synchronization, and reconciliation mechanisms. The current and near-term state is shown in Figure 6. This figure shows how even in this advanced MSI architecture, the various levels of fusion involve message passing. The reason the message passing architecture has worked so well for so many years is that it is rigorously engineered, configuration controlled, and tested so that data can be automatically processed and acted upon with predictable and safe results. With this architecture, nodes can join up in theater and exchange data that has such high integrity and enforcement of "business rules", that ordnance, machinery, interceptors, and other resources can be controlled with high assurance. At the same time, the very processes that have made this possible have also limited the depth and breath of data sharing and been brittle to new and emerging mission situations. For example, a major limiter of level 1 correlation is the poor representation of uncertainty in the messages – track quality (TQ) in the TADILs and Areas of Probability (or Uncertainty) in the character-based messages (OTH-T Gold, USMTF, TABULAR). At best, TQ is a positional error area but it is well known that it is often just a hit / miss counter. While engineers have been ingenious in inferring the actual 4-state uncertainty (covariance matrix) by modeling the sending node's sensors (and sometimes even trackers), it is well known that simply sending the source covariance would be simpler and better (assuming the covariance is accurate!)

The next generation fusion architecture we have created in this project provides new ways to exchange, synchronize, and reconcile multi-node information along the lines of Figure 7. In this architecture, note the following possibilities:

- The TADIL and other nets can still co-exist, for coalition, legacy, engagement and AIC functions, etc.
- The entire command and control database can be shared to another node, or subsets, as required, without the need for costly formatting and parsing software. Both XML and J/ODBC exchange can be done with COTS services that are part of Navy OA.
- Through the simple design of placing the node source as a primary key attribute, a node can maintain its version of belief side-by-side with the other node's versions. Indeed, this is how we conducted the association experiments described in Appendix B, by placing the migrated foreign key, REPORTING-DATA, "above the line', as a primary key attribute. For this limited experiment, it is evident this design is very convenient to fusion algorithms and, by virtue of its simplicity, is likely a good design for other purposes.

- A whole new area of fusion progress on multi-source reconciliation is opened up by this architecture. For example, if given another node's correlation decisions and discovering difference with own node's, one could then look at the correlation candidates and scores to see what caused the differences, then the source tracks and kinematics that caused the differences, and so on. In other words, it is possible to back-track the reasons for the differences ("explaining away") via the "pedigree" data implicit in the ontology's related data, and present that root cause to the operators, rather than just the result which could be too complex and time consuming to evaluate.

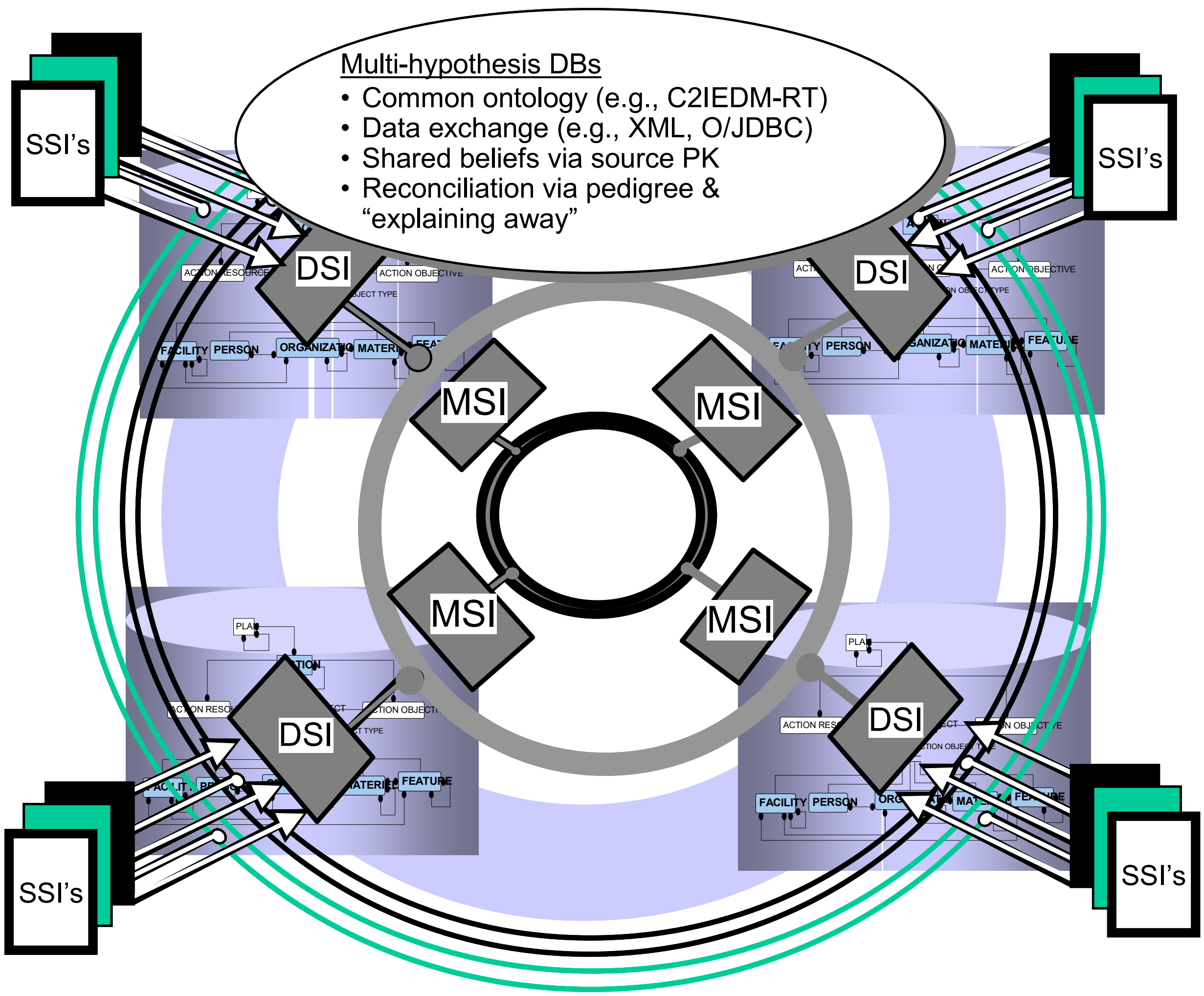


Figure 7. Distributed Processing Using the Next Generation Fusion Architecture (DoDAF SV-1)

3 PHASE I PROGRESS AND RESULTS

The section describes the work and results from Phase 1. The results have been very positive. Fusion software from actual BMC2 systems has been conformed to operate under a publish / subscribe triggering mechanism embedded in an ontology derived from a highly interoperable object-oriented model implemented in an embedded DBMS. At the same time we have been able to push the envelope on formalizing a way to use the ontology for more advanced fusion inference foreseeable in the future. The following subparagraphs will describe:

- Steps we took to enhance the standard C2 data model to serve as a fusion ontology model
- How we implemented the model in a COTS DBMS
- The experiments run with fusion software from existing systems
- Axiomatic theory advances

3.1 Ontology for BMC2 Fusion

An ontology is simply a model of the world. In Philosophy, the term refers to a model of the entire world, a systematic account of existence. In artificial intelligence (AI), ontology refers to a model of some part of the world – some domain, or area of interest. AI requires an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. There is a very close correspondence between the military concept of a Common Operating Picture and the concept of ontology in AI. They both refer to the way we understand and thus reason about a situation. Military commanders, like intelligent agents, reason better if they have a shared, accurate, and complete understanding of the situation on the ground. In AI systems, the only things that "exist" are things that can be represented. When the knowledge about a domain is represented in a declarative language, the set of objects that can be represented is called the universe of discourse. The ontology of a program can be described by defining a set of representational terms. Definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is the statement of a logical theory.

Ontologies are catalogues of explicit representations of domain semantics; they show how things are conceptualized or what they mean. Ontologies, by definition, are intended to be understood by machines. Thus, an ontology, as commonly defined in the IT world - and as specified by formal ontology procedures such as IDEF5 - requires that 1) terms describing entities are defined by reference to a common namespace and 2) machine processes to interpret and reason about entities are defined through formal, computable methods. Ontologies require explicit agreement on the representation of meaning and precisely specified processing assumptions. They provide representation-process pairs that are sufficiently specified to enable machine processing of architecture data. Ontological models support reasoning at the class level – engendering in machines the same kinds of processing savings enjoyed by human reasoners. For example, we often reason about instances by invoking what we know about the class to which they belong, or hypothesize new classes by generalizing across the common features of instances.

Part of the power of this approach comes from the use of a high level ontology to anchor the base classes, that is, a superordinate ontology that would sit conceptually above and provide an anchor for the proposed battlespace information ontology. This ontology would know things like what an airplane is, and what it means to fly. This would allow the machine to reason using the sort of approaches often called common sense. SUMO (Suggested Upper Merged Ontology) is a possible standard put forward as part of the IEEE Standard Upper Ontology (SUO) Working Group. The goal of SUO is to develop a standard upper ontology to promote data interoperability, information search and retrieval, automated inferencing, and natural language processing. SUMO has been translated into various representation formats, but the language of development is SUO-KIF (a version of the first-order predicate calculus). KIF also served as the basis for the predicate calculus operations in IDEF5.

IDEF5 employs two languages: the schematic language and the elaboration language. The schematic language is perhaps closest to IDEF1 and IDEF1X; it provides a relatively easy to use graphical language that has been tailored to express the most common forms of ontological information. The elaboration language contains all definitions and characterizing axioms, in a structured text language with the full expressive power of first-order logic and set theory. It can express almost any condition, or relation, or fact needed to express any given kind of thing, property, relation, or process found in a domain. The *core* of the IDEF5 Elaboration Language, which enables the expression of axioms, is based on the KIF [Genesereth]. IDEF5 includes an extensive library of information consisting of characterizations of commonly used relations, the top seven shown in Figure 8.

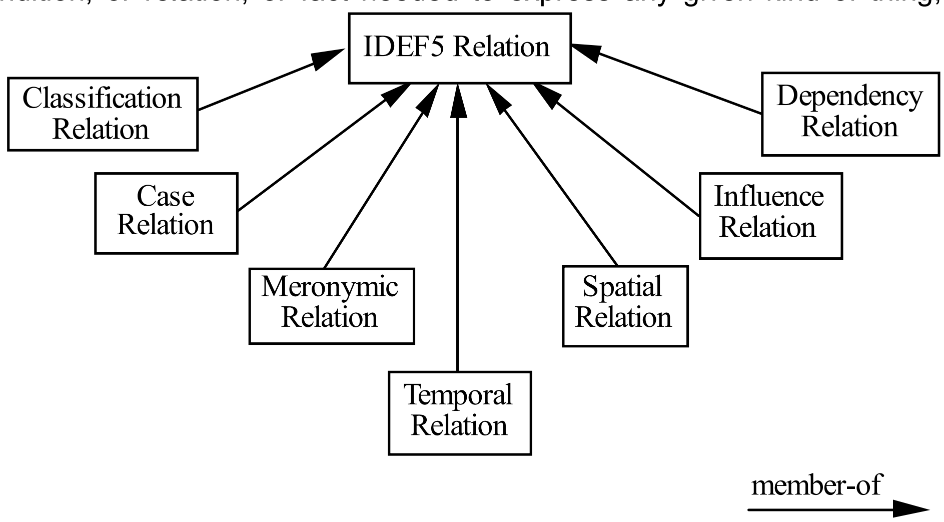


Figure 8. Ontology Relation Types

3.1.1 Ontology Model at IDEF1X Stage

The partial ontology design we developed has the following two principal components, described in the following subparagraphs:

- An object-oriented, Entity-Relationship model, enhanced for automated reasoning and BMC2 fusion
- Data services, via an embedded DBMS, with a publish / subscribe triggering mechanism

We needed an ontology foundation for battle space awareness that:

- Supports a comprehensive global representation of the elements and properties of battle space awareness and multi-sensor data,
- Supports fusion and inference employing reference and non-real-time context information, enforces coherence of all battle space awareness data,
- Modularizes data access requirements across networks by establishing conditional independences, and

- Illuminates data interdependence in support of sensor queuing

Such a foundation provides a declarative representation that will support automated reasoning. To actually accomplish automated reasoning, we require an inference and inference tasking model that capitalizes on the rigorous inductive and deductive capabilities inherent in formal ontologies, supports user-hypothesized as well as machine generated inferencing, and allows for control of the fusion agenda by using the ontology as a template of how a human expert would think about the fusion problem.

Demonstrating the viability of our approach, we started with the Command and Control Information Exchange Data Model (C2IEDM). C2IEDM is the NATO standard data model for command and control [11], and is the model for the Army's Future Combat System (FCS) [9][23] and Single Integrated Ground Picture (SIGP) [39]. It replaces the U.S. DoD data model for command and control, the C2 Core, and is included in the Net Centric data repository. While it has yet to be adopted by SIAP or the U.S. Navy, no alternative currently exists. It is the result of over 15 years of Command and Control data modeling by the worlds leading data modelers.

A very high-level overview we developed of C2IEDM is shown in Figure 9. C2IEDM is an object-oriented, Entity-Relationship (OOER) model, in that it uses extensive object class hierarchies. These provide compactness and extensibility. Compactness assures that common properties do not have to be repeated, and extensibility because new data elements can inherit properties from their superclass. There are many benefits to an OO design, documented in many textbooks and the OMG web site. As well, there are many benefits to ER design, documented in many textbooks and testified to by the phenomenal success of a wide range of commercially available relational database software, such as Oracle. We believe the C2IEDM is sufficient in scope to cover BMC2, based upon the similarity to Information Element taxonomies we have developed and mapped to cover all of TADIL-J DFI/DUIs, MIDB, IBS, and other BMC2 data elements.

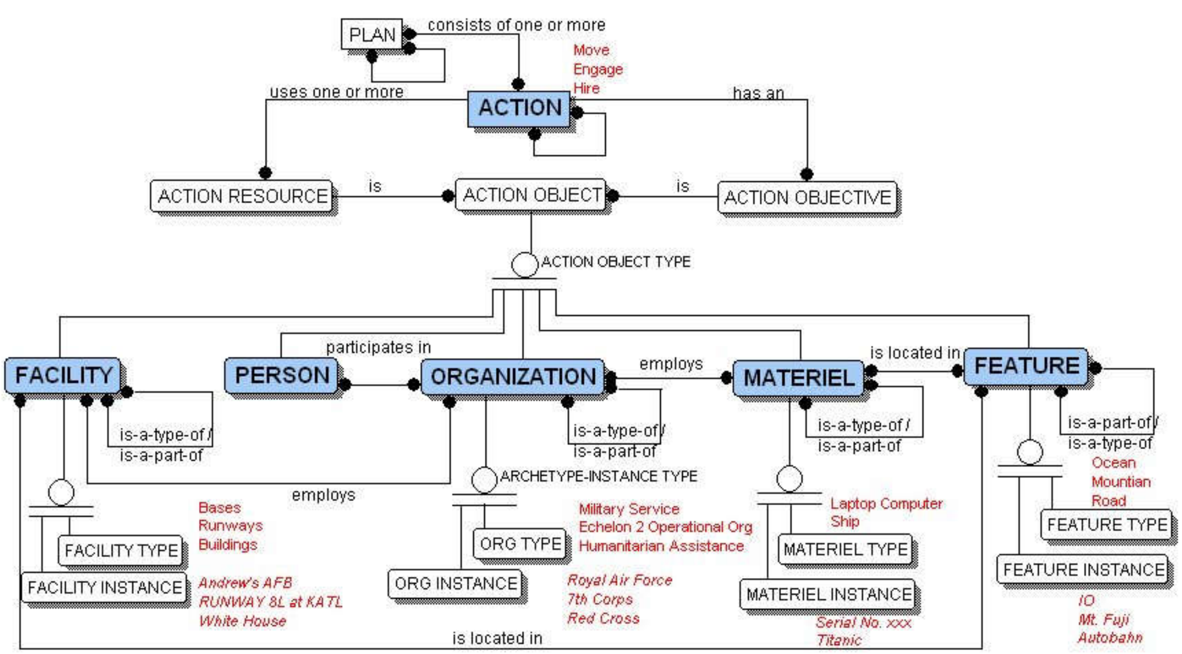


Figure 9. C2IEDM-RT Overview

From a real-time or BMC2 perspective, however, the C2IEDM a first appears insufficient and not feasible for real-time application. Indeed, early Generic Hub and C2 Core work was more linked to non-real time applications, such as the Joint Operation Planning and Execution System (JOPES), and non real time message standards, such as U.S. Message Text Format (USMTF). However, the SBSI staff has found ways to overcome these shortfalls, and take advantage of the C2IEDM's benefits. We call our extensions and modifications C2IEDM-RT – where the RT stands for real-time. Our extensions fall into 4 categories.

a. Uncertainty handling

We added uncertainty representations to C2IEDM to support uncertainty handling by explicitly representing probabilities within domain knowledge and to enable machine reasoning about information, beliefs, and uncertainty parameters derived from various fusion engines. Uniquely, this approach addresses one of the most difficult issues in achieving SA, real-time fusion of (sometimes inconsistent) information from disparate sensors and sources. This supports semi-automated data fusion using knowledge representation techniques to explicitly represent domain knowledge and to enable machine reasoning about information, beliefs, and uncertainty parameters derived from various fusion engines. We had written about the approach in [44], and carried the same view forward in this SBIR. There were three types of extensions done:

The simplest case is that of a model relating two objects via an associative entity. A model fragment may support many-many relations and/or information about the relation that is not specific to the individual objects. The associative entity may provide a many-to-many relationship between PERSON and ORGANIZATION, allowing a person to be a part of many organizations, conversely, allowing an organization to have many people. Based on common sense and powerful, it is nevertheless binary -- you are either a part or not; no maybe's. To represent uncertainty in this case, we need to add a confidence value to the associative entity and a multi-hypothesis index into the primary key. An instance in the associative entity represents each hypothesis. To take a military example, one instance in the associative entity might express that Track 5 might be a friendly with 40% certainty, another instance might associate Track 5 with a hostile organization with a 30% certainty, while another associates Track 5 with "unknown" with a 20% certainty. Probability masses could also be used.

In the case of migrated foreign keys, it was necessary to create an associative entity, with the confidence value and multi-hypothesis index, so the multiple hypotheses as to the referenced object could be maintained.

In the case of value-bearing attributes, it is necessary to add an uncertainty representation for those values. The case we focused on for fusion purposes was kinematics. In this case, the attributes have coupled uncertainty and so a covariance matrix was added.

b. BMC2 Precision

The C2IEDM did not support precision tracking and timing so it was necessary to add a 9 state kinematic state, and an additional attribute for sub-second time maintenance. We have not yet added body orientation (e.g., direction cosines) and body axis displacements and rates (e.g., roll, pitch, yaw and their rates) but these would fit well as amplifiers on RELATIVE-POINT in the same manner as kinematic-state. These data elements will be important for ballistic missile tracking. It will probably be necessary to add uncertainty representations for orientation and navigation because of the importance of accurate and precise estimation for this type of application.

c. Temporal and Multi-Belief Dimensions

The C2IEDM has a very good structure for maintenance of data source, REPORTING-DATA. In order to support both multi-beliefs (e.g., local and remote tracking) and maintenance of track history all that had to be done was move the REPORTING-DATA foreign key into the primary key.

d. Formalized Relations

One of the problems with existing data models is that the relations specified among entities are not formally specified. Rather, relations in data models are associated with terms intended to be understood by humans. In order to support machine logical processing, the relations must be specified with formal systems, such as first order and modal logics. This is the key difference between data models and ontologies.

e. Abstraction

It is simple and natural for a human fusion expert to see corresponding elements from Intel and track databases as imperfect reflections of a single entity – but viewed from two perspectives. It is not straightforward for traditional fusion software to make this sort of abstraction-derived inference. The lack of an abstraction layer causes data integrity degradation and convoluted fusion architecture

We added multi-tiered relationships to allow explicit representation of abstraction. While it is simple and natural for a human fusion expert to see corresponding elements from Intel and track databases as imperfect reflections of a single entity (but viewed from two perspectives) - it is not straightforward for traditional fusion software to make this sort of abstraction-derived inference. The lack of an abstraction layer causes data integrity degradation and a convoluted fusion architecture

f. Measurement and Reference Database Connections

We did connect the DDRE Transmitter model (a model of an ESM / ELINT sensor measurement) and some intelligence database elements, to verify that the C2IEDM supported these connections.

The model at this point is as shown in Figure 10. This does not show the measurement and reference database connections in the interest of display size. Those fragments were shown herein in Figure 11 and Figure 12. The elements with turquoise backfill and green attribute text were added as part of this work. Although a printed copy of this document will only be able to show the model notionally, in an electronic copy, the diagram can be magnified to read the text. C2IEDM-RT is modeled in IDEF1X but because it is fully attributed with reasonably regular verb phases, the relationships can be categorized into the IDEF5 relation categories shown in Figure 8 so it can become an executable ontology. Like SUMO, because C2IEDM-RT's higher-level class properties are grounded in well-reasoned invariants, they are not only more stable and consistent; they tend to have wide concurrence and validation.

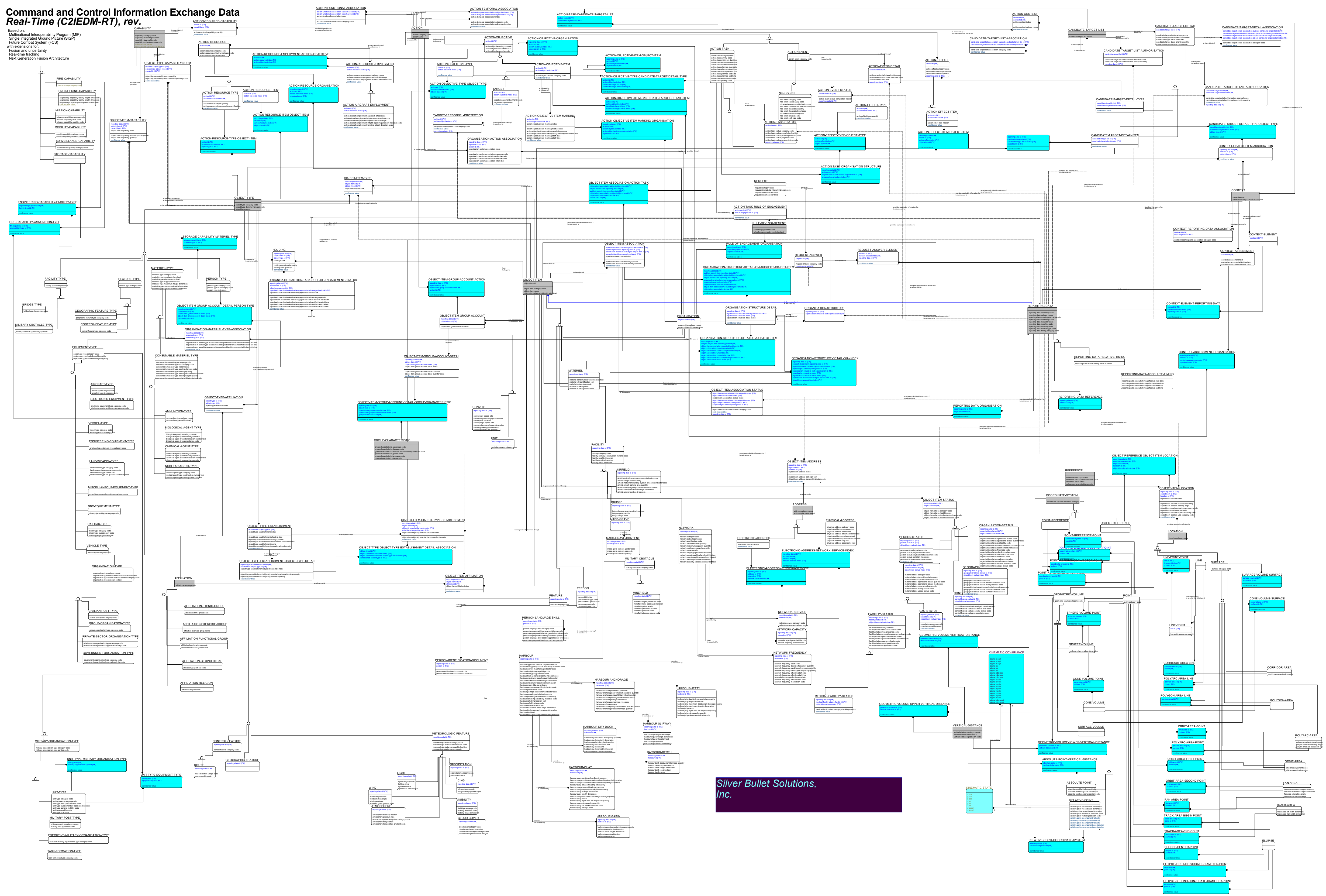


Figure 10. C2IEDM-RT Showing Extensions for Fusion

3.1.2 Sensors and SSIs

The MSI architecture categorizes sensors by the phenomenology they sense which dictates the processing to interpret the data; these categories are termed Similar Source Integrators (SSI). SSI's can process data at the measurement or SSI estimate level. SSI's can have their own communication nets. The SSI's for this project include:

- Radar SSI. The radar SSI interfaces with on-board and off-board radar detector processors. The off-board net is the CEC net.
- ESM / ELINT SSI. The off-board net is the Link-16 EW Net Participation Group (NPG). This NPG serves an EW "scratchpad" to exchange EW-specific data and coordinate EW activities such as Trouts, Rents, and Emitter Search Requests.
- IRINT SSI. The IRINT SSI TBS

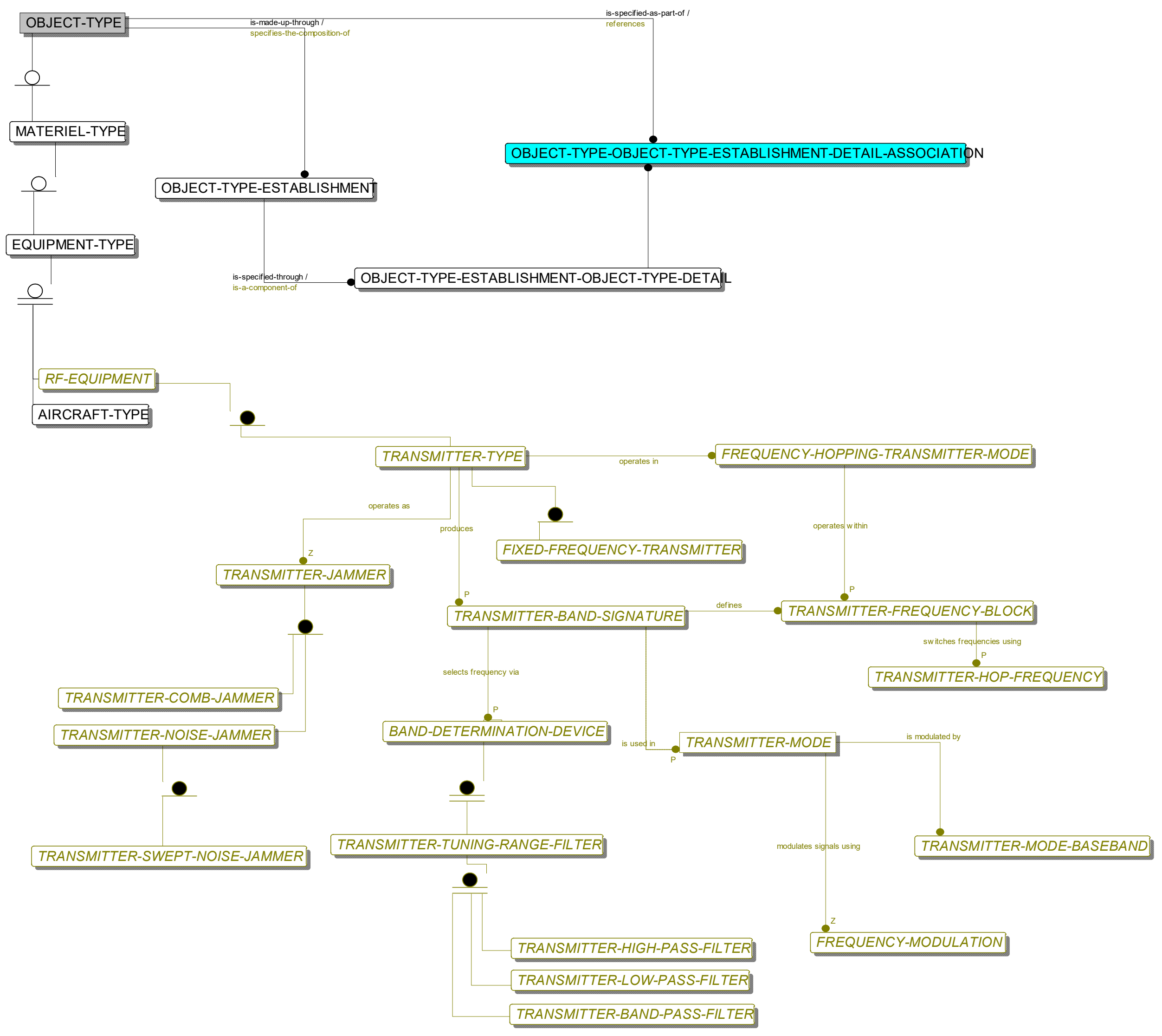


Figure 11. DDRE Transmitter Model Connected to C2IEDM-RT

The C2IEDM-RT ontology supports these and other SSI's via inclusion of what we call measurement models. Many of these have been developed by the DoD scientific panels under DDRE sponsorship. The DDRE models employ many of the same superclasses as C2IEDM-RT and can, therefore, be plugged-in to C2IEDM-RT relatively easily. The DDRE Transmitter model was plugged into the C2 Core (former C2IEDM) and used in the fusion loading research described in Appendix B. The C2IEDM-RT model fragment shown in Figure 11 shows how readily model pieces built with common superclass entities can be connected together. The italicized entity names are from the DDRE model; all others are from C2IEDM-RT. This is a model of an ESM / ELINT measurement, showing how the measurement is caused by a signal caused by a transmitter caused by an aircraft. The aircraft "cause" is embedded in the OBJECT-TYPE-ESTABLISHMENT entities, very general type-coded entities that allow representation of many owning, installed, and other relationships between object types. The turquoise-filled entity was added to the real-time version by SBSI to allow for uncertainty and multiple hypotheses as to the establishment. For example, intelligence may believe a certain radar is on an aircraft with a certain confidence and may believe it could be some other radar with yet another confidence.

The following DDRE models can be used as an initial starting point:

1. ACOUSTIC PROPAGATION
2. ANTENNA
3. ANTENNA SCAN TYPE
4. ANTENNA TYPE
5. ASTROMETRIC ELEMENT
6. DEVICE TYPE
7. DOCUMENT SEGMENT
8. DOPPLER WEATHER RADAR
9. ECM EXPENDABLE TYPE
10. GEODETIC STATION
11. GEOPHYSICAL ANALYSIS FORECAST
12. GEOPHYSICAL MEASURING DEVICE
13. GEOPHYSICAL MODEL
14. GEOPHYSICAL PLAN SURVEY
15. GEOPHYSICAL POINT CLIMATOLOGY
16. GEOPHYSICAL POINT CLIMATOLOGY
17. GEOPHYSICAL SATELLITE
18. GRAVITY
19. IONOSPHERE
20. LEAD-IN COMPONENT
21. MEASURING DEVICE LOG ENTRY
22. METEOROLOGICAL ANALYSIS FORECAST
23. METEOROLOGICAL LEVEL SUMMARY
24. METEOROLOGICAL POINT OBSERVATION
25. OCEAN PROFILE
26. RESEARCH AND DEVELOPMENT
27. ROUTE SURVEY
28. SOLAR
29. TRANSMITTER
30. TRANSMITTER TYPE
31. WEATHER

3.1.3 Data links

The U.S. is developing a Common Link Integration Processing System (CLIPS) that, among other functions, will normalize all TADIL data formats. Although still under development, the normalization is expected to be similar to the mapping shown in Table 3. The C2IEDM-RT ontology supports these data formats. As part of the Phase I SBIR, we mapped the TADIL-J Data Field Indicators (DFI) and, where necessary, Data Unit Identifiers (DUI) related to data fusion to C2IEDM-RT data elements and found the C2IEDM was totally sufficient in class structure and required only subclass or category code extensions to cover the TADIL-J fusion-related messages. (Those messages were the J2s, J3s, J7s, J4s, J6s, J7s, J12.1, and J14.0.) In prior work, we mapped VMF DFI/DUIs to information taxonomies similar to the information categories in [41] [42] which was found to be similar to C2IEDM and so believe C2IEDM-RT would cover VMF. Since TADIL-A DFI/DUI all map to TADIL-J DFI/DUI via the data forwarding volume of [80], it suffices to map TADIL-J to ensure TADIL-A is covered. Similarly, most of the

V/R TADIL-C series elements map to TADIL-J even though a formal data forwarding specification does not exist. This is known because the Command and Control Processor (C2P) IDS harmonized these data elements so the command and control systems could send/receive messages from fighter/attack aircraft in a uniform way.

Table 3. Normalized Data Format for TADILs

Normalized Message	Name	TADIL-J	TADIL-A/B	TADIL-C
0200	PU Report	2.0I 2.0E0	1 81 $5_{300,310}$	
0220	Air PPLI	2.2I 2.2E0		
0230	Surface PPLI	2.2I 2.2E0		
0240	Subsurface PPLI	2.3I 2.3E0		
0250	Land point PPLI	2.5I 2.5E0		
0260	Land Track PPLI	2.6i 26E0		
0271	Air, Surface, Subsurface PPLI – IFF AMP	2.2C1 2.3C1 2.4C1		
0272	Land PPLI - Amp	2.5C1		
0273	PPLI – Mission Amp			
0274	PPLI – Relnav Amp	2.2C3 2.3C3 2.4C3 2.5C3 2.6C3		
0275	PPLI – Displaced Position Amp	2.3C4 2.4C4 2.5C4		
0300	Reference Point	3.0I 3.0E0	$4A_{340}^{(DRT=2,3)}$ $84A_{340}^{(DRT=2,3)}$ $4B_{342}$ $4C_{303,540,541}^{(SW=2,3)}$ $5_{200,310}^{(SW=0,1)}$ $85_{310}^{(SW=0,1)}$ $9F(AC=0_{402})$	
0301	Reference Point – Offset Position Amp	3.0C1		
0302	Reference Point Area Amp	3.0C2	$9F_{?}$ $89F_{?}$	
0303	Reference Point - Sonobuoy Amp	3.0C3	$4C_{300,540,541}$ $84C_{540,541}$	
0310	Emergency Point	3.1I 3.1E0	$5_{200,300}$ 5_{300}	
0320	Air Track	3.2I 3.2E0	2 82	
0322	Track Identification Continuation	3.2C1 3.3C1 3.4C1 3.5C1	$11D_{750}^{(TR=0)}$	

Normalized Message	Name	TADIL-J	TADIL-A/B	TADIL-C
0323	Intelligence	6.01 6.0E0	11M 811M 2/82$_{320}$ 3/8$_{330}$ 84A$_{300,340}$	
0330	Surface track	3.31 3.3E0	3 83	
0340	Subsurface Track	3.4I 3.4E0 3.4C1	4A$_{300}$ 84A$_{300}$	
0342	ASW Amplication		4B$_{300}$	
0350	Land (Ground) Point/track	3.5I 3.5E0 3.5C1		
0401	EW – Frequency Message	3.7C4 14.0E0 14.2C4	6A$_{1600, 1601}$ 86B$_{403, 1600,1601}$ 6D(CNTL=4$_{1620}$)	
0402	Area of Probability Message	3.7C2 14.0C2 14.2c2	9F(AC=0$_{300}$,1) 89F(AC=0)	
0403	EW – Emitter Message	14.0C3 14.2C3	86B$_{401, 1600,1604}$ 6D(CNTL=8$_{1620}$)	
0404	EW – Scan Characteristics Message	14.0C4 14.2C8	6C 86C	
0405	EW – Identity Parametric Message	14.0C5 14.2C5		
0406	EW – Call sign Message			
0540	Acoustic Bearing/Range	5.4I 5.4e0 5.4C2	4C$_{300, 303,540}$ 84C$_{303, 541}$ 4D$_{541}$ 84D$_{541}$	
0541	Acoustic Bearing/Range Amp	5.4C1	4C$_{300, 303,540}$ 84C$_{303, 540}$ 8D$_{540}$ 84D$_{540}$	
0700	Track Management	7.01	9A(AC=1,2,4,5,7)	
0710	Data Update Request	7.1I 7.3C1	9A(AC=3)	
0720	Correlation Request	7.2I	9B(AC=7)	
0730	Pointer	7.3I 7.3C1	9C	
0740	Track Identifier	7.4I 7.4E0	9E(AC=3,7)	
0750	IFF/SIF Management	7.5I	9A(AC=9) 11D$_{322}$	
0760	Filter Management	7.6I 7.6E0		
0761	Filter Management Area Description	7.6C1		
0762	Filter Management Points Description	7.6C2		
0770	Association	7.7I	9B(AC=6,15$_{1260}$)	

Normalized Message	Name	TADIL-J	TADIL-A/B	TADIL-C
1010	Mission Correlator change			
1100	Command	9.0I 90E0	10A(ORD=2 – 5) 15	
1101	Command Amp	90C1	9E(AC=$1_{1230,1250}$,2_{1231})	
1120	ECCM Coordination Message			
1121	ECCM Coordination Continuation Message			
1220	Engagement Status	102I	14	
1121	Additional Air Engagement Status	102C1		
1230	Handover	103I	103E0	9E(AC=0,$1_{1101,1250}$) 10A(ORD=0,1,6)
1231	Handover Amp	103C1	9E(AC=2_{1101},4)	
1232	Additional Mission Correlator			
1250	Controlling Unit Report	10.5I	9A(AC=6) 9E(AC=$1_{1101,1230}$)	
1260	Pairing	10.6I	9B(AC=0–5,15_{770})	
1400	Air Control Mission Assignment	12.0I		1
1401	Target Position and Data	12.0C1 12.0C2		2
1403	Point Location Data			
1404	Ground Position Data			
1410	Air Control Vector	12.1I 12.1E0		3
1411	Close Control			9
1412	Strike Control			31 (sub label 21)
1420	Precision Aircraft Direction			
1430	Flight Path Message	12.3I 12.3E0		
1440	Air Control Controlling Unit	12.4I 12.4E0		
1450	Target/Track Correlation	12.5I 12.5E0 12.5E1		
1460	Air Control Target Sorting	12.6I 12.6E0		
1461	Air Control Engagement Status	12.6C1		
1520	Air Platform and System Status	13.2I	$11B_{1521}$	
1521	Air C2 Status	13.2C1	$11B_{1520}$	
1522	Air Stores	13.2C2		
1523	ASW Status		11C	
1530	Surface Platform & System Status			
1531	Surface C2 Status			
1532	Surface Equipment Status			
1560	Electronic Warfare Status			
1570	Controlled Aircraft Status			0
1571	Controlled Aircraft TACAN			1

Normalized Message	Name	TADIL-J	TADIL-A/B	TADIL-C
	Position			
1572	Controlled Aircraft Target Position			3A
1573	Controlled Aircraft Position Report			3B
1574	Target Velocity Report			3C
1600	EW – Parametric Information Message	3.7I 14.0I	$6A_{401, 1601}$ 6B $86B_{401, 403, 1601}$	
1601	EW – Parametric Information Continuation Message	14.0C1	$6A_{401, 1600}$ $86B_{401, 403, 1600}$	
1620	EW – Control Coordination Message	14.2i	6D(CNTL=0-3,5-7,9-14; 4_{401}; 8_{403}; $14\text{-}15_{1621}$)	
1621	EW – Control Coordination Continuation Message 1	14.2E0	6D(CNTL=$14\text{-}15_{1620}$)	
1626	EW – Control Coordination Continuation Message 2	14.2C6		
1700	Threat Warning Alert			
2000	Plain Text			
2001	Tracking Parameters			
2002	Mission Number		9E(AC=6)	
2003	Training target	12.00 12.01 12.02		
2004	Training IFF	12.03 12.04		
3001	Alerts Message			
3003	Link 4A status summary			
3004	Link 11 Status			
3005	Link 11 PU Status			
3006	Link 16 Summary Status			
3007	Link - 16 NPG Status			
3010	Unit Inactive			
3011	Data Link Initialization			
3013	Link – 11 Call Up			
3014	AIC Assignment			
3015	AIC Assignment status			
3017	OTCIXS/TADIXS A Status			
3027	Ownship Nav Data SINS			
3030	Ownship Nav Data WSN – 5/CVNS			
3031	Pitch and Roll Data			
3033	Gridlock Pads Data			
3040	JTIDS Navigation Data			
3041	JTIDS Relnav Data			

3.1.4 Reference Databases

In order to provide support to ATDS operators, we intend to make machine use of information from critical reference databases. These databases are generally formatted for human consumption, not for machine consumption. However, it is possible to engineer them for machine consumption, and this approach has previously been proven viable. For example, several members of the SBSI staff did this kind of work, groundbreaking at the time, for the Advanced Combat Direction System (ACDS) Block 1. This major Navy project aimed for new target identification technologies for tactical command and control called Multi-Source Identification (MSID), which worked in conjunction with Dissimilar Source Integrators (DSI) and Similar Source Integrators (SSI). Although the computing technology of the era did not permit the full realization of these technologies, SBSI staff was able to show a more complete realization of the ESM SSI capability in an SBIR called EW Identification (EWID). EWID was successful because it was not bound by the militarized computing technology of the time and was able to use ruggedized computers with much faster CPUs and large amounts of RAM. The SSI / DSI / MSID architecture was later extended by the Combat System Functional Allocation Board (CSFAB) and in the Multi-Source Integration (MSI) System Engineering Team (SET), both of which are reference architectures for the E-2C MSI and SIAP.) This architecture was a major advance over the search-detect-track architectures of the time that are still in place in most BMC2 systems and that cannot be reasonably extended for multi-source fusion and situation awareness functions.

MSID relies heavily on a knowledge base engineered from intelligence and other reference databases. The source data can change over time and at varying rates ranging from daily to annually. The changes depend on a multitude of unpredictable factors such as geopolitics, intelligence methods, technology developments, and foreign weapons sales. Like a human ID or EW operator, MSID performance will depend critically on accurate, complete, and timely knowledge. The knowledge base must, therefore, be maintained and updated as the reference databases change.

MSID requirements were formulated in response to Fleet deficiencies for target identification, summarized in the CNO NTDS Functional Allocation Study. ID operators were overloaded. Awaiting target identification was a bottleneck to command and control information flow. Uncertain and incomplete target identifications degraded command and control effectiveness. IFF is not a sufficient identification technique. Many targets entering the battle space do not emit IFF. Modes 1, 2, and 3 are non-secure and, therefore, any adversary can easily squawk a false code. The only encrypted mode, Mode 4, is carried only by US aircraft and ships.

This approach is based upon an analysis of the operator decision process. It was observed that operators utilize technical knowledge of targets and operational knowledge of their area of operations in their evaluation of sensor data pertaining to a track. They also consider evidence and clues gleaned from multiple sensors. Despite many attempts, there is no single sensor solution for target identification. Promising single-sensor solutions have failed to scale-up to realistic target variety types. Instead of a single sensor solution, one can expect an increase in multi-sensor evaluations. Merging this multi-sensor data in conjunction with target and area knowledge remains the most promising approach to leveraging all the combat system resources.

In some respects, the MSID a-priori database resembles the classical "threat library" of EW systems, the "library" of emitter parameter ranges and emitter-platform associations. However, MSID goes beyond these classical systems in two respects. One is the breadth of data. BMC2 systems are the fusion systems supporting the strike or expeditionary warfare group for multi-warfare. As such, they need to provide target identification of not only hostile or threat platforms, but also all platforms. They must identify friendly, neutral, suspect, and hostile tracks in the air, on the surface, below the surface, and on land. There are mission and architectural reasons for this breadth. The mission reasons are to not only reduce fratricide and potentially politically catastrophic neutral engagements, but also to improve command and control effectiveness by reducing uncertainty -- “unknown” tracks are an uncertainty. MSID capabilities are particularly germane in today’s and future warfare where tactical flexibility requirements will not allow sectorization of the battle space and where air-land battle concepts extend to the sea resulting in naval, air, and land forces, both US and coalition, operating in close proximity.

MSID requires knowledge about the each of the individual source streams, and the capability to understand how these streams relate to each other – what sort of inferences can be derived by combining across sources. In this architecture, the engagement and sensor subsystems are allowed reflexive responses, much as human behavior has reflexive response as a defensive mechanism. The BMC2 systems, as the integrating and fusion point for the combat system, then supports a more considered response based upon more complete information and a larger number of resources available to analyze the information, i.e., more operators and more computing resources. In this model, the subsystem is designed with a smaller “threat library” so that there is little ambiguity but some chance of false ID. The BMC2 system then uses a larger library that is inherently more ambiguous but has less chance of false ID. The BMC2 systems then employ additional techniques to resolve ambiguities such as

- Use of off-platform sensors via coordination on the Link-16 / 22 EW Net Participation Group (NPG) using the J14.0 parametric reports and the 31 different kinds of EW coordination supported by the J14.2 message,
- Dissimilar own-platform data such as radar and IFF,
- More extensive knowledge base data such as flight corridors, fusion algorithms that can determine platform range and velocity from LOB inputs, and rule-based processing.

Similarly, engagement subsystems are allowed reflexive but negatable actions. For example, a TAS Mk-23 is allowed to reflex within its Controlled Reaction Zones (CRZ), allowing launcher slewing and aim pointing and director slew and acquisition immediately. If the BMC2 system “considered response” reviews the action and can terminate the engagement.

The need for broader data has a cost in terms of library size. In the work we did for ACDS, the knowledge base had 25 times as many platforms as SLQ-32 libraries. In the EWID project, we had 100 times as many. The depth of increased data includes order-of-battle data and shipping and airline density data not used by classical EW systems. In the EWID project, we used this data to estimate the likelihood of an identification alternative. It was based on the common sense and mathematically accurate observation that, all other factors being equal, a value that is more highly represented in a population is more likely. This data is essential in supporting the resolution of the larger numbers of ambiguities caused by the broader library. However, it does increase the amount of library data considerably. We intend to take a similar approach for ATDS.

3.1.4.1.1 <u>Order of Battle</u>

The role of order-of-battle (OB) data in ACDS (now SSDS Mk-2) is unique among command and control and EW systems. SSDS Mk-2 uses OB for the platform-level target identification as an indicator of the types of platforms that can be expected in the detection area. There are several types of OB data:

- Naval Order of Battle (NOB). Provides the homeport lat/long of an individual ship.

- Air Order of Battle (AOB). Provides the airfield/airbase lat/long and the typical deployment of aircraft at the base. The aircraft deployment is the number of aircraft of a particular alpha-mod.
- Electronic Order of Battle (EOB). Provides the lat/long and name of land-based radar-equipped sites including SAM sites, Early Warning sites, Ground Controlled Intercept sites, Air Traffic Control sites, and so on. A site may have several types of radars, such as a Patriot site.
- Ground Order of Battle (GOB). Provides the lat/long and name of land-based sites composed of buildings such as headquarters, barracks, and so on. Occasionally, a GOB site is associated with radar.
- Missile Order of Battle (MOB). Provides the lat/long and name of non-radar carrying missile sites such as ICBM silos.

3.1.4.1.2 ELINT Parameters

- NERF (Navy Emitter Reference File). A collection of Electronic Warfare (EW) data, worldwide in scope and consists of Order of Battle (OOB), friendly (blue), non-hostile (white), and hostile (red) parametric data. Encompasses military, commercial emitters and non-communications emitters. It encompasses airborne, shipborne, missile borne, and land-based emitters of all functions.
- EWIRDB (Electronic Warfare Integrated Reprogramming Database). The baseline database of electronic characteristics for every known enemy threat. Managed by the Defense Intelligence Agency and its four intelligence production centers – which test, evaluate, and observe threat systems' electronic parametrics. Provides the foundation for observation and comparison of intentional modifications to any threat radar parameter.
- USELMS (US Electromagnetic Systems) database. Maintains information about friendly systems' radar parameters to enable US and allied radar-warning receivers, electronic-countermeasures sets, and jamming pods to differentiate between friendly emissions and enemy threats.
- EPL (ELINT Parameters List). Contains observed parameter data useful for determining most operating ranges. It includes information on Electronic Intelligence (ELINT) notations, function codes, and associated platforms.
- KILTING (National Technical ELINT Data Base). The National Security Agency (NSA) national technical ELINT parametric database on United States (US) and foreign emitters. This comprehensive database contains the characteristics and attributes of non-communications emitters, as well as the technical signal parameter detail necessary to support ELINT customers.

3.1.4.1.3 Equipment Fit

Equipment fit can be used to infer the platform / site that the candidate equipment (e.g., emitter) might be operating from. It can also be used to infer the platform / base / launcher that an aircraft, helicopter, patrol boat, or weapon came from. This can be useful in inferring mission associations and in adjusting order-of-battle to account for deployed assets and thereby adjust presence potentials. Source for fit are:

- NERF (Naval Emitter Reference File) A collection of Electronic Warfare (EW) data, worldwide in scope and consists of Order of Battle (OOB), friendly (blue), non-hostile (white), and hostile (red) parametric data. Encompasses military, commercial emitters and non-communications emitters. It encompasses airborne, shipboard, missile borne, and land-based emitters of all functions.

- NID (Naval Intelligence Database) and MIDB MEPEDS Provides characteristics and performance data for a wide range of weapons, including aircraft, helicopters, missiles, merchant ships, naval ships and submarine classes.

3.1.4.1.4 Presence Potentials

These data sources can be used to derive a-priori Probability Density Functions (Pdf) for the likelihood of a certain type of object being in a location or area.

- MIDB (Military Integrated Database) GMI (General Military Intelligence). Provides national order of battle, facilities, and unit data. Provides structures for tactically derived raw reporting data and analyst generated local situation versions of national records and targeting data.
- Friendly Order of Battle (FROBDB). Blue data defining the location, condition, and status of friendly forces in theater
- Historical Temporal Shipping (HITS). Maintained by NOAA, HITS gives densities of different types of ships in an ocean grid database by time of year. This can be used as an a-priori source for white shipping presence potentials, that is, as a general Pdf.
- ICAO and FAA Flight Center. These sources report flight plans. Although an aircraft may be off its flight plan some, this is evidence in the ID equation.
- Merchant Ship Database (MSDB). Maintained and disseminated by ONI, this database is a fused source of actual surveillance with voyage plans registered by captains with coast guards, insurance companies, and so forth.
- Airlines Guides. These can be used to infer white AOB, that is, the typical numbers of types of aircraft at various airports. Like any AOB, this is weak evidence.

3.1.4.1.5 Geospatial, Geophysical, and Meterologic.

- Airspace (AIRSPC) Defines the geometric airspace used in a theater
- Flight Corridor databases
- Digital Terrain Elevation Data
- Topography
- Hydrology
- Climatology
- WXDB (Weather Database) Forecast and observational information for the theater of operations or base location

We have done some work already in fitting these reference database models into the C2IEDM-RT, as shown in Figure 12. (In a print copy of this document, the Entity-Relationship diagram will be viewable only notionally. However, in an electronic copy, it can be magnified to readable level.) The legend for this model sub view is as follows:

Turquoise background: added for multi-hypothesis

Lime green background: from NID
White & grey background: from C2IEDM ~ DDA / C2 Core / CADM
Yellow background: requested mod to C2IEDM
Green text: moved, added,

The green fill entities are lifted from the Office of Naval Intelligence (ONI) Naval Intelligence Dataset (NID), the authoritative source for the Navy for Characteristics and Performance (C&P)

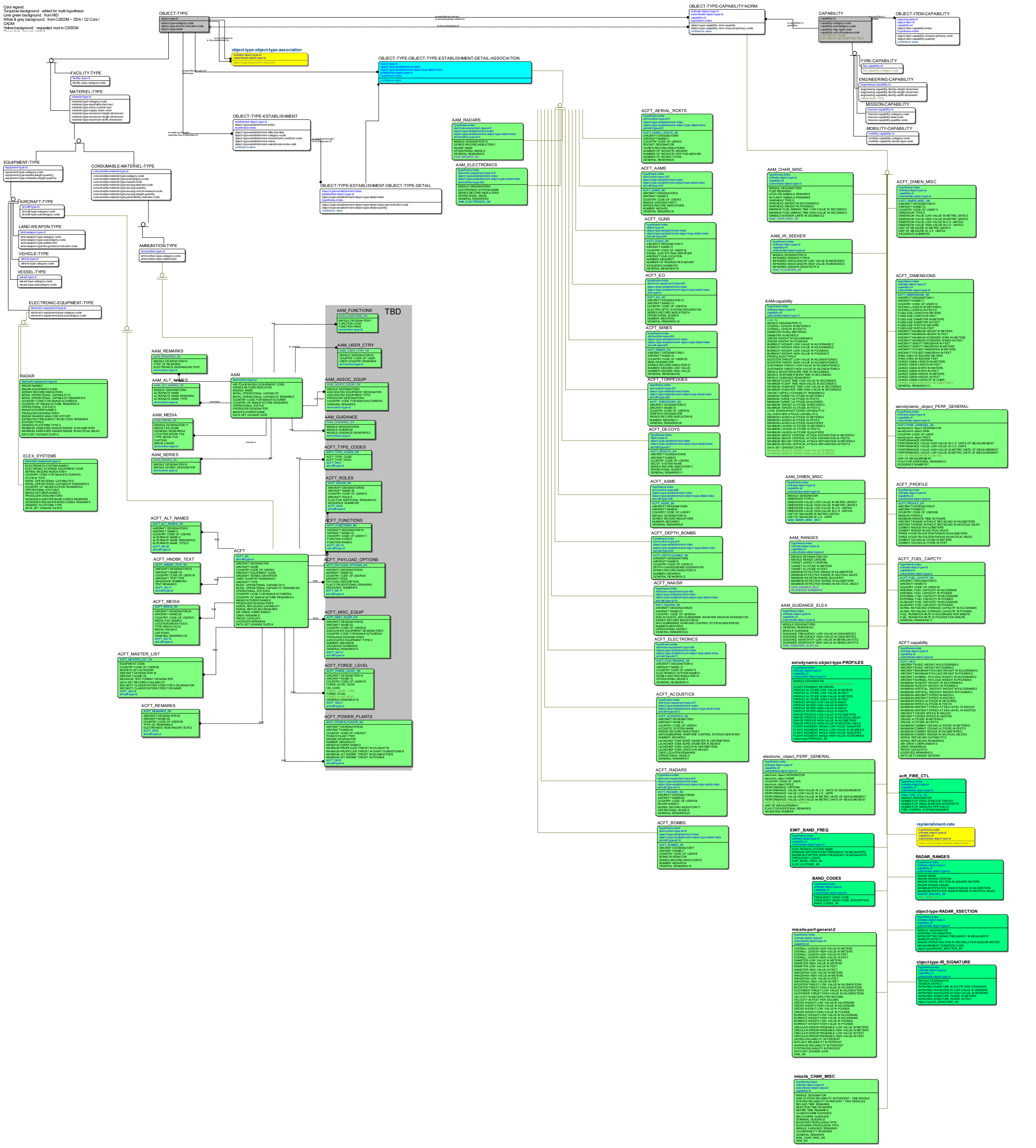


Figure 12. C2IEDM-RT Sub view with NID Aircraft and AAM Tables Integrated

intelligence data. This is just some preliminary work; the NID has hundreds of tables for all kinds of platforms. But the ease with which the Aircraft and AAM tables fit under C2IEDM gives us confidence that all the others can be done too.

3.2 Embedded DBMS

Because we had performed experimentation of the real-time performance of the embedded DBMS prior to the Phase I SBIR effort and those tests were very stressful and had been passed [45], we did not perform additional embedded DBMS performance testing in Phase I. Instead, we advanced to experimentation with running actual fusion algorithms in the ontology OA.

In the previous study, we employed the same type and volume of data access from actual fusion systems deployed either today or in R&D for possible future systems as the basis of our work. The experiments involved searching though the entire track file for candidates. Sequential searching has always been prohibitive, even in applications dependent structures [74][75] Techniques for associative memory (see, for example, [79]) are used wherein the objects of interest are referenced by their attributes instead of their primary identifiers.

In the timing study, the performance drivers of the data manager were the retrieval (or access to) the candidates by the gross gate function and the maintenance of the associative memory. The number of retrieval candidates required varied, depending on the 2-D covariance of the input track report, the density of tracks about the input, and their 2-D covariances. The track sensor update rate characteristics and densities from the system's expected operating environment, the same ones used to determine the system's requirements and later to parameterize timing analyses. We used these to determine the table sizes (number of instances), track input arrival rates, and numbers of candidates expected to derive from the associative memory.

The data structure was made equivalent to that for the deployed system but using a normalized E-R model, as derived from the then U.S. standard for command and control, the Command and Control Core (C2 Core) data model. (Note: That model is equivalent to the C2IEDM Model proposed for this study, and is a derivative of it.) Use of the model facilitated ESM/ELINT Similar Source Integrator (SSI) Classification and Correlation Candidates Retrieval. In the test case, multi-source ESM and ELINT reports were input for a theater-wide area. The experiment was based on an application developed for multi-source ESM and ELINT fusion against theater-level Electronic Order of Battle (EOB). The objective of the system was to correlate the input parameter reports against an EW reference file and the theater order-of-battle. The theater-level OB provided the initial track file loads of all expected but not currently observed tracks. As sensor data was input, the reports were correlated against this a-priori knowledge as well as tracks heretofore received.

All data was maintained in RAM in applications dependent data structures. The arrival rates of ESM and ELINT reports were 5/sec. For each report, the initial lookup involves comparing the report to emitter reference file variables. The EW reference file has around 3,000 types of emitters. Once emitter hypotheses are formulated, then the order-of-battle candidates are retrieved. There are often 30,000 platform and facility candidates corresponding to:

- Naval ship home ports / anchorages
- Aircraft types at airbases and airfields
- Air bases
- Radar and SAM sites

- Headquarters and other military facilities with emitters
- Merchant ships
- Civilian and general aviation
- Current live track file

As a result of the procedures described in the previous paragraph, the MSI and ESM databases were sized and used to populate a set of data structures. This was a very large fusion database compared to most fusion systems in existence today. The procedure also resulted in identification of data update types and rates as shown in Table 4. High and low arrival rates were estimated and the scenario operated for 3600 scenario seconds, enough time to gain associative memory hits for most of the update types.

The results of these preliminary experiments provide good evidence that, as would be expected, high-performance embedded DBMS's have much higher data access times than application dependent data structures. However, the results do show that even under intense scenarios and with massive fusion on a general-purpose medium performance off-the-shelf computer using a non-realtime operating system, the embedded DBMS can perform adequately. The MSI experiment had average sensor updates of 473/sec resulting in requirements to access and average of 3,368 complex object structures per sec. Similarly, the ESM/ELINT experiment had average 112 updates per sec. resulting in the requirement to access 4,512 complex objects per sec. These accesses were against a track, situation awareness, and intelligence database with information on over 130,000 complex objects. With more powerful processors, a realtime operating system, and a distributed computing environment, the embedded DBMS could be expected to perform even better. A key enabler in the experiments was the associative memory. This kind of layering, of application-dependent associative memories, with high-performance embedded DBMS's, followed a conventional off-line DBMS for amplifying display and historical data, was proven to be a feasible architecture for BMC2 application.

Table 4. Track Update Rates and Candidate Retrieval Estimates for Loading Tests

	# Objects	Update Rates						Database Quantities																	
		MSI		ELINT/ESM																	RF Equpment				
		Update Period		AM Hits	Update Period		AM Hits	Aircraft	Aircraft Type	Weapon	Ship	Vehicle	Airport	Seaport	Garrison Site	Missile Site	Electronics Site	other Facility	Aircraft Type - Facility Association	Ship - Facility Association	# per object	Materiel Typical	# Materiel Low	# Materiel High	# Types
Fire Control (JCTN and Fires Net)																									5000
TBM	4	0.25	sec	5	0	0	0			4											0.0	0	0	0	
Supersonic Aircraft and Missiles	50	1	sec	5	4	sec	30	25		25											2.0	50	15	50	
Subsonic Aircraft and Missiles	500	4	sec	5	10	sec	30	450		50											2.0	500	200	500	
Gen Av & Helo	400	4	sec	5	10	sec	30	400													2.5	500	250	500	
Land Mobile	100	4	sec	5	10	sec	50					100									0.3	9	5	14	
Land Fixed	100	10	sec	5	10	sec	20								25	25	25	25			0.5	34	30	37	
Tactical Radar																									
ATC	250	4	sec	10	10	sec	30	225		25											2.5	313	63	313	
Surface & Nav	100	10	sec	5	30	sec	50	8		2	90										4.0	400	360	440	
Tactical Decision Link (JDN)																						0	0	0	
Air	500	10	sec	10	30	sec	50	450		50											2.0	500	200	500	
Surf	250	30	sec	10	1	min	100				250										4.0	600	0	0	
Sub	15	30	sec	5	30	sec	30				15										0.5	2	0	0	
Land Fixed	400	10	min	5	10	min	50						40	10	100	75	100	75			0.3	68	34	68	
Land Mobile	200	30	sec	20	60	sec	200					200									0.3	19	9	19	
Operational Decision Net (JPN)																									
Surf	500	1	min	40	1	min	50				500										4.0	1200	0	0	
Sub	30	5	min	5	1	min	50				30										0.5	3	0	0	
Land Fixed	600	12	hr	30	1	min	30						70	30	100	100	200	100			0.3	101	0	0	
Land Composite Mobile	300	1	hr	40	1	min	100					300									3.0	338	56	338	
Merchant Shipping	10000	1	wk	40	1	wk	40				10000										1.0	5000	100	5000	
Order of Battle Net																									
Airbases	400	1	wk	20	1	wk	30						400						2400		2.0	800	800	800	
Aircraft Types per Airbase	6	1	wk	200	1	wk	100		1000																
Naval Ports & Anchorages	100	1	wk	20	1	wk	30							100							1.0	100	100	100	
Ship / Boats per Naval	24	1	wk	100	1	wk	100				2400									2400	4.0	9600	9600	9600	
Electronic Sites	4000	1	wk	20	1	wk	30										4000				2.0	8000	8000	8000	
Military Ground Sites	7000	1	wk	20	1	wk	30								7000						0.3	1750	1750	1750	
Missile and Gun Sites	3000	1	wk	20	1	wk	30									3000					1.0	3000	3000	3000	
Civil Aviation Flight Plans	200	4	hr	20	0	hr	0	200																	
Airport Flight Scheds - Airports	50	1	wk	10	0	wk	0						50												
Active Aircraft Flight Scheds per Airport	100	1	wk	30	0	wk	0	5000																	
		473		3368	112		4512	6758	1000	156	13285	600	560	140	7225	3200	4325	200	2400			32886	24572	31027	5000
		Updates and AM Hits Per Sec						**Table Sizes**																	

3.3 Tracking Filters

We conducted two tracking filter experiments. In both of these the subset of the ontology used was very small, the purpose of the experiments being to see what was involved in conforming existing software to the architecture. The ontology fragment implemented in the embedded DBMS is as shown in Figure 13. In these experiments, a simple simulator generates radar processor output reports in the form of Range and Azimuth data. As these records are added to the RECEIVER-OUTPUT table, the tracker filters are triggered to execute. Note that in both the RECEIVER-OUTPUT and KINEMATIC-STATE tables, records are never over written but are only added, by virtue of the design which placed time as an identifying key attribute, part of a composite key. Although ATDS today does not maintain this temporal dimension, there are benefits that may be worth considering, now that computing resources allow the option. For the

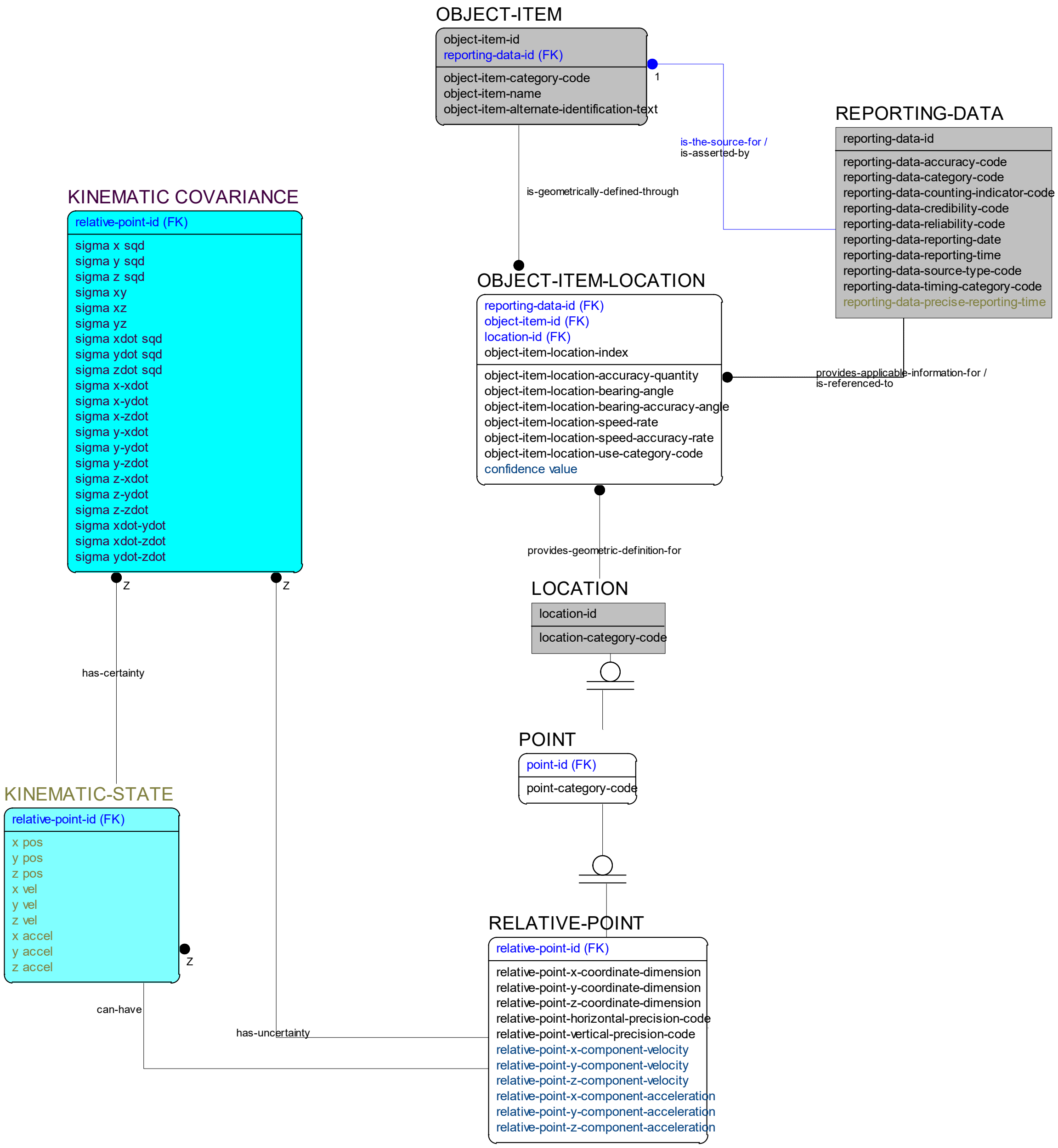


Figure 13. Ontology Fragment Used for Tracking Filter Experiments

purposes of the experiments, the temporal dimension allowed us to look at the tables after the scenario runs.

3.3.1 "1090" Tracking Filter Experiment

The "1090" tracking filter owes its name to Navy Computing Center report number 1090, written in 1965. It was one of the first tracking filters ever developed and is still in limited use in the US Navy today, for processing of TPX-42 ATC radar reports. As such, it was a god 'acid' test in a way because it was so unconventional. For example, since it pre-dated Kalman and even $\alpha\beta$ filters, it did not have a persistent covariance matrix in the form used almost universally today. So some persistent storage of the non-standard filter data was required. We view this as an anomaly, caused by the extraordinary obsolescence of this software, and were satisfied by the fact that there was a remedy, namely, allowing the software to persistently maintain some of the data it maintains anyway. While slightly inelegant, we don't believe this contingency will have to be exercised often.

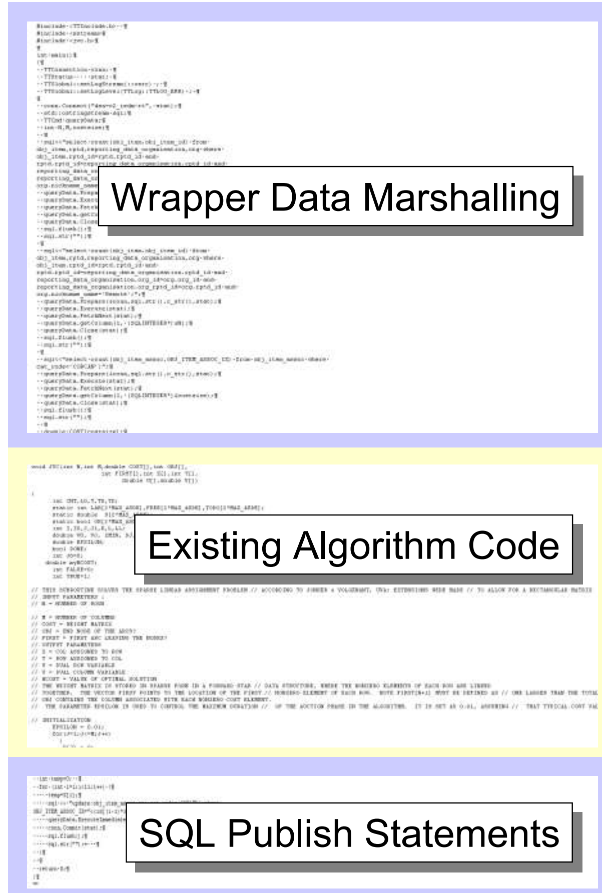


Figure 14. NGFA Wrapper Framework

The code we used was from the Ship's Self Defense System (SSDS) Mk-2 and is shown in Appendix B. Because this code was used in a test environment to develop the actual SSDS code, it had file loaders and outputters at the beginning and end that we had to delete. Then we had to build a 'wrapper' that would respond to the DBMS trigger and translate the data from the ontology to the format the legacy code required. This architectural "framework" [[68]] is shown in Figure 14. The results were that this worked, the simulator filled the RECEIVER-OUTPUT table, as each record was added, it triggered the wrapper which executed the 1090 code which the wrapper then formatted into the KINEMATIC\-STATE table.

3.3.2 Dynaest Filter

For this experiment, we wanted some code for a modern tracker and from a very different development environment. Yakov Bar-Shalom's Dynaest library [8] from the University of Connecticut was about as different from the 1090 experiment as could be. Not only are the filters from the world-renowned fusion scientist, they are programmed in MATLAB, a rapid development environment for mathematical code. Among MATLAB's many features, one that is extremely convenient for filter code is its linear algebra library.

For this experiment, we used the same ontology fragment as for the 1090 (Figure 13) but because this filter is modern, the KINEMATIC-COVARIANCE table was also used. There was no persistent storage in this filter, every input and output was maintained in the DBMS. Also, the MATLAB code was modularized so that it was not necessary to modify the rehosted element at all. The off-the-shelf code is shown in and the wrapper is shown in Appendix B. Its architecture is simpler than the 1090's in that no modification of the off-the-shelf code was required. This experiment was also a success and the table contents after the scenario run are shown in Appendix B. We were pleased to see that this modern filter was much easier to adapt to the C2IEDM-RT and required far less wrapping. We might expect this often, that more modern processing elements adapt more easily to the ontology than do legacy components.

3.4 Assignment Algorithms

Assignment algorithms are used in both association (trying to figure out what measurements correspond to what tracks0 and correlation (trying to figure out what tracks from two sources are in fact the same). An assignment algorithm aims to match one set to the another, as TBS depicts. Some, such as early nearest-neighbors, assign pairs based upon pairwise affinity and a sometimes elaborate decision logic; others attempt to minimize a global cost. We experimented with one of each for the local-remote correlation problems, wherein the ATDS tries to figure out if tracks on the Link-11 and Link-16 nets (remote tracks) are the same as those being tracked by the APS-145 and CEC (local tracks) The ontology fragment used for these experiments is shown in Figure 15. Note the elegance of this structure in that current data structures in Navy systems use two files, one for remote and one for local whereas this ontology maintains according to the type and structure of the data (e.g., kinematics) and maintains the source as an attribute. This is a more general data architecture that could support many sources, not just “local” and “remote”. The assignment matrix (in the case of JVC) and the “Y-NOT’ links (in the case of SSDS) are both maintained in the OBJECT-ITEM-ASSOCIATION structure which supports showing a relationship between two OBJECT-ITEM instances, in this case with an association type code of “correlated”. The association confidence value, the same attribute used for hypothesis confidence throughout the ontology, maintains the cost (JVC) or probability (SSDS). In a final architecture, a standard confidence

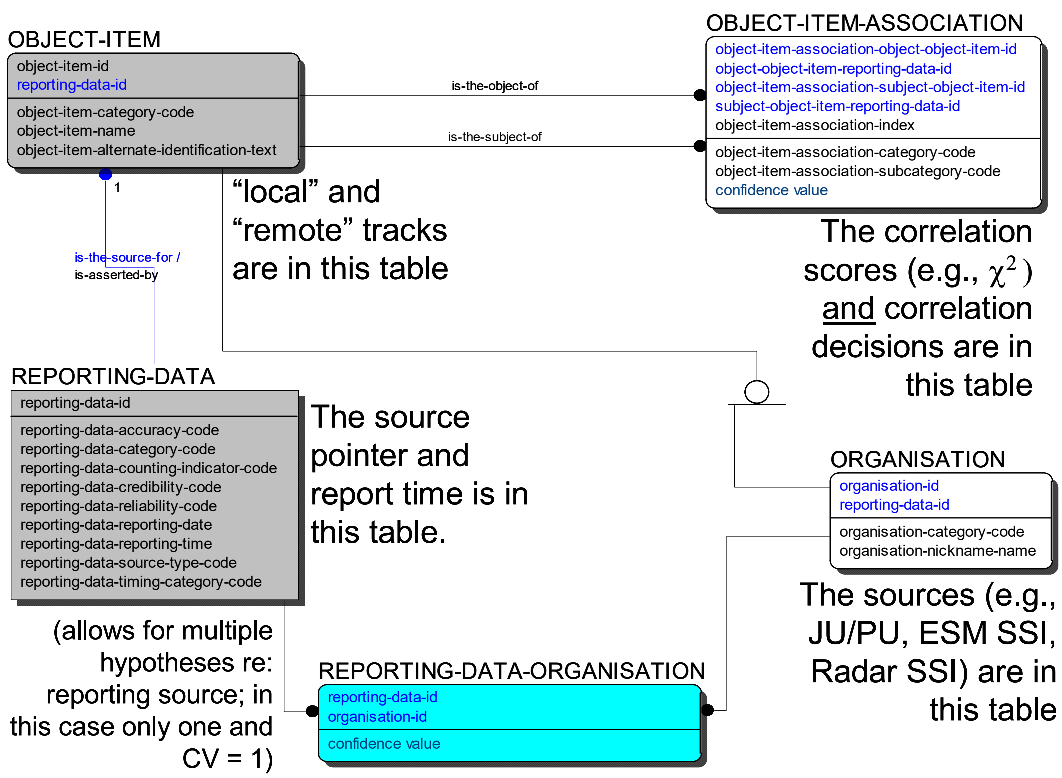


Figure 15. Ontology Fragment for Assignment Algorithm Experiments

value would be defined and all interacting algorithms would have to have their wrappers do any de-biasing or translation of correlation scores, costs, etc. to the confidence value standard.

3.4.1 JVC

Like the 1090, the JVC [40] code provided by Dr. Oliver Drummond had file readers and outputters on the ends that had to be eliminated. However, from that point on, the conformance was simply a matter of data translation since all the persistent data was maintained in the ontology. The translation was as shown in Table 3-5. Note that the sparse matrix handling in the original code is handled very elegantly by the associative data structure in the ontology and yet achieves the same goal, that of storing only the sparse feasible costs and not the entire matrix. The difference is that the DBMS can lookup the sparse entries given the indices because it uses state of the art hashing and indexing to lookup all data. In effect, the DBMS has sparse matrix handling as a special case of its more general handling of all sorts of lookups.

Table 3-5. JVC – Ontology Data Translation in Wrapper

JVC Data Elements		**Ontology Data Elements**		
Name	**Definition**	**Table**	**Field**	**Translation Comments**
N	Number of rows where a feasible solution means every row is assigned to a column	OBJECT-ITEM		Number of records whose record tracking (or reporting data) source is "local"
M	Number of columns where a feasible solution means a column is assigned to no more than one row, i.e., it can have no assignment.	OBJECT-ITEM		Number of records whose record tracking (or reporting data) source is "remote"
COST	The input costs in a one-dimensional array, due to the sparseness of the matrix	OBJECT-ITEM-ASSOCIATION	Confidence value	For all with association type code of "candidate correlation"
OBJ	A one-dimensional array parallel to the COST array which has the index of the column corresponding to the COST entry	OBJECT-ITEM-ASSOCIATION	subordinate-object-item-id	1. Translate the id key to 1-M 2. Association type code is "candidate correlation"
FIRST	A one-dimensional array corresponding to each row in the virtual (sparse) cost matrix where each row entry points to the start of data for the row in the COST and OBJ one-dimensional arrays	OBJECT-ITEM-ASSOCIATION		For each object-item, will the sum of the previous object-item's numbers of subordinate-object-items.
X	For each row, the index of the column assigned to the row	OBJECT-ITEM-ASSOCIATION	subordinate-object-item-id	Association type code is "correlated"
Y	For each column, the index of the row assigned to the column	OBJECT-ITEM-ASSOCIATION	ordinate-object-item-id	Association type code is "correlated"
U	A dual variable, the 'dual price' of row I			specific to algorithm, not persistent
V	A dual variable, the "dual price" of column J			specific to algorithm, not persistent

The cost matrix inputs were generated using the two trackers in the prior experiments. Each tracker is run in succession, as if the 1090 were the remote tracker and the Dynaest the local tracker. The RECEIVER-OUTPUTs have bias applied to simulate navigation and sensor errors typical of gridlock and sensor data registration seen in the Fleet, that is, lat, long, and azimuth offsets and a multiplicative range scale factor. In addition, a probability of detection is applied based on a hypothetical range to target and a random function. As well, the biases are randomly varied. Again, this experiment was a success, as the code was conformed to run against the ontology. The output results are shown in TBS.

At this point it is worth again noting the temporal dimension of the ontology since now it allows for backtracking of reasons-why for correlation decisions. This can be very important in aberrancy removal. While the goal is to have as few correlation aberrancies as possible, they are inevitable. The temporal dimension could provide a more elegant way to handle than, say, the complex logic embedded in the Tomahawk control system, the only Naval tactical system to attempt a rigorous aberrancy removal. Many combat system elements impose a hysteresis filter to stabilize decision making but when decorrelations or change correlations occur, make no attempt to re-do fusion decisions and estimates that may have been contaminated by the error.

3.5 **ESM / ELINT Classification (Phase 1 Option)**

ESM / ELINT classification is a component of multi-source target identification for targets that have radar emitters. In the Phase I Option, we experimented with automated ESM / ELINT classification requiring inferences across several nodes in an ontology. Previous experiments involved inferring across two nodes. The ontology subview for the multi-nodal experiment is shown in Figure 16. The ESM / ELINT Similar Source Integration (SSI) process makes inferences over this ontology as suggested by the overlay in Figure 17. We used algorithms, code, and data structures from a prior SPAWAR SBIR seeking to improvide ESM / ELINT SSI for GCCS-M and tactical C&D systems.

Why is a fusion algorithm necessary for ESM / ELINT classification? For years, systems like the Navy shipboard SLQ-32, fighter aircraft Radar Warning Receivers (RWR), and the E-2's Passive Detection System (PDS) have been using table lookup functions to classify emitters.

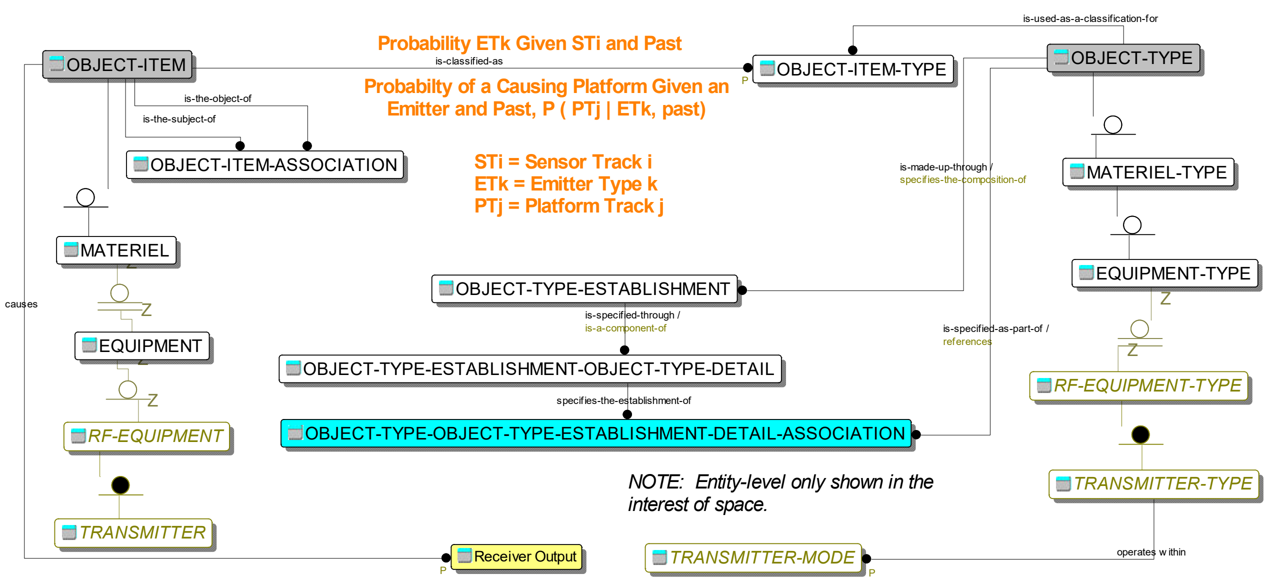


Figure 16. Ontology Data Model Entity Level

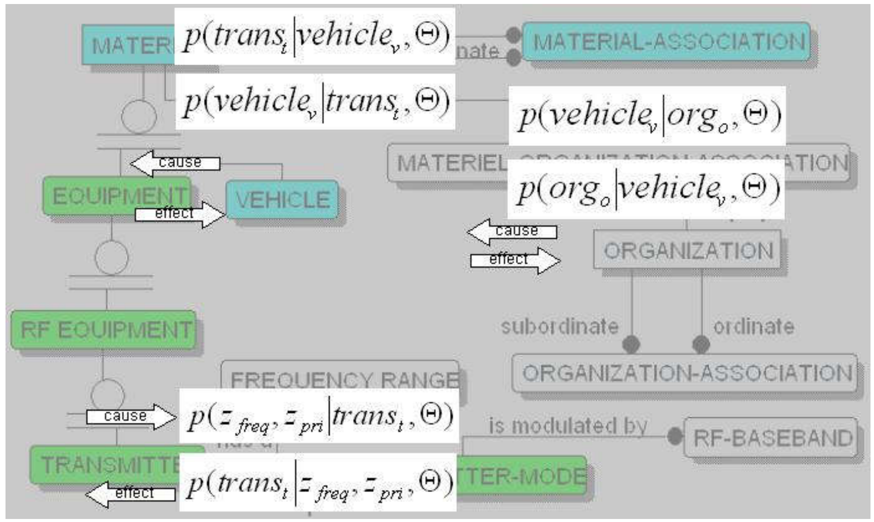
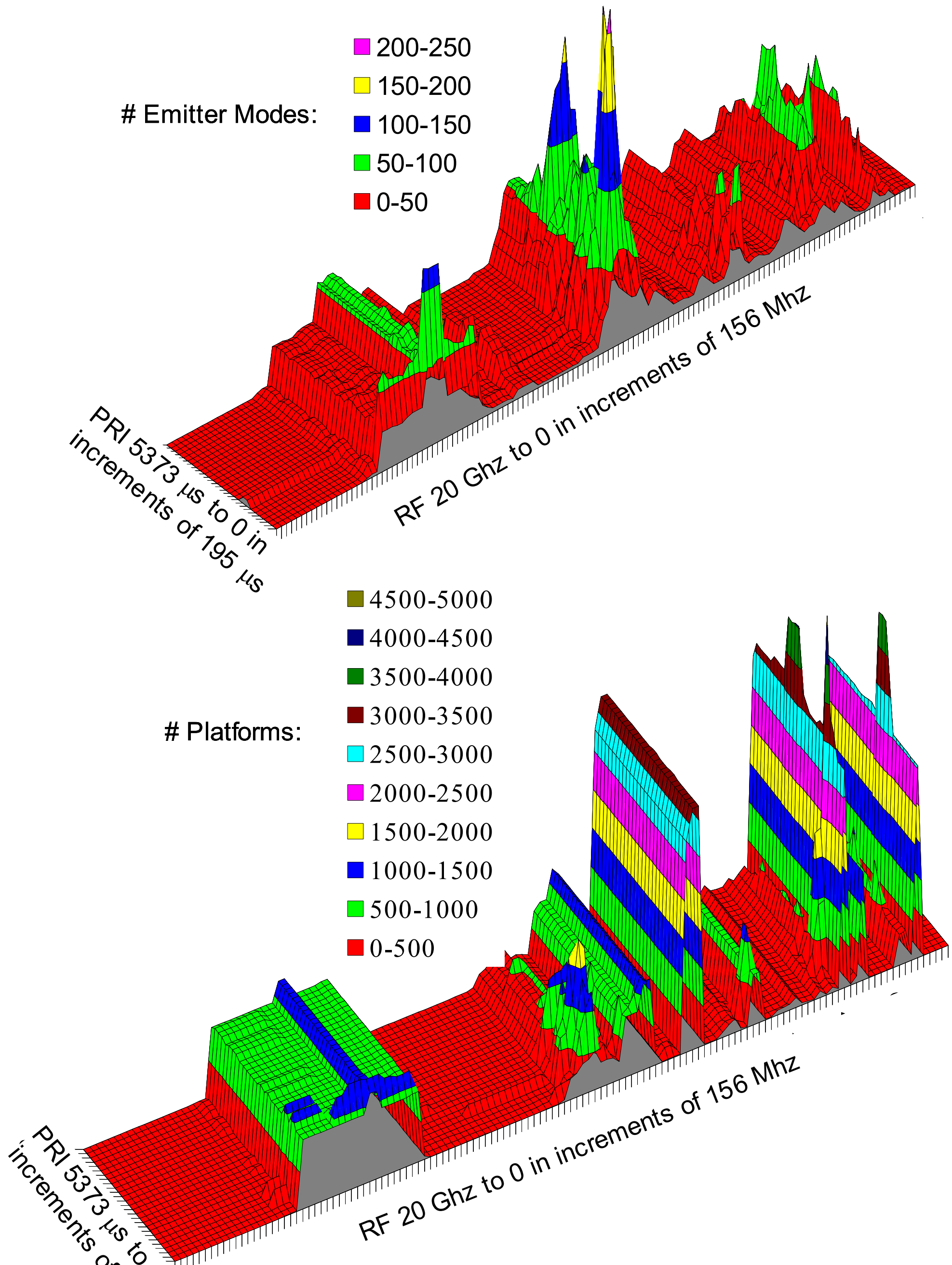


Figure 17. ESM / ELINT Classification Experiment

However, in many cases lookup is not sufficient for multi-source identification because of ambiguity. Defensive systems like SLQ-32 and fighter aircraft RWRs use small EW libraries to reduce ambiguity because their receivers are tuned to threat frequencies and signals that are detectable in the vicinity of the defensive platform. The notion is that something emitting certain waveform characteristics detected nearby is probably a problem. When ESM and ELINT are used for Battle Management, Command and Control, and Situation Awareness, the ambiguity increases because a higher number of tracks are considered at much longer ranges. Signals are used for evidence beyond fire control and seekers (e.g., navigation radars, altimeters, surveillance), and the supported operational function requires a considered (not a reflexive) response. To get an idea of the magnitude of ambiguity, see Figure 18. For Long Range Full Target Type Multi-Source ID, ESM / ELINT classification may require discrimination among over 200 possible emitter modes or over 4500 potential platform types – for a single signal (specific combination of PRI and RF). The analysis summarized on Figure 3, against the Naval Emitter Reference File (NERF), shows that simple lookup will often result in extreme ambiguity. Therefore, some additional evidence utilization is necessary to rank and eliminate candidates.

Human analysts reduce ambiguity using a wide assortment of clues and knowledge bases in an inferential reasoning (e.g., deductive, abductive, probabilistic) process that is akin to detective-type reasoning. Many aspects of this process cannot be mimicked by artificial systems at the current state of the art. The human knowledge base is vast, drawing upon years of accumulated information and associations. Also, reasoning methods and their combinations and pattern processes are not fully understood. However, artificial computation can provide enormous aids to, and for some processes, substitution for human SIGINT correlation and fusion. This arises from the ability to consider available knowledge bases thoroughly and perform massive amounts of precise mathematical computations.



Platforms	Types of Objects
Electronic Order of Battle (EOB)	specific SAM site
Ground Order of Battle (GOB)	specific headquarters building
Missile Order of Battle (MOB)	specific silo
Naval Order of Battle (NOB)	specific hull
Air Order of Battle (AOB)	specific airbase
Aircraft	aircraft model to alpha modifier of fighter
Weapon	model to alpha modifier of missile

Figure 18. ESM / ELINT Ambiguity for Long Range Full Target Type Multi-Source ID

Bayesian networks provide techniques for automating probabilistic inferences. Traditional expert systems are extensional, with all the information for propagation locally available from local immediate antecedents. This makes them tractable when creating knowledge bases and efficient when computing over them at run time. However, these benefits come at a significant cost in intelligent reasoning power. Consider the case of a signal 1 (s_1) emanating from a target for which the parametrics match reference database min/max intervals for emitter type a (et_a) and emitter type b (et_b) on platform types x and y (pt_x and pt_y), respectively. In a traditional expert system, et_a and et_b and pt_x and pt_y would be activated according to the strength of the sensor information and pre-determined propagation formulas, regardless of information from other sensors. For instance, IFF, SAR/ISAR, or other signal information (s_2, s_3, …) may be associated with the target that could render pt_y less likely. The rules $s_1 > et_b > pt_y$, which on the surface are analogous to $P(pt_y| et_b)$ and $P(et_b | s_1)$, cannot convey $P(pt_y | et_b)$ and $P(et_b | s_1, s_2)$. The probabilistic, or intensional scheme, however, becomes intractable if one has many dependent clue types since n! joint probabilities need to be specified. Bayesian networks make the probabilistic method tractable through intermediary nodes that convey all the dependent joint probability information while localizing the considered clues as is done in the extensional systems.. Multi-source correlation, which has a high degree of intensionality, is also supported. More than just spatial information needs to be considered. For example, detection of a signal that is likely to be et_a known to be installed on pt_x that also carries et_b increases the probability of a correlation when a signal is detected that is likely to be et_b. Conversely, the likelihood of the correlation influences the probability of the second signal being from et_b. Or consider the case when the SIGINT reports are instantaneous or contact reports, not track reports, that must be tracked to smooth out noise, estimate other state variables not directly measured (e.g., velocity), and allow time extrapolation (backward or forward) for time-synchronous correlation comparisons. If the measurements are bearing-only, typical for ESM and many SIGINT systems, the platform range can be estimated based on likely emitter and platform types using known emitter power ranges, seeker turn-on ranges, known platform altitude envelopes, horizon limitations, etc. However, the tracker outputs will influence the identification results. More accurate and complete knowledge of what something is contributes to making decisions about what things go together and how, in fact, how things may behave and, possibly, where they might be expected to go. The converse is also true: knowledge of what things go together, how they are behaving, and where they are often aids identification estimates performed by humans (i.e., exploitation of kinematics and behavior for identification). This type of circular dependency is difficult, if not impossible, to handle in standard extensional systems.

3.5.1 ESM / ELINT SSI Algorithm Narrative Overview

The design is as shown in Figure 19. First, input EW classical parameters (RF, PRI, etc.) reports are normalized to a common "superset" format, the Sensor Track (ST)

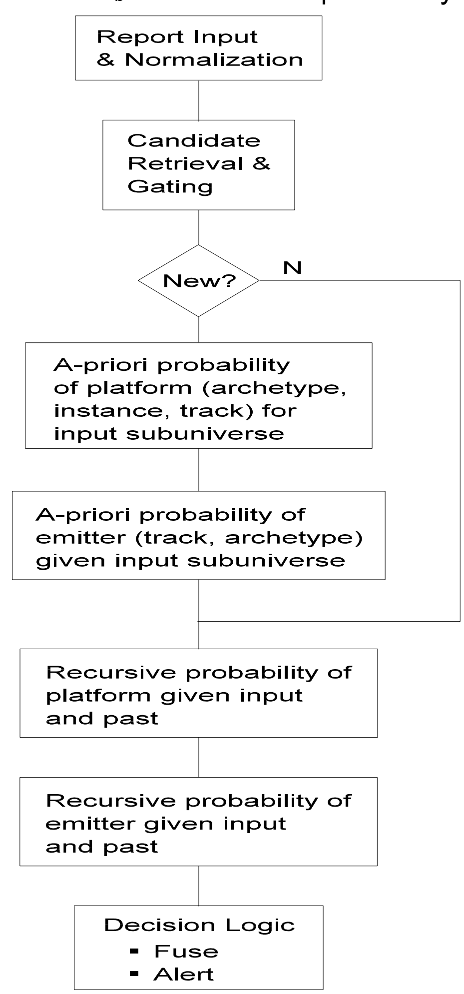


Figure 19. ESM / ELINT SSI Algorithm Design Overview

format. In this process, a model of the sensor characteristics is used to estimate variance parameters not provided by interfacing systems. Next the EW parameters are used with an associative memory to identify candidates without searching the database. This scheme allows near instantaneous mode candidate determination. The candidates are further pruned according to compatibility of discrete parameters.

The next 6 steps accomplish the recursive Bayesian net. If the report is not indicated by the sensor system or an association process as corresponding to a previously reported track, a-priori values must be computed. The “a-priori’s” are computed dynamically to gain the added benefit of using more localized probability universes; due to the large surveillance ranges that might be required, having perhaps vastly different region-by-region a-priori's. Also, since the OB is dynamic, not static, the a-priori’s evolve, in a sense are "learned" as the algorithm runs. Dynamic a-priori's are computed for platform and emitter candidates using OB, Characteristics and Performance data, and other parameters. The next steps are recursive probability calculations using the just-computed a-priori's and the values in the track file that were computed on a previous update cycle. In addition to OB, C&P, and the other parameters, this process uses the EW library and interpretive models of the meaning of the library parameters.

The following subparagraphs provide a narrative description of the algorithm.

3.5.1.1 Identification Estimation

Much more emphasis has traditionally been placed on measurement and reasoning about kinematics, and much less less has been place on identification. Moreover, the two are typically treated as wholly separate processes – even though it is clear that an objects speed, location, trajectory, etc yield important clues about what the object might be. This design combines both kinematics estimation and identification. The goal is to reduce ambiguity through the full use of all available clues.

Owing to the success of state estimation theory in target tracking and other applications, we researched analogous formulations for the identification problem. State estimation typically addresses continuous state variables defined in metric space [5]. In [56], the notion of defining a metric on the discrete identification space was introduced, as part of a way to try to define a measure of performance for multi-source target identification. Identification vectors are analogous to continuous variable state estimates and covariances. For example, consider a single dimension continuous variable. Its state estimate and 1-element covariance matrix convey the same type of probabilistic information as an identification vector. In fact, the continuous variable state estimate and covariance could be approximated as a probability

Environment / Category

$$\begin{bmatrix} 0.9 \\ 0.1 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \left\{ \begin{array}{l} air \\ surf \\ sub \\ land \\ space \end{array} \right.$$

Platform

$$\begin{bmatrix} 0.20 \\ 0.10 \\ 0.09 \\ 0.01 \\ 0.00 \\ 0.00 \\ \vdots \end{bmatrix} \left\{ \begin{array}{l} fighter \\ bomber \\ attack \\ transport \\ AEW \\ EW \\ \vdots \end{array} \right.$$

Specific Type

$$\begin{bmatrix} 0.00 \\ 0.33 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.10 \\ 0.00 \\ \vdots \end{bmatrix} \left\{ \begin{array}{l} F5 \\ F14 \\ F15 \\ F16 \\ F18 \\ F111 \\ F117 \\ MIG21 \\ MIG23 \\ MIG25 \\ MIG27 \\ \vdots \end{array} \right.$$

Type Modifier

$$\begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \\ 0.00 \\ \vdots \end{bmatrix} \left\{ \begin{array}{l} F14A \\ F14A+ \\ F14C \\ F14D \\ \vdots \end{array} \right.$$

Figure 20. Identity Vectors

vector by defining each vector element to correspond to an interval in the variable space and with probability values in the vector corresponding to the probability mass in that interval. In other words, there are some parallels between the mathematics of these identification vectors and the standard mathematics of state estimation theory. The ESM / ELINT classification algorithms produce an identification vector with ranked probabilities for emitter and platform type as shown.

3.5.1.2 Recursive Bayesian Net Inference Algorithm

The ESM / ELINT classification algorithm is the recursive Bayesian algorithm applied as a series of Bayesian Net links using the an actual EW parametric database and the Order-of-Battle (OB) database. Bayesian nets are the state-of-the-art in probabilistic inferential reasoning (i.e., machine thinking). This provides a recursive maximum a-posteriori probability computation exploiting a-priori INTEL information such as OB and C&P and refining estimates over time and over multi-sensor ESM/ELINT updates. Bayesian techniques have been known as applicable to AI nearly since its inception. However, they were considered intractable for complex applications, requiring knowledge of too many joint probability distributions [14][6]. The Bayesian net provides a methodology for modeling the probabilistic dependencies in the real-world problem space, thereby often enormously alleviating the requirements for joint probability knowledge.

The Bayesian net conforms with the ontology architecture because it is based on an ontologic model of the targets, physical phenomenologies, and measurements and resembles human identification thinking. Additionally, the recursive algorithm is the discrete non-metric-space analog of a zero-process-noise Kalman filter [37], thus generalizing the estimation problem. The recursive probability approach also has the desirable property that ambiguity decreases monotonically with updates. Also, Bayesian nets provide a basis for the representation of explicit knowledge, unlike techniques such as neural networks which are used to represent knowledge which is not explicit (i.e., pattern recognition). This ontology in effect hosts a Bayesian net by attributing probability formulas for traversals through the net. The relationships encoded in the NERF (or other EW parametric and OB databases) are ideally represented as Bayesian nets because Bayesian nets faithfully represent the dependencies between variables.

3.5.1.3 A-Priori Intelligence and Surveillance Fusion

Current ESM and C3I systems maintain Order-of-Battle (OB) as a semi-static file, independent of the track file (e.g., GCCS). This separation can result in inconsistencies in the systems knowledge bases as the encyclopedic pre-deployment OB becomes asynchronous with the surveillance track file. The ESM / ELINT classification approach combines the OB file and track file into one coherent file using the schema shown previously at the entity level in Figure 16 and below at the attribute level in . In this architecture, the track file is initialized with OB data. Part of the challenge was to find a common conceptual data model for OB, composed of NOB, AOB, EOB, aircraft, weapons, and their relationships, and surveillance tracks. Some OB elements, aircraft and weapons, are archetypal, not actual instances of aircraft or weapons. Aircraft archetypes are related to the actual individual aircraft via linkages to airbases, with the linkage indicating how many of that type of aircraft are typically located at that airbase. Then correlated surveillance reports are used to update OB data or instantiate OB archetypes. This has had enormous implications from knowledge engineering, inferential reasoning, and software engineering vantages.

This approach furthers the blurring of the distinction between INTEL and surveillance, a trend that increasingly benefits warfighters. In this case, the blurring is due to the fact that OB is based on some form of reconnaissance, analysis, or other intelligence surveillance at some point in time. INTEL is treated as merely previous surveillance. By capitalizing on the vast investment in everyday INTEL RECON, S&T analyses, etc., the track picture is initialized with a comprehensive knowledge of the theatre of operation. Correlating surveillance against this a-priori track file is equivalent to identifying targets to a defined taxonomy since the OB defines the identification universe. This approach is flexible and has many features that can be grown into such as:

- Tactical EOB. A dynamically adaptive EW "library" and EOB. Conventional ESM systems use pre-engagement EW libraries to identify contacts. This architecture supports use of received ELINT as well as pre-engagement INTEL. The use of ELINT provides the parameter ranges actually being used by a platforms emitter instead of the generic ranges found in libraries. Also, use of ELINT provides better locational information over the general locations available via OB so that the more informational maximum a-posteriori likelihoods can be used instead of the conventional maximum likelihoods.
- Order-of-Battle Accounting. The INTEL database is the initial track database, with sensor reports used to discover or account-for the pre-engage INTEL.
- Inheritance of archetype properties on instantiation even if there is no current sensor data detecting those properties. This allows for recognition of those properties as potentially belonging to the instantiated track if they do manifest themselves.
- Kinematic scoring between surveillance and OB kinematic information. This is made possible a knowledge structure using C&P and other parameters to transform OB data to surveillance expectations.
- New track hypotheses can be based on local region populations suggested by OB that have not yet been accounted for by previous surveillance reports. This consideration allows the a-priori probability of target types (e.g., Mig-29, Mirage F-5) to vary geographically according to OB and other parameters. In currently deployed systems, the expected target density is static and uniform over the entire surveillance volume. This architecture allows to vary in time and by location and by non-kinematic parameters, to better model knowable information. OB is kept in sync with the track file, it updated tactically, and is used as a universe for which surveillance reveals expected items. This is a higher fidelity model than conventional likelihood methods which use default uniform target density values for the entire universe of operation.

3.5.1.4 ESM/ELINT/OB Fusion Knowledge Engineering

A big challenge in developing multi-source identification systems is the large amount of implicit knowledge, subconscious processes, and poorly articulated inferences used by humans in making identification. Identification estimation as done by humans is not strictly formulaic. Human tactical analysts do not make fusion decisions based merely upon the sensor inputs. In subtle ways they consider INTEL, target characteristics, known adversary tactics, battle condition, etc. Humans rather easily perform "approximate reasoning". That is, they have the ability to reason with uncertain data and vague concepts, and to determine patterns in noisy/incomplete environments. There are various approaches used to emulate different aspects of such reasoning, including probabilistic reasoning (e.g., Bayes, Dempster-Shafer, Generalized Evidential Processing Theory), fuzzy logic, automated pattern recognition via neural networks, etc. No single technique "solves the problem," but several may be used in combination to address a specific problem. As a simple example, consider the following situation. After a receiver output event, a table lookup is executed yielding a list of 3 possible

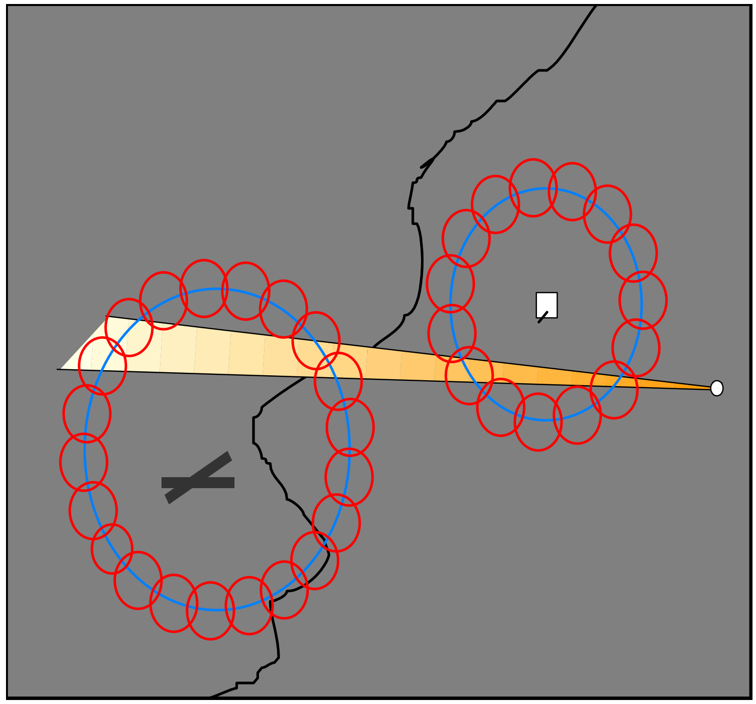


Figure 21. Weapon -> Launching Aircraft -> Home Base Linkage and PDF Estimation

radar units that could have generated the received signal. At this point, we assume the 3 possibilities are equally probable. Each of the 3 radars is represented in our system as materiel-item types. Each is associated with some platform known to utilize that type of radar, via the is-part-of relation. So R_1 is-part-of P_1, R_2 is-part-of P_2, and so on. Thus we now assume, again with equal probability, that one of 3 platform types is present. The system also has information about military bases in the area of operations. Military bases and carriers are represented as establishment types, and are associated with platforms known to exist on them, via the is-holding relation. Suppose the only base in the area, B_1 is-holding platforms of type P_1, and is not holding platforms of type P_2 or P_3. This raises the probability that the receiver output event was caused by a radar system of type R_1, which is-part-of platform P_1, which is possibly on sortie from base B_1. It also lowers the probabilities associated with platforms P_2 and P_3 as illustrated in Figure 21.

Databases exist that convey aspects of this information, most of which are within the "umbrella" of the Navy Warfare Tactical Data Base (NWTDB) database standardization program. These include the Naval Emitter Reference File (NERF), Navy Intelligence Dataset (NID), Military Integrated Intelligence Data System (MIIDS) Integrated Data Base (IDB), DMA DAFIS air routes, Joint Munitions Effectiveness Manual (JMEMs), and others such as JOPES, SORTS, and so forth. As new INTEL or surveillance manifests itself, the ontology in-effect update these databases by virtue of their inclusion in the track file and the consequent updates there. As more exotic knowledge is required for fusion, knowledge representation becomes a challenge. Technologies such as fuzzy sets could provide expressive power to the existing database technologies for data dictionaries (e.g., fuzzy data element definitions for such status as unit morale).

The ESM / ELINT classification algorithm encodes INTEL knowledge bases as semantic nets upon which Bayesian net mathematics are attributed. The algorithm was built upon an organization of the EW track files in three tiers as shown in Figure 22. The intercept/sensor level, Sensor/Intercept Track File (STF), corresponds to a reported sensor track or a tracked set of contacts from a reporting source. The STF also includes the EW "library" emitter modes. Combining library and sensor data in a single file allows a single process to match new reports against previously received reports and pre-engagement INTEL and for the EW "library" to dynamically adapt to the real-time tactical electronic situation. An underlying concept in this is the treatment of pre-engagement INTEL as previous surveillance, a concept that introduces powerful new reasoning capabilities into the data fusion problem. Representation of the relationship between knowledge base elements is important and the ontology provides a flexible tool for representation in a high fidelity manner. For an inference point of view, Bayesian net mathematics can be overlaid on the ontology, thereby creating not only a knowledge representation, but also a reasoning method and a way to manage uncertainty.

The Emitter Track File (ETF) is the consolidated representation of the fused STF's into emitters. This is the first level operators would normally be interested in. The Intermediate Track File is the consolidated representation of the fused ETF's into platforms. The term "intermediate" is used to convey that cross-discipline fusion to other INT's (e.g., COMINT, IMINT, RADINT) is still to be performed to create the final all-source fused track.

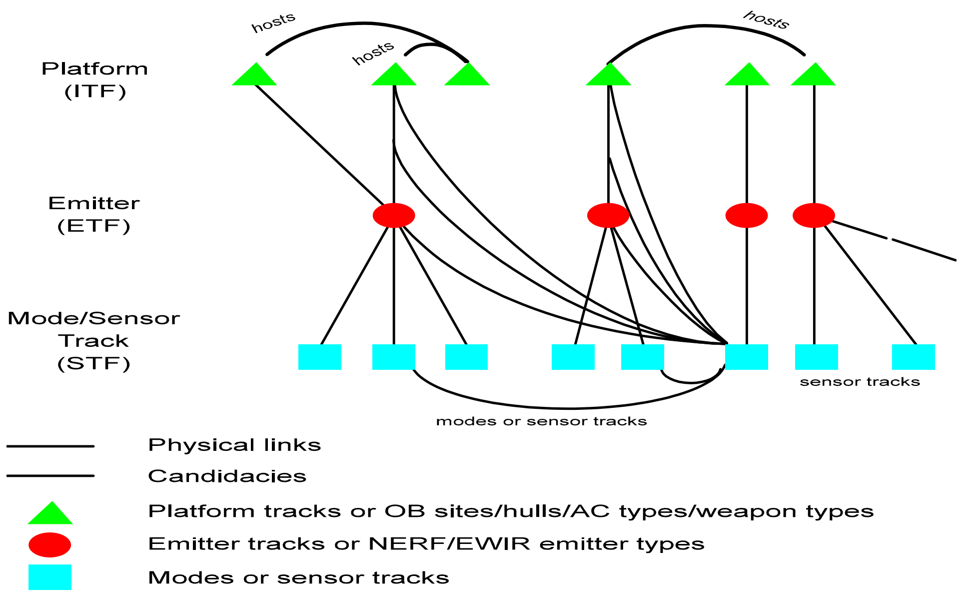
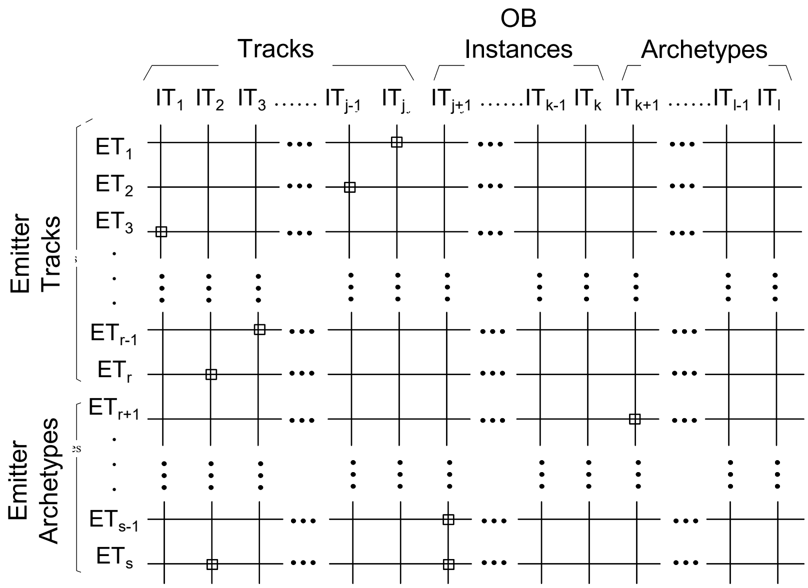


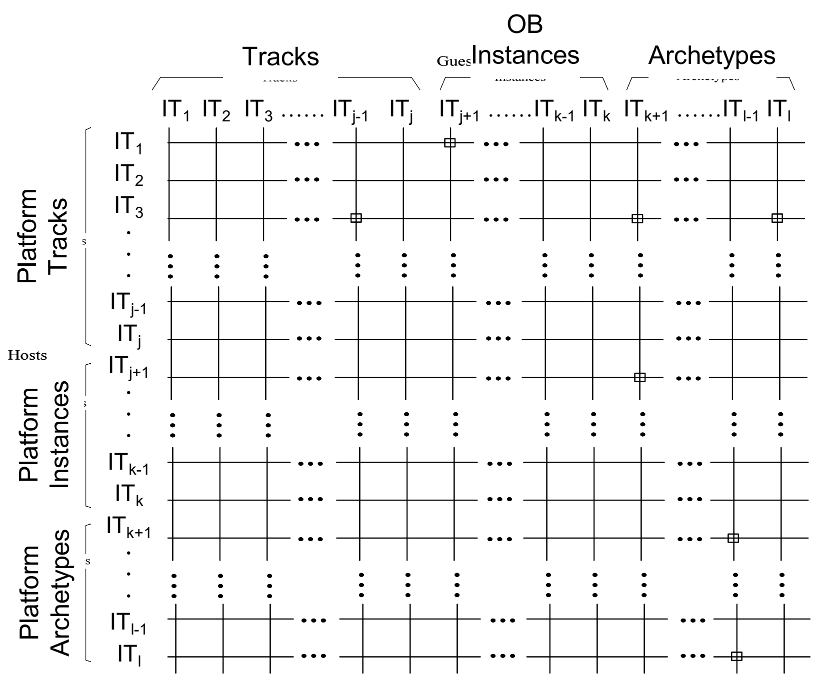
Figure 22. Overview of Classification Algorithm Data Structure

The track file levels in the legacy software algorithm are related via two-way ITF-ITF links, ITF-ETF links, and ETF-STF links. The first two are many-to-many; the last is one-to-many. The levels and the links are related via candidacy links, indicating and storing possible identification and correlation candidacies. Candidacy links are STF-STF, STF-ETF, STF-ITF, STF-ETF/ITF. Candidacy links store the probability values for recursion. Candidacy links and their half-rules are shown in . As can be seen, the links explicate or reveal the nature of OB, ESM, and ELINT fusion over all categories of target via the logical constraints [84] governing their formulation. For instance, the first diagram, showing ST-ST links shows how input track reports (running down from ST_1 to ST_m have candidacy links to other tracks and also to modes. The filled nodes at the crossings exemplify a candidacy link. In the next diagram, both candidacy links and physical links are shown, the former as filled nodes, the latter as small squares. In the legacy software, a complex set of link utilities provide a consistent means for adding, dropping, updating, and traversing physical and candidacy links.



- For example track IT_2 is a ship of same class as instance IT_{j+1} with ET_{s-1} radar discovered as ET_r and ET_s dormantly linked via inheritance at confirmation (auto or manual)
- All IT tracks have a linked ET since only sensors are ESM/ELINT
- An IT track can have 1-to-many ET tracks and/or archetype links
- An IT archetype can have 1-to-many ET archetypes links

Figure 23. IT – ET Linkages Convey Radar "Fit"



- For example, Airbase IT_{j+1} hosts A/C type IT_{k+1} which hosts weapon type IT_{l-1}
- For example, Ship track IT_3 hosts archetype A/C IT_{k+1}, archetype weapon IT_l, and track IT_{j-1}
- Tracks can host tracks, instances, and/or archetypes
- Archetypes can only host archetypes
- Upon confirmation (auto or manual), instances become tracks and tracks inherit instance data. Archetypes get OB count decremented.

Figure 24. IT – IT Linkages Convey Theater Information

3.5.1.5 Non-Parametric Statistical Scoring

Many ESM input reports are kinematically in the form of Line Of Bearing (LOB) or bearing-only reports. These are only awkwardly representable in Cartesian covariance matrices since there is no know range, necessary for the coordinate system conversion. Cartesian trackers of bearing-only inputs have known behavioral anomalies. Some algorithms have represented LOB's as high eccentricity ellipses. Other algorithms use linear LOB/AOP trackers but, unfortunately, the range estimate can "runaway" for single source LOB tracking as the range converges to zero. This is due to the difficulty in making a good initial range guess, the ambiguous interpretation of a bearing change in terms of the target motion model, and, in linear Kalmans, the linearization in the observation equation [88][54][55]

Modern computing speeds and memory capabilities allow radical approaches to these problems. In particular it may not be necessary to coerce bearing-only measurements into Cartesian form, but to maintain the probability density values individually over a grid. Non-parametric techniques can be used to address these problems. For example, kinematic scores could be calculated using non-parametric statistics, treating uncertainty regions as non-parametric Locational Probability Densities (LPD's). Non-parametric LPD's can be valuable in the littoral area where coastlines, mountains, waterways, etc. can influence the likely location of certain types of platforms [87][83]. Such representation can also be useful for multiple-LOB and AOB data correlation scoring and fusion and also for terrain tailoring wherein kinematic probability mass is redistributed to areas where the target type is more likely to traverse, as Figure 25 suggests. In a simple case, probability mass for a ship's AOP that overlapped land would be redistributed to the oversea region. Correlation scores are computed as approximate integrals over the overlapping PDF's. Kinematics are fused as normalized element-by-element products of the PDF grids. The number of discrete PDF elements maintained could vary depending on the real-time requirements, data update rates, computational resources available, and mission accuracy requirements.

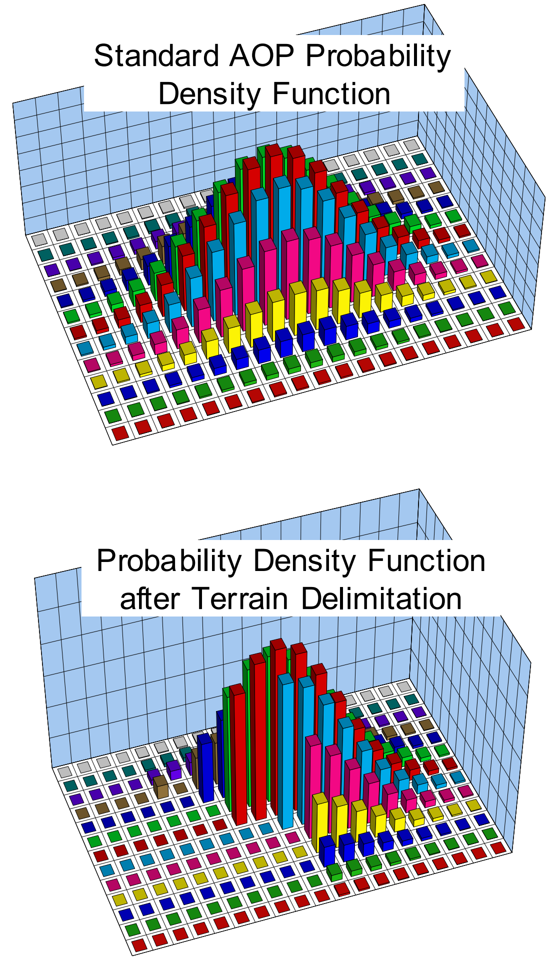


Figure 25. Non-Parametric Kinematic Uncertainty Representation & Terrain Delimitation

Another example is non-parametric ESM parameter scoring. This technique can be used to one-dimensionize scoring parameters, allowing for increased future abilities for parameter range non-parametric shaping for, say, channelized transmitters or uniform distributions. Other cases are multiple frequency radars and staggered PRI. The non-parametric parametric representation can also support higher fidelity representation of learning of or history-keeping for long-observed emitters.

3.5.1.6 Other Characteristics and Performance Clues

The algorithm also uses a variety of additional minor clues to further influence the probability vectors and reduce ambiguity. These are:

a. Effective Radiated Power (ERP) INTEL data is used along with estimated track range and known sensor detection sensitivities using estimates such as in [91]

b. NID Platform Operating Ranges
c. Platform Operating Range Defaults
d. NID salvo size and firing rate

3.5.1.7 Ontology Taxonomic Ambiguity Resolution Aids

For identification problems, own system limitations, natural factors, ambient environment, adversary techniques, and so forth can conspire to create problems requiring massive information processing and high-order human decision making. EW operators evaluate measurement data on their CRT's, signal modulations on audio, other sensor data on their PPI's, and intelligence data via system lookups, briefings, messages, intelligence documents (e.g., EPL and EW OPTASK), and open source documents. EW identification is an information intense activity. The algorithm's strengths are that it can consider and collate vast amounts of data in its reasoning that can aid an analyst/operator in resolving high-ambiguity tracks. The candidate identifications are ranked according to the ontology's taxonomies that provide an organized manner to show the most probable "branch" of the taxonomic hierarchy. That is, the most probable Category's Platforms are presented followed by the most probable Platform's Specific Types, etc. The operator can explore other branches of the hierarchy using standard windows tools to select a lesser probability item (e.g., a lesser probability Category). Upon such an event, the entire hierarchy would switch to the operator-selected branch. So even if there is a lot of ambiguity in the rankings and none of the "leaf" taxonomy candidates has a high probability, perhaps as the probability masses sum up to the higher taxonomy levels, an actionable candidate will emerge.

3.5.2 Algorithms

The algorithms follow in the flow shown below. The notational correspondence between this model and the formulae in the succeeding text is as follows:

STmeasured	RO	RECEIVER-OUTPUT
STsensor-track	RFT	TRANSMITTER
STa-priori-mode	RFTT	TRANSMITTER-TYPE
ETsensor-track	RFE	RF-EQUIPMENT
ETa-priori	RFET	RF-EQUIPMENT-TYPE
ITsensor-track	POI	OBJECT-ITEM for a FACILITY or MATERIEL-ITEM of Ship, Aircraft, Missile, Weapon
ITa-priori	POT	OBJECT-TYPE for a FACILITY or MATERIEL-ITEM of Ship, Aircraft, Missile, Weapon

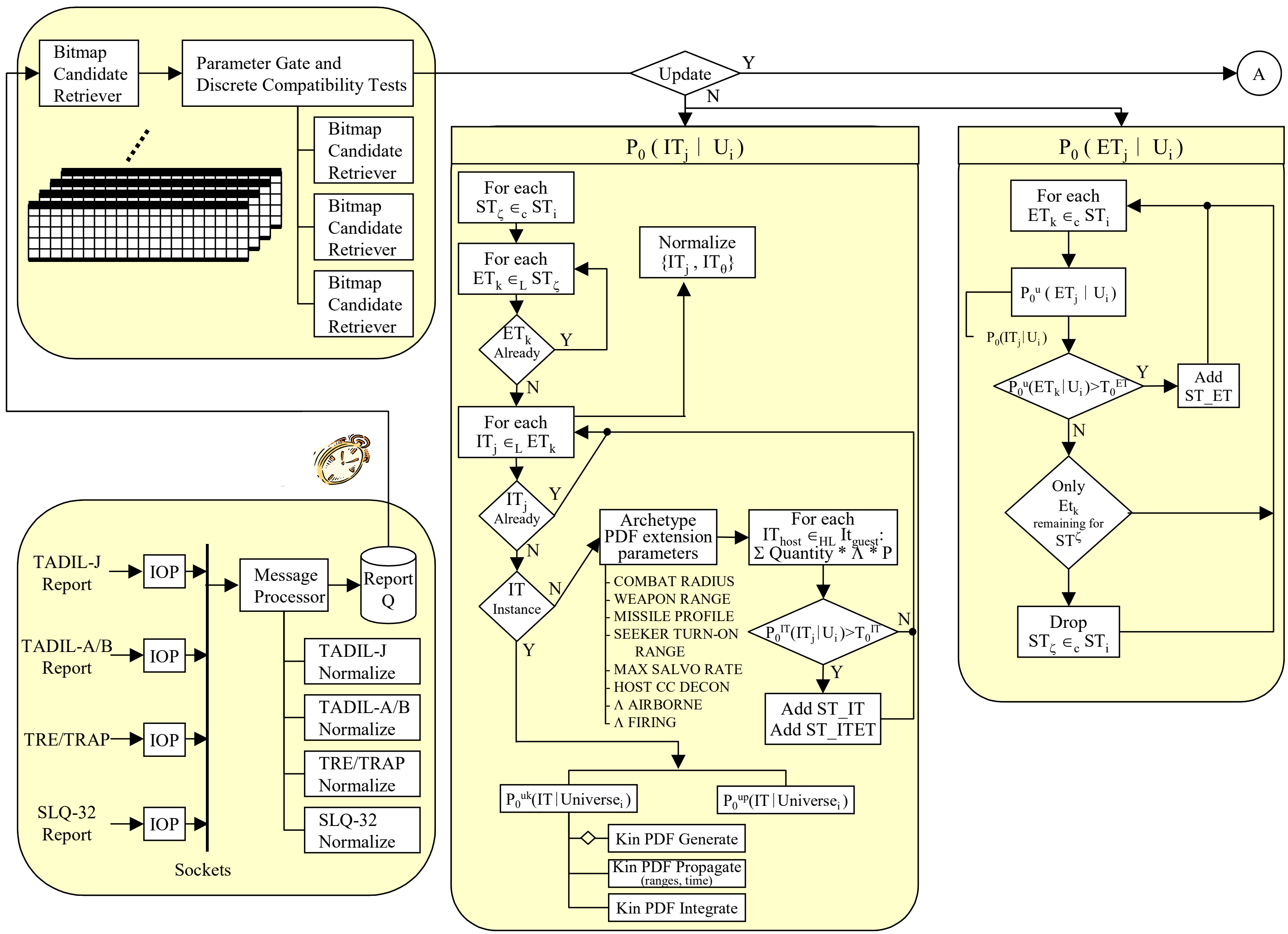
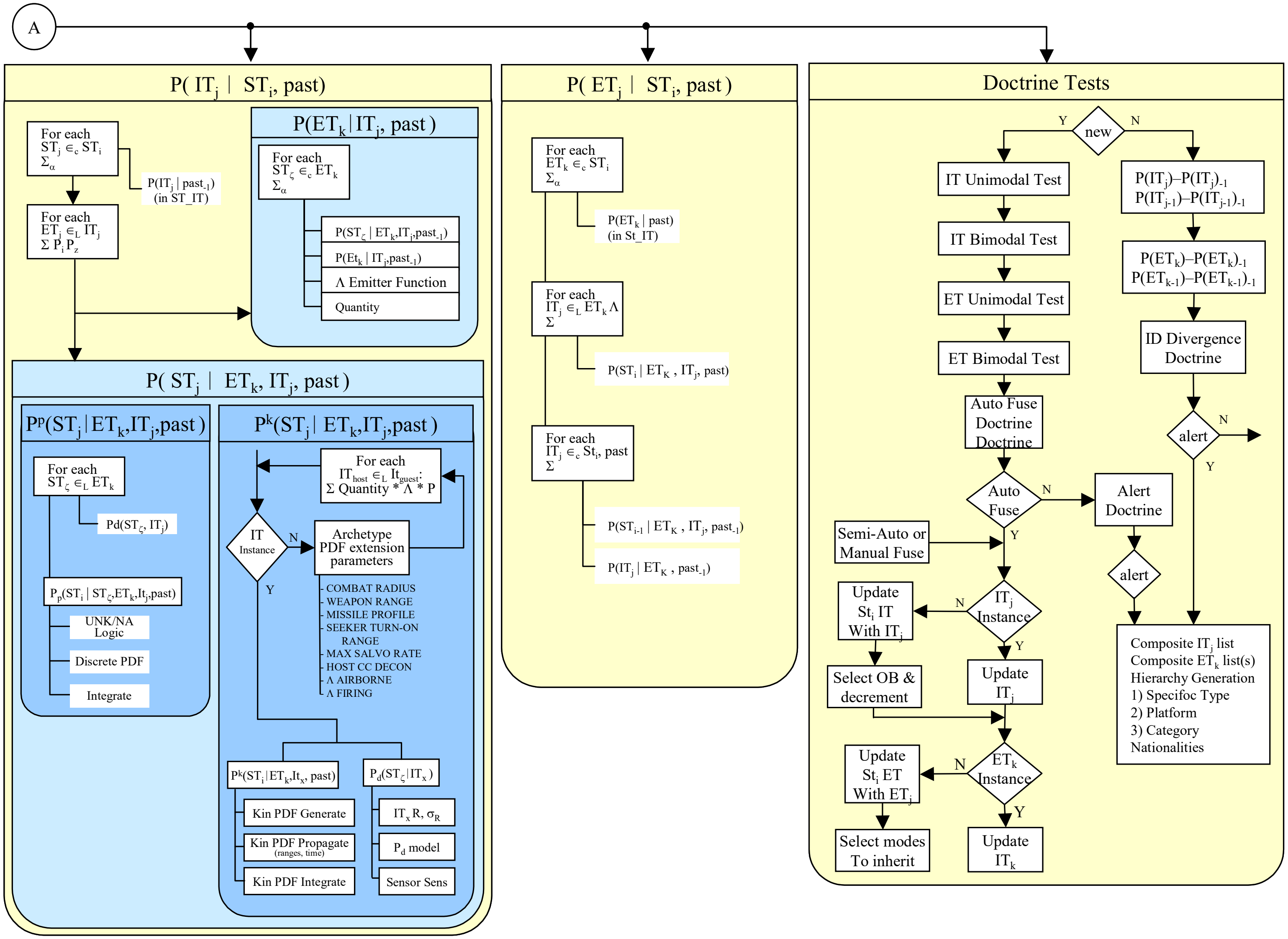


Figure 26. ESM SSI Algorithm Flow (page 1 of 2)



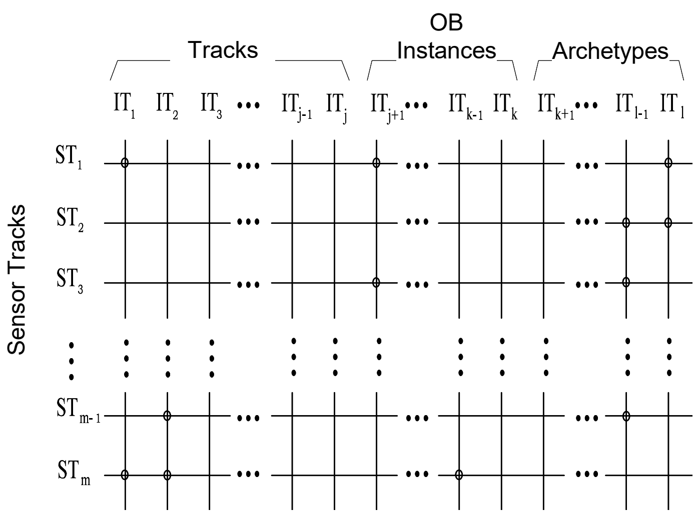
ESM SSI Algorithm Flow (page 2 of 2)

3.5.2.1 Parametric Candidate Selection and Gating

The primary gating criteria for all ESM and ELINT reports is waveform parameters. This "gating" is by virtue of the bitmap retrieval scheme which instantaneously retrieves RF, PRI, SCAN_CHR and MOD_TYPE candidates which are then logically and-ed to create the final candidates list. This is used as the first discriminant over kinematics because, for ESM and ELINT, waveform parametrics will normally be more discriminating, particularly true for ESM bearing-on (LOB) reports. However, a kinematic hash subsequent to the parametric hash could be added later to meet further real-time requirements.

The mode/ST candidate bitmap is then decoded into a scratchpad ST and ST-ST candidacy links which are RFT-RFT [[78]] and RFT-RFTT associations in the ontology. Scoring begins by chaining up the ET-ST (RFE-RFT) and IT-ET (RFT-OI) physical links to identify IT candidates linked to the ST candidates (OI-OT). Of course, multiple IT's per ST candidate are typical; even multiple ET's per ST are typical.

3.5.2.2 Probability of an IT_j Given ST_i and Past



- An ST track can like 0-to-many IT tracks, instances, and/or archetypes
- Holds P(ITj | past)-1

Figure 27. A Sensor Track (ST) has Candidates with Intermediate Track (IT) Instances and Archetypes

$$P(IT_j|ST_i,past)=\alpha\, P(ST_i|IT_j,past)P(IT_j|past)$$
$$\alpha \text{ over all } IT_\varsigma \in_c ST_i$$
Equation 1
$$\text{(The } IT_\varsigma \text{ constitute a partition over the } ST_i \text{ hypothesis space.)}$$

where:

IT_j	the OBJECT-TYPEs representing the order-of-battle FACILITY, AIRCRAFT, SHIP, and WEAPON.
ST_i	is the input sensor track that starts in RECEIVER-OUTPUT and then is given an OBJECT-ITEM corresponding to a TRANSMITTER
α	is the Bayes inversion “normalization” value taken over all IT_ζ that

	are candidate elements of the ST_i hypothesis space, that is, that have a possibility of having been the cause of ST_i. Note the partition assumption.

$$P(IT_j|past) = \begin{cases} \text{previous pass value (in ST - IT cand)} \\ \quad \text{if : } ST_i update \\ \\ \left[P_0^u(IT_j)\right]^{N\{IT_\varsigma \in ST_i\ ,IT_\theta(ST_i lat, ST_i lon)\}} \\ \text{as per Platform A - Priori Classification Probabilities algorithm} \\ \quad \text{if : } newST_i \end{cases} \qquad \textbf{Equation 2}$$

where the un-normalized a-priori P_0^u is normalized across the IT_ς that are candidate elements of ST_i and an OB uncertainty, IT_θ (unknown), varying over lat, lon [[86]]. IT_θ accounts, in a Dempster-Shafer manner, for the unknown in the a-priori data.

3.5.2.3 P_STi_ITj_p

$$P(ST_i|IT_j, past) = \sum_{\forall\ ET_k \in_L IT_j} P(ST_i|ET_k, IT_j, past)\, P(ET_k|\, IT_j, past) \qquad \textbf{Equation 3}$$

where:

$ET_k \in_L IT_j$ are all ET_k OBJECT-TYPEs that could be installed-on/in or assigned-to (ESTABLISHMENT or HOLDING) the IT_j OBJECT-TYPEs representing the order-of-battle FACILITY, AIRCRAFT, SHIP, and WEAPON. There is an assumption that the ET_k form a partition of IT_j in "Link" (ESTABLISHMENT, HOLDING) space.

ET_k is the OBJECT-TYPEs representing the TRANSMITTER-TYPE

IT_j is the OBJECT-TYPEs representing the order-of-battle FACILITY, AIRCRAFT, SHIP, and WEAPON.

ST_i is the input sensor track that starts in RECEIVER-OUTPUT and
then is given an OBJECT-ITEM corresponding to a TRANSMITTER

3.5.2.4 Probability of an Emitter Given a Carrying Platform and Past (P_ETk_ITj_p and includes Pu_0_ETk_ITj and Pu_0_ITj_Etk)

$$P(ET_k|IT_j, past) = \begin{cases} P(ST_{i_{last}}|ET_k, IT_j, past_{-1})\, P(ET_k|IT_j, past_{-1}) \\ \quad \text{if: } ST_i\, update \text{ (both terms in ST - ET-IT cands)} \\ \\ \dfrac{QUANTITY[ET_k, |IT_j] * \Lambda_{EMTR\_FUNC}(ET_k)}{\displaystyle\sum_{ET_\zeta \varepsilon\ IT_j \wedge ET_\zeta \text{ in } ST_i \text{ universe}} QUANTITY[ET_\zeta, |IT_j] * \Lambda_{EMTR\_FUNC}(ET_\zeta)} \\ \quad \text{if: } new\ ST_i \end{cases}$$

Equation 4

$ST_{i_{last}}$ refers to the values of ST_i prior to this update

QUANTITY is the number of ET_k installed-in/on or assigned-to IT_j

Λ_{EMTR_FUNC} is the emitter likelihood function, the likelihood that ET_k would be operated and transmitting at a given moment in time

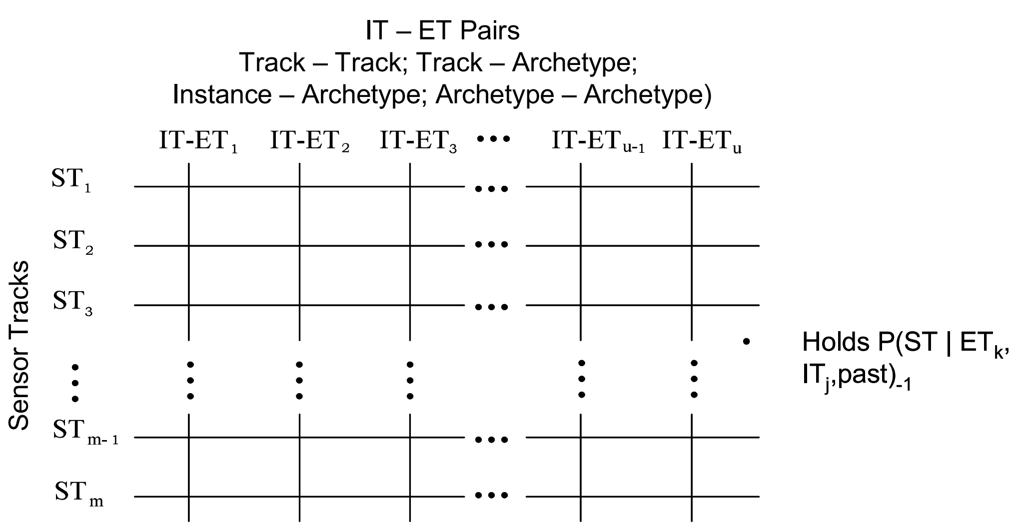


Figure 28. A Sensor Track (ST) has Candidates with Emitter Platform Distance and Archetypal Configurations

3.5.2.5 P_STi_ETk_ITj_p

$$P(ST_i|ET_k, IT_j, past) = \left[w_K * P^K(ST_i|ET_k, IT_j, past) * w_P * P^P(ST_i|ET_k, IT_j, past)\right]^{1/2}$$ **Equation 5**

(NOTE: weighted geometric mean.)

3.5.2.6 Pp_STi_STz_p

$$P_p(ST_i|ET_k, IT_j, past) = \sum_{\varsigma} P_p(ST_i|ST_\zeta, ET_k, IT_j, past)\ P_p(ST_\varsigma|ET_k, IT_j, past)$$ **Equation 6**

(NOTE: the ST_ζ constitute a partition of ETk once they are made mutually exclusive.)

$$P_p(ST_\varsigma|ET_k, IT_j, past) = P_0(ST_\varsigma|ET_k) P_d(ST_\varsigma|IT_j)$$ **Equation 7**

(NOTE: probability of detection of the candidate allows use of NERF ERP.)

3.5.2.7 Pu_0_Etk

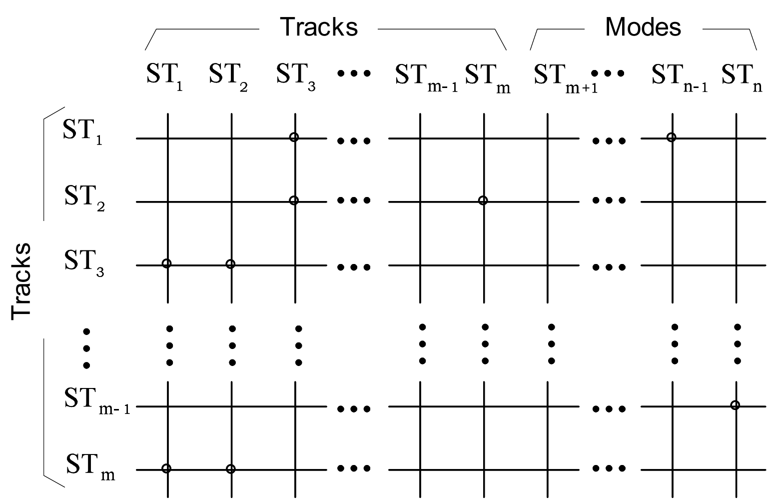
This function computes an un-normalized sub-universe a-priori of an emitter given an st_i sub-universe -- defined to be et_k's having at least a mode overlap with st_i and, if et_k is an instance, the st_i and et_k kinemaps overlap.

$$P0(ST_\zeta | ET_k) = \frac{1}{\#\ \mathrm{mod}\, es\ and\ ST's \in_L ET_k \wedge \in U_i}$$ **Equation 8**

(NOTE: only modes and ST's in ST_i universe.)

3.5.2.8 Parameter Probability Scoring (Pp_STi_STz_p)

The parameter score is relatively easy since recursion through the ET's and IT's is not required, just scoring to the candidates the scratchpad ST holds. If the candidate ST is a mode, the min/max intervals are converted to 2σ discrete Gaussians (7 intervals currently). For ST tracks, it is planned for the history of hits will be maintained using some form of fading memory, state-transition detection, etc. The two sets of discrete PDF's are integrated in a similar manner to the kinematic integration. Missing parameters are handled using the geometric mean of the available parameter scores, the equivalent of χ^2 additive doping factors or multiple thresholds. Discretes are checked for compatibility using compatibility tables. Of course, presence of a parameter in one track/mode while it is Not Applicable in the other is cause for disqualification. Like the kinematic scores, the ST-ST scores are combined with a-priori's and, using conditionalization and inversion rules, the final set of ST-ST, ST-ET, ST-IT, and ST-ITET probabilities is formed. These are maintained for display, decision logic, and next pass recursion.



- Usually, if B is a candidate of A (i.e., A "likes" B), B likes A but not always (ST_m likes ST_1 but ST_1 doesn't like ST_m)
- It is possible for a track to not like any other tracks implying new track (ST_{m-1})
- It is possible for a track to not like any modes implying a new mode (e.g., WRM)
- Probabilities based on mode/track parameter probabilities and kinematic probabilities with linked ET, linked IT, and IT-IT basing and loadout links, and C&P, geopolitical, and other parameters.
- Holds $P(ST \mid ST, past\ )^{-1}$

Figure 29. A Sensor Track (ST) has Candidates with other Tracks and NERF/EWIR Modes

$$P_p(ST_i|ST_\varsigma, past) = \left(\prod_n w_{p_n} * \sum \left[f_{ST_i} f_{ST_\varsigma} \Delta^2 \right]^{1/2} \right)^{1/n} \qquad \textbf{Equation 9}$$

(NOTE: weighted geometric mean.)

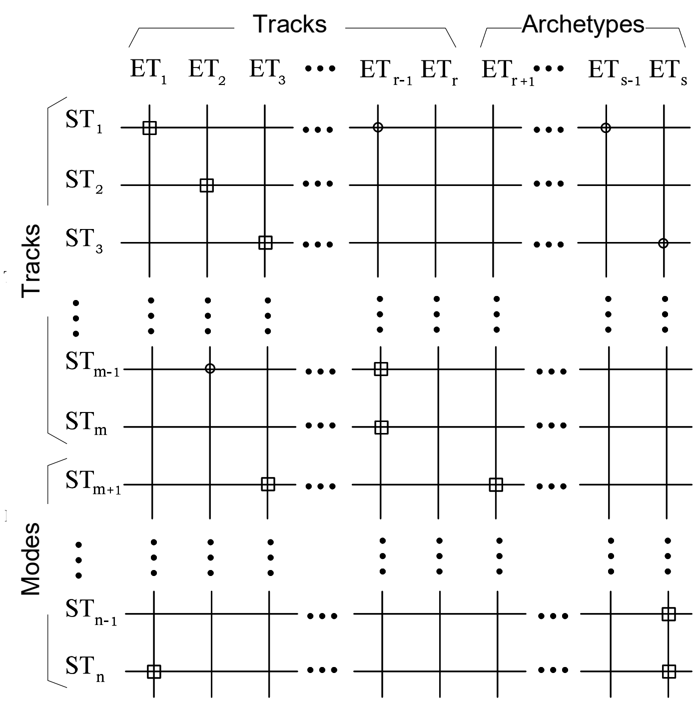
3.5.2.9 Probability ET_k Given ST_i and Past

$$P(ET_k|ST_i, past) = \alpha P(ST_i|ET_k, past) P(ET_k|past) \qquad \textbf{Equation 10}$$

$$P(ET_k|past) = \begin{cases} \text{previous pass value (in ST - ET cand)} \\ \left[P_0^{u1}(ET_k)\right]^{N\ :ET_k in ST_i universe} \end{cases}$$

Equation 11

3.5.2.10 Probability of a Mode Given an Emiter and Past (P_STi_ETk_p)



- An ST always has a least one ET link and vice-versa
  - Each mode links to a single archetype and 0-to-many ET tracks
  - Each ST track links to one and only one ET track
- Only the concurrent and transitional modes are linked to ET tracks via inheritance
  - Allows mode tracking
- Multiple ST tracks linked to a single ET track are the result of a correlation decision
- An ST track can have candidacies with 0-to-many ET tracks and/or archetypes
- Candidacy scores are a function of ST track/mode score, IT track/instance score, ST-ET and ET-IT links containing mode-emitter, basing/loadout defaults, and C&P, political and other parameters
- Holds P(ST | ET_k, past $)_{-1}$

Figure 30. A Sensor Track (ST) has Candiates and Links with Emitter Tracks and Emitter Archetypes

$$P(ST_i|ET_k,past) = \sum P(ST_i|ET_k,IT_j,past)P(IT_j|ET_k,past)$$

Equation 12

First term is as defined in the Probability IT Given ST and Past algorithm.

3.5.2.11 Probabilty of a Causing Platform Given an Emitter and Past (P_ITj_ETk_p)

$$P\left(IT_j \middle| ET_k, past\right) == \begin{cases} P(ST_{i_{last}} | ET_k, IT_j, past_{-1}) P(IT_j | ET_k, past_{-1}) \\ \quad \text{if : } ST_i update \text{ (both terms in ST - ET-IT cands)} \\ \\ \left[P_0^{u2}\right]^{N \,:\, ET_k \varepsilon IT_j \wedge ET_k in ST_i universe} \\ \quad \text{if : } newST_i \end{cases} \qquad \textbf{Equation 13}$$

3.5.2.12 Kinematic Score Algorithm (Pu_0_ITj and Pk_STi_ETk_ITj_p)

This function computes an un-normalized sub-universe a-priori of an platform given an st_i sub-universe -- defined to be it_j's having at least a mode overlap with st_i and kinematic overlap either as an instance or in OB if an archetype.

For IT-ST kinematic scoring, the input ST kinematic data (LOB or AOP) is converted to a discrete PDF. LOB conversion considers the sensing origin, max detection range of that sensor against the ST candidate's Maximum Effective Radiated Power (from NERF), horizon versus altitude of the linked IT candidate, and seeker-turn-on-range for active weapons IT candidates. Time since initial detection by that sensor (if the ST report is a coherent track report) is also used to "count down" the approximated initial ranges.

IT-ST scoring recurses if the IT is an archetype, thereby necessitating estimating its probability density about its host ship, aircraft, or airbase. For ships that have not been detected by a sensor but could be within surveillance range according to NOB, an expected max range from homeport is looked-up based on IT TADIL-J Specific Type, or if the Specific Type value is NS, the next level up, Platform [[36]]. For weapons, NID max-salvo-rate and max-range values are used along with other detected launches from the firing platform and the firing platforms weapons loadout (in NID and NERF). If the weapon candidate is launched from an aircraft, the PDF must also be iterated against the aircraft's max-combat-radius (from NID) from any airbase linked to the aircraft (via IT-IT links) that are within the kinematic gate of the reported ST, extended to account for aircraft and weapons radii. This iteration of ranges is illustrated in Figure 21. If the IT candidate is a track, then its PDF must be propagated (forward or backward). A real-time algorithm for this propagation is an area of continuing research. If history exists, backward would be the choice. Exceptions and large uncertainties in many of the input values are tolerable to the algorithms because they do not have a sensitive response to small variations. Despite uncertain or incomplete data, inclusion of these factors provides an advantage over current approaches which, in effect, assume uniform distributions and total pre-engagement ignorance. Once the PDF's are generated and aligned, they are scored by taking the multiple sums of the products of the overlap discrete PDF elements:

Once the PDF's are generated and aligned, they are scored by taking the multiple sums of the products of the overlap discrete PDF elements:

$$\sum_i \sum_j PDF_1(x_i, y_j) PDF_2(x_i, y_j) (\Delta x_i \Delta y_j)^2 \qquad \textbf{Equation 14}$$

This involves determining a set of summing intervals for each degree of freedom that is the superset of the ST report and IT candidate discrete PDF intervals. The result requires no normalization -- it is the probability that two such PDF's could be manifested from the same target.

As the ET-ST, IT-ET, and IT-IT recursions pass back values, they are combined according to conditionalization (total probability) and Bayesian inversion rules. In the process, previous pass values will be invoked. For the first score of an ST against an ST, ET, IT-ET, or IT, a-priori values must be used. Since a-priori values have meaning only within a specified universe, we take the universe as small as possible (3σ), resulting in dynamic "on-the-fly" a-priori's. This is in contrast to current identification and fusion systems that compute a-priori's pre-runtime. Our research led us to this approach when we looked into the universe size to be used for a-priori's. Since we anticipate a 2000 nm radius surveillance volume, we expect the a-priori's to vary significantly over the surveillance volume. It seemed best to choose as the universe for a report the smallest universe that we can be reasonably confident will always contain the track..

3.5.2.13 Probability of Detection of an Input Measurement Given Platform Causing (Pd_STz_ITj)

3.5.2.14 Decision Logic

Decision-making logic can be by multiple uni- and bi-modal thresholds for Semi-auto and Full-auto modes. However, since multiple ST's can be linked to an ET and multiple ET's can be linked to an IT, it is necessary to form a composite of the multi-source tracks to get a decipherable set of candidates for each IT and ET track. (Even if the ST had no identification candidates, it would result in a new ET and new IT linked to it.) This is done using ST composite scratchpad ST's in which to generate the composites for each ET that may be linked to an IT (currently parameter-wise max'd at 10). Composites are not maintained, they are generated only for decision making, either automatic or semi-auto. Other ambiguity resolution and decision logic schemes can be used as well.

If the mode is semi-auto or either of the auto-thresholds fail, the operator would be notified of an identification ambiguity requiring more intuitive or subjective judgment. The operational principle is to give the operator some level of control over the thresholds resulting in notification so he can level his workload or, alternatively, his confidence in the automatic algorithm. The software could also monitor for overload indications such as alert queue backup.

Upon selection of an alert, the operator would be presented with summary data on the IT with buttons for exploring its linked IT's and ET's, each resulting in popups. From the ET popups, ST's can be called up. Candidates could be shown in scrolling lists in order of probability. The operator would probably prefer to see the most probable Category, Platform, Specific Type, DIA platform, and Emitter types initially. As the operator scrolls the lists of candidates he/she wishes to explore and takes the select action, the hypothesis hierarchy tree could switch to the selected branch. [[48]]. The operator, when analyzing a track from the emitter point of view, probably wants to know all the Emitter type candidates, not just the ones for the best platform.

3.5.2.15 Track Merging

Upon an identification decision, whether automatic or semi-automatic, a number of adjustments take place. The hooked IT is merged into the selected candidate along with its linked ET's and their linked ST's. Kinematics are fused at the IT level. Parameter min/max's are updated in the ET for display purposes only -- the linked ST's fully convey the parameter data for identification and fusion purposes. If the candidate ET or IT is an archetype, its OB count is decremented.

The most likely airbase or weapon launcher is used for aircraft and weapons. The archetype characteristics are then inherited by the track. For ET's, all the concurrent or transitional modes are inherited according concurrent and transitional groups of modes as indicated in the ET-ST links. The inherited modes will be indicated as dormant or undetected but will be available to match future incoming reports. The mechanism for the inheritance is to add links to the single mode records. Similarly, for IT's, IT-ET links are added for the undetected emitters so that if they should be detected, the IT track will become an identification/correlation candidate.

3.5.3 Phase I Experimentation

In the Phase I we had to scope the experimentation to the essential purpose of the Phase I, that is, to determine if and how legacy (existing) fusion algorithms can be hosted to run in the new fusion architecture based on the fusion ontology running in a managed object environment (the TimesTen DBMS). The experiment design overview is shown in Figure 31. We had to simplify to this level for this stage of experimentation in keeping with prudent experiment design so many of the mathematical and inferential features of the existing algorithm were not tested. Rather, we focused on the data access, the invocation mechanism, and the belief updating. The key challenge we found in the experiment was coping with the data access. In prior experiments we had used wrappers to marshal up the data needed by an algorithm, get it translated into the data structure format the algorithm was accustomed to, and then de-marshal / parse the results back into the ontology. We realized early on that this would not be practical for this type of algorithm because of the massive amount of marshalling and parsing that would be required. The solution was an object class layer synchronized with the TimesTen DBMS. That, along with each of the components of the experiment are described in the following subparagraphs along with the results.

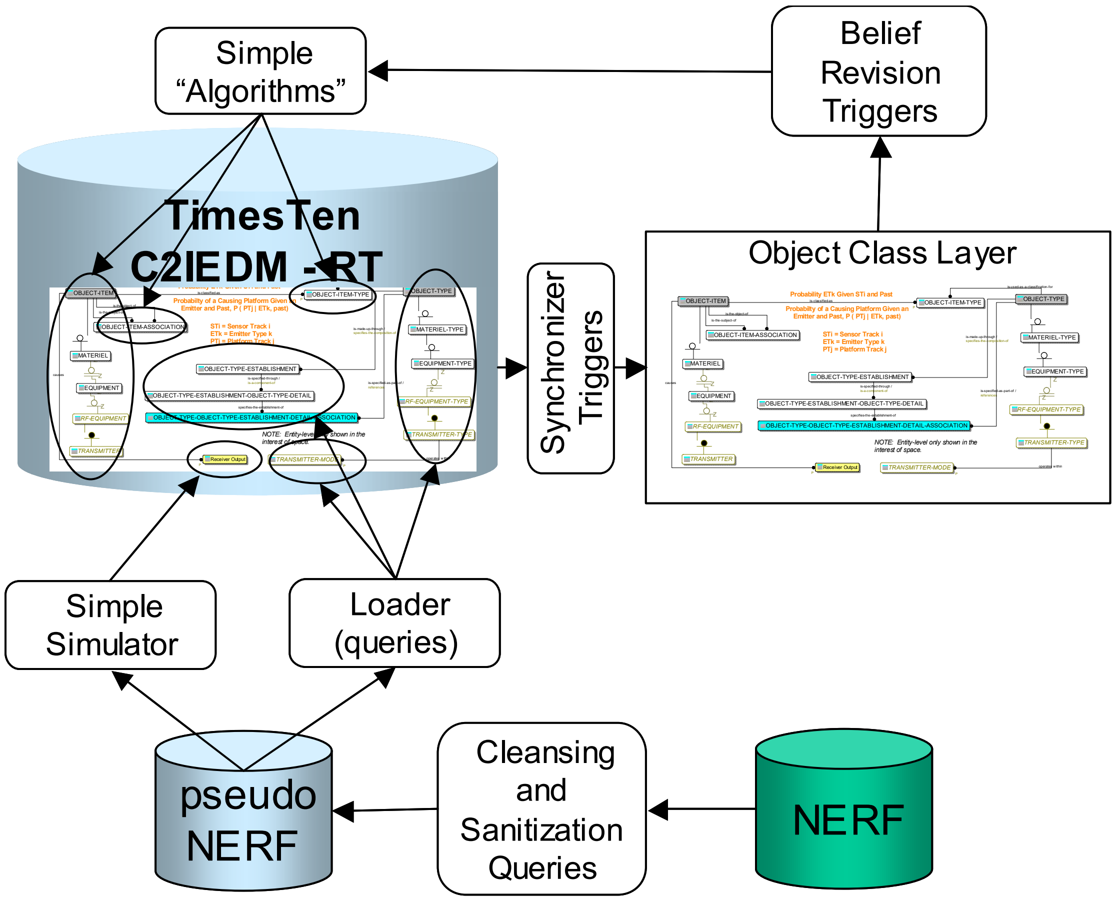


Figure 31. Phase I Experiment

3.5.3.1 NERF Setup and Loading

The prior population for this experiment will be dummy data for ease of experimentation and classification purposes. For this experiment, 69,757 object types and 35966 object items will be created with overlapping equipment and ELINT parameters. This was based on the classified NERF which was redesigned, sanitized, and formatted as shown in the schema below:

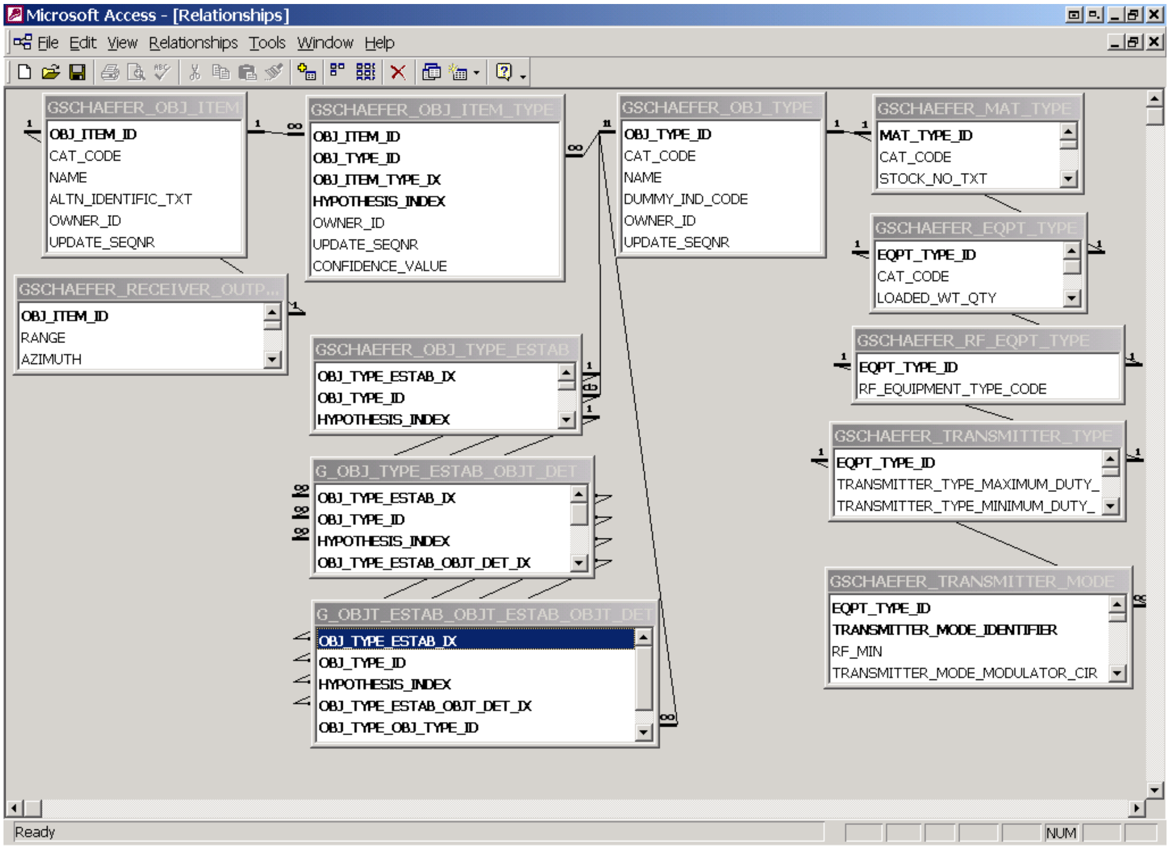


Figure 32. NERF Schema Used for Phase I Experiment

The sanitizer wiped out all traces of the original data – names such as DIA equpment codes, ship names, ELNOTs, etc. were replaced with numbers, actual values such as RF Min were replaced with random values. The idea was to create a database that had some of the ambiguity and size characteristics of NERF but not any sensitive data.

The loader simply went from the NERF structure to the C2IEDM-RT structure to create an initial a-priori database in the TimesTen / C2IEDM-RT fragment used for this experiment shown in Figure 33.

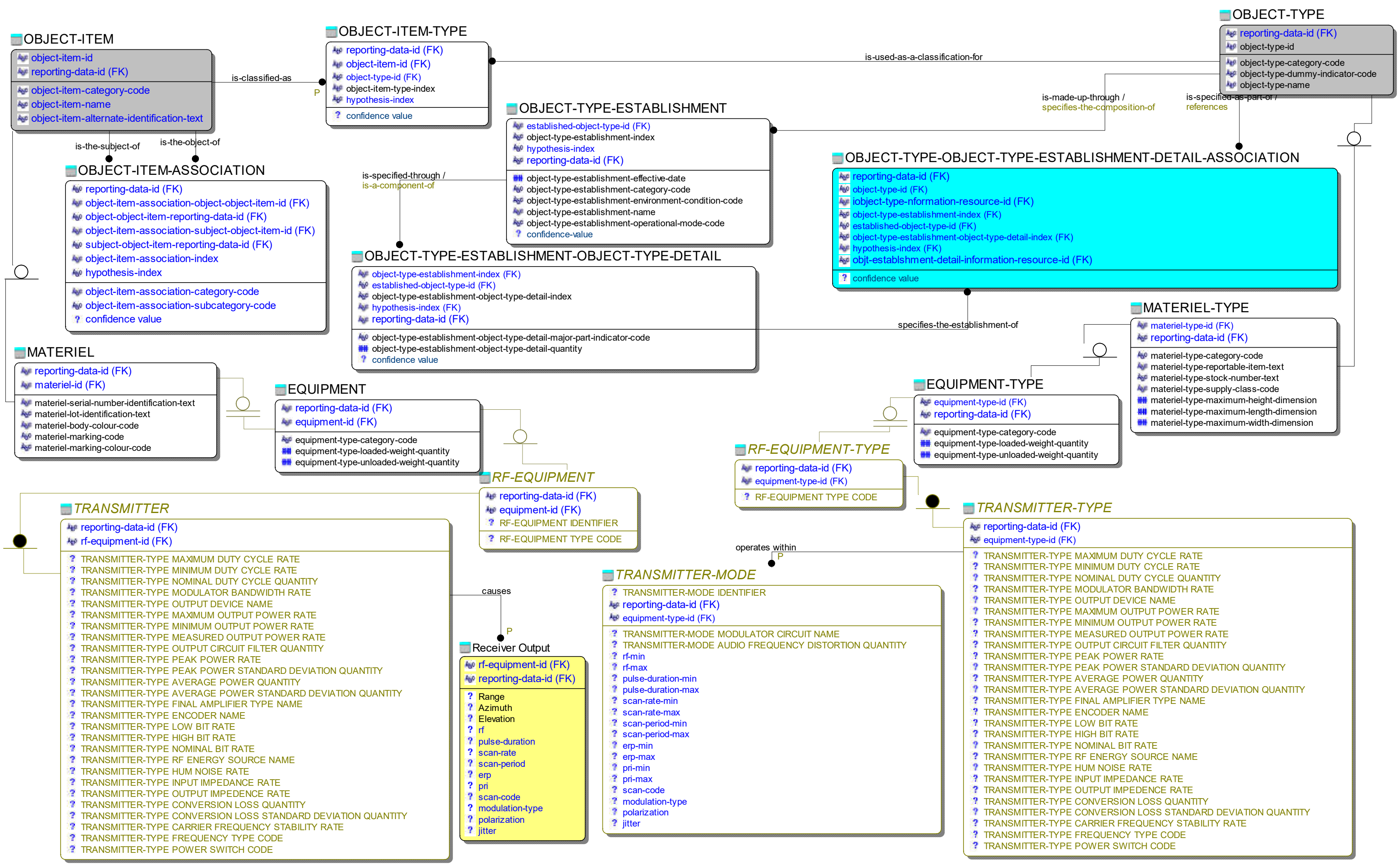


Figure 33. Attribute Level Model for ESM / ELINT Classification Experiment in Phase I

3.5.3.2 Simple Simulator

The simulator simulates an ESM / ELINT track with basic classical parameters:

RF	The carrier frequency of the signal
PRF (or PRI)	Pulse Repetition Frequency or its arithmetic inverse, the Interval. The interval is the time duration between pulses for a non-CW radar
PD	Pulse Duration
Scan Period	For scanning radars, the primary periodicity
Scan Rate	For scanning radars, technically this should be one of the other periodicities. For example, for the US Navy SPS-48E, this would be the phase generated elevation scan while the primary would be the azimuth scan. However, scan rate is often put in reference databases as merely the arithmetic inverse of the Scan Period.

The source for the simulated tracks was the pseudo-NERF itself. The simulator runs through a set of randomized object items, one by one.

3.5.3.3 Object Class Layer and Synchronization

The repetitive nature of the experiment, and the need for fast processing, made it evident that instead of having the data structures access the DBMS each time, a local copy of the data stored within the object classes would be necessary. In this test, information from the database is used to instantiate object classes, creating a data access layer. The algorithm will interact with these data structures instead of the DBMS directly. Reading and writing information is treated differently. Reads involve only a local copy of the data. Writes are to the local copy and the database. Had the DBMS been running with other systems interacting with the data, it would have been necessary to use event triggers for synching local copies and changes other systems had made. The data access layer consists of three main structures: obj_item, obj_type and obj_item_type. Obj_item closely resembles the RECEIVER_OUPUT table since this is where the data is that it is linking to. For the same reason, obj_type's attributes are like those of TRANSMITTER_MODE and obj_item_type, similar to the table OBJECT_ITEM_TYPE. Each attribute in the structures is based on a sub class containing the code for SQL manipulations.

```
class obj_item {
  public:
    obj_item();
    obj_item(int,double,double,double,double,double);
    ro_pri pri;
    ro_rf rf;
    ro_sp sp;
    ro_sr sr;
    ro_pd pd;
    int id;
};
```

```
class obj_type {
  public:
    obj_type(int,double,double,double,double,double,double,double,double,double,double);
    int id;
    tm_pri_min pri_min;
    tm_pri_max pri_max;
    tm_rf_min rf_min;
    tm_rf_max rf_max;
    tm_pd_min pd_min;
    tm_pd_max pd_max;
    tm_sr_min sr_min;
    tm_sr_max sr_max;
    tm_sp_min sp_min;
    tm_sp_max sp_max;
    vector<int> platforms;
};
class obj_item_type {
  public:
    obj_item_type(int,int,double);
    int oi_id;
    int ot_id;
    double cv;
};
```

3.5.3.4 Belief Revision Triggers and Simple Algorithms

From this information, an estimate of the type of object (emitter, platform, etc.) is desired. The fusion inference seeks to find the most likely type of object to have caused this signal measurement to occur. The causality chain begins with the object that causes its radar to operate which causes the signal which causes the ESM / ELINT receiver to excite which causes the measurement to be sent to the fusion processor. This simplifies and allows for greater flexibility of the coding needed to embed a database for use in the experiment. The algorithm is as follows. Each of the classic parameters of the simulation data is checked to see if it is within the ranges of the transmitter modes of our object types. Also, special cases are considered, such as parameter data that is not applicable or unknown. From this and the relationships between transmitter modes, the emitters associated with them, and the known platforms of these emitters, we can derive lists of platforms and confidence values of the estimations.

3.5.3.5 Running the Experiment and Results

Initially, all known parameter range information about object types are loaded into the object_type data structures. Once started, simulation data is presented once per second to the algorithm and the results are recorded. The simulated reports are inserted into RECEIVER_OUTPUT. A new instantiation for the data access layer is created to tie this back to the database. The algorithm runs using the information in the structures and beliefs are compiled. Once again, new sets of data objects are instantiated, linking the results to the database table OBJ_ITEM_TYPE. Because there are 35966 reports total, each taking about one second, the scenario takes 10 hours to play out. A sample run of just 2 minutes creates over 700,000 results.

Table 6. Sample Values in Object-Item-Type

OBJ_ITEM_ID	OBJ_TYPE_ID	OBJ_ITEM_TYPE_IX	OWNER_ID	UPDATE_SEQNR	CONFIDENCE_VALUE	HYPOTHESIS_INDEX
1	11851	1	1	1	1.54E-04	1
1	11855	1	1	1	1.54E-04	1
1	11858	1	1	1	1.54E-04	1
1	11862	1	1	1	1.54E-04	1
1	11863	1	1	1	1.54E-04	1
1	11865	1	1	1	1.54E-04	1
1	11866	1	1	1	1.54E-04	1
1	11867	1	1	1	1.54E-04	1
1	11869	1	1	1	1.54E-04	1
1	11870	1	1	1	1.54E-04	1
1	11872	1	1	1	1.54E-04	1
1	11873	1	1	1	1.54E-04	1
1	11876	1	1	1	1.54E-04	1
1	11879	1	1	1	1.54E-04	1
1	11880	1	1	1	1.54E-04	1
1	11881	1	1	1	1.54E-04	1

3.5.3.6 Real-Time Considerations

Even though the purpose of this experiment was to test data access and invocation mechanisms, it is worth noting again some realtime considerations. We have known since the TBS research that using standard non-real-time RDBMS SQL queries is infeasible even if the database is indexed by each parameter min and max since the matches still have to be "AND"ed. We reported in TBS that an associative memory is needed to operate “in front of” the DBMS. This Phase I experiment confirmed that. Without a local copy for data reads, the SQL could not keep up with “real time”. The mostly in-memory DBMS, with speeds in the microseconds, could not compete with the nanosecond read times of RAM. It took generally 10 seconds to complete one second’s worth of input. Even though this is on a common office computer (2.4 GHz, 2 Gbyte RAM) running a non-realtime operating system, the need for the associative memory is strongly indicated.

Real-time techniques avoid searching by using pre-encoded match maps. In earlier experiments, we implemented a method illustrated in Figure 34. This method uses an index by measured parameter into a mode "bitmap" that can then be "AND"ed with the other measured parameter lookups thereby resulting in a bitmap, all of whose entries correspond to modes/sensor tracks whose parameter ranges include the input report.

The result is no searching or sorting -- all such work is done ahead of time. This real-time technique is appropriate not only for shipboard command and control and ASMD, but also for any environment in which there are quick react or high data flux requirements such as C3I systems and aircraft RWR's.

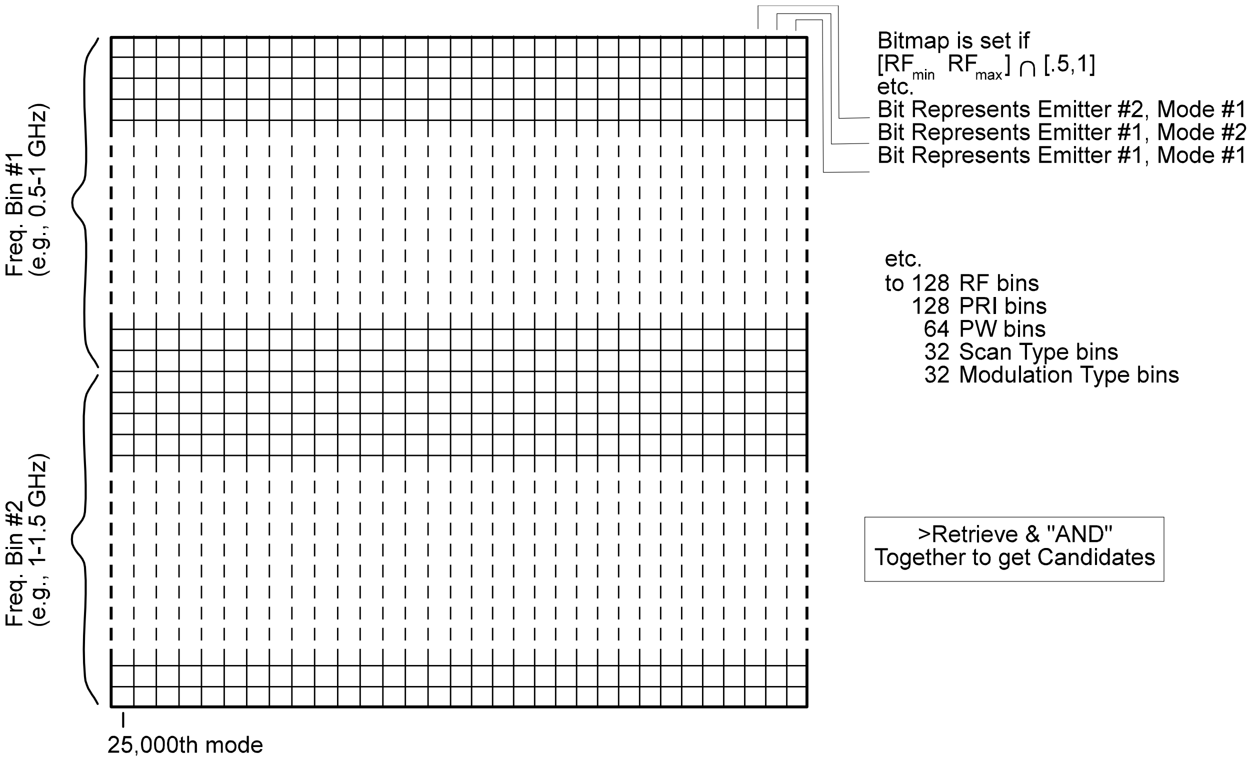


Figure 34. Real-Time EW Library Mode Candidates Retrieval Method

3.6 Axiomatic Theory of Ontology-Based Fusion.

Consider the problem of fusing data, and information from multiple sensors and sources, into a unified assessment of ground truth. Currently armies of human analysts, who collectively bring a wide range of knowledge to the problem, accomplish this process. It is unlikely that any single analyst would posses anything approaching the range of required knowledge, as the domains and perspectives vary too greatly. The expertise brought to bear draws on the physical, social, political, and military sciences. The diversity of knowledge required also makes it difficult to automate the fusion process. In many cases, the specific knowledge brought to bear by analysts is available from one or more reference or intelligence databases. The problem with making this type of stored knowledge accessible to automated systems is threefold. First, the stored knowledge sources must be formatted for machine processing and continually maintained in that format as they evolve. Second, one must provide declarative representation (ontologies) of relevant tactical situations so that incoming and stored information can be combined into a single logical representation. Third, one must provide inference algorithms that support machine reasoning about the ontologies.

The problem of combining multisource fusion with prior knowledge has been solved repeatedly by a succession of animals over evolutionary time periods. Successful animals continually combine sensory data with stored knowledge about situational contexts to construct a model of the current tactical and strategic situation. Penguins are not looking for the same predators when walking as they are while swimming. In both contexts they are combining auditory, visual, tactile, and olfactory information to construct a world model that supports both strategic (Where might I find food?) and tactical (How might I escape predation?) considerations. We are developing formal methods to allow computers to do the same sort of thing. This has been accomplished in a variety of domains, but has not yet been done for data fusion in military situations.

3.6.1 Formal method.

Fusion analysts often deal with situations where different sources of information (e.g., track files, intelligence databases, imagery) refer to the same objects, but reflect different attributes, time periods, or views of the objects. In the old story, the blind men constructed wildly varying models of the elephant, because they each constructed a model at a level of aggregation below the actual level of the elephant entity. In reality, it is often simple and natural for a human fusion expert to see corresponding elements from Intel and track databases as imperfect reflections of a single entity – but viewed from two perspectives. It is not straightforward for traditional fusion software to make this sort of abstraction-derived inference. The lack of an abstraction layer causes, (1) data integrity degradation and (2) a convoluted fusion architecture. We have developed an abstraction hierarchy that represents fusion entities, activities, and intents at the most abstract level, and supports hypotheses about specific things, actions, and goals based on observations and on inferences from those observations.

Reasoning is the process of using facts and inference rules to produce hypotheses and conclusions. Automated (knowledge-based) reasoning systems require at least two components: knowledge representation and inference. Knowledge representation is domain-dependent and must be acquired and formalized for each new domain of interest (knowledge engineering). Inference mechanisms are domain independent but they rely on specific formats of knowledge representation. Our inference mechanisms rely on first order and modal logics, which provide a representation formalism and inference mechanism (resolution) for many knowledge-based systems (e. g. expert systems, question- answering systems).

3.6.2 Theoretical approach.

Our theoretical argument is essentially two-fold. First, we argue that successful fusion cannot be accomplished by focusing within fusion levels, nor by building strictly upward from level 1 results to level 2 results and so on. Rather, good human analysts work across fusion levels – reasoning, sometimes hypothetically, back and forth among fusion levels. Second, we argue that successful automated fusion cannot be accomplished solely with traditional associative and statistical methods. A meaningful, conceptual understanding of hypothetical tactical situations is required to logically put together a picture of an actual tactical situation based on incomplete evidence.

For automation to occur, this “meaningful, conceptual understanding” cannot reside solely in the heads of human analysts but must be formally represented in software. Declarative techniques for representing prior knowledge and meaningful situation models must be applied, along with procedural techniques for reasoning about situations. These techniques are available today in a mature state after 40 years of development by the Artificial Intelligence community, but they are only recently garnering interest from the fusion community. “Ontology” is the contemporary term for a semantic network that is robust and complete within a proscribed domain of interest.

Automated inference is based on techniques pioneered as production systems, expert systems, theorem provers, and problem solvers. These techniques are at least functionally isomorphic to the way human beings understand, reason about, and solve problems in real world situations. Our current best avenue for getting computers to reason about tactical situations (Levels 2 and 3) is by applying these techniques on top of the statistical and associative methods we are already using for Levels 0 and 1 fusion. There is good evidence that this layered approach mirrors closely how human cognitive systems interact with sensory and perceptual systems.

3.6.3 Reasoning over abstraction hierarchies

Our intention to create agents that intelligently organize information into abstraction hierarchies is fundamental to the knowledge-based reasoning aspects of our approach. The decision derives from the fact that knowledge is domain dependent, and fundamentally hierarchical in nature, with facts and concepts built upon one another. It can be argued that such hierarchical organization underlies all of human reasoning and is the basis for expertise in all domains. This appears to be true at all levels of cognition. We know, for example, that stimulus characteristics of the real world are filtered and averaged prior to perception. Human information processing is based on increasingly refined categorizations of input data, producing increasingly sophisticated/complex distinctions and generalizations about the data. In cognitive systems, symbolic computations are performed on abstracted data.

It is beyond the scope of this document to prove that abstraction hierarchies serve as the basis for all human reasoning. The argument presented here is essentially a simple one: conceptual structure and logical processes *can be represented in software* as organized through a hierarchical framework that defines the relationships between parts and wholes. This hierarchical framework is at least *implicit* in all cognition, and its characteristics can be used to replicate the observed properties of cognitive and logical systems.

At any rate, our goal is not to simulate or model human reasoning, nor to prove conclusively how human cognitive hardware works. Our goal is to enable machine reasoning on large unstructured information sets and thus support human reasoning. Machine reasoning over very large, distributed information sources (i.e., the Semantic Web) has been shown to be amenable to this type of hierarchy approach. By developing a robust representational format for knowledge (standard abstraction hierarchies) we will be able to develop algorithms that operate on the abstraction hierarchies independently of the content of these hierarchies. That is, the reasoning algorithms will work in any knowledge domain, once the relevant conceptualization for that domain has been cast into the standard format for our abstraction hierarchies.

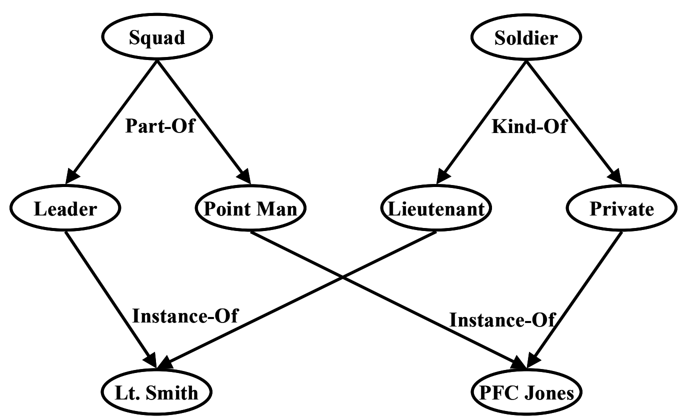


Figure 35. Example of Reasoning Across Boundaries

A second reason to use abstraction hierarchies for representing large amounts of information becomes clear when we try to display the information for human consumption. One of the most powerful aspects of using abstraction as an organizing principle is the inherent capability for selectively concealing and revealing complexity. Decisions about what to reveal and how to organize information will be based on the needs and goals of the user. We will implement mechanisms that allow control of 1) abstraction altitude: the level of detail about objects and processes that is relevant to a user's needs, and 2) abstraction perspective: the set of information about objects and processes that is relevant to a user's needs. Deciding what to reveal must be a dynamic process, as reasoning often involves moving up and down levels of abstraction altitude, and/or simultaneous consideration of multiple abstraction perspectives.

“Reasoning” in this context refers to formal logical processes (induction and deduction) as well as non-axiomatic processes (analogy, explanation, argumentation, refutation, revision, comparison, abduction, etc.) on abstraction hierarchies. As examples of reasoning across abstraction hierarchies, consider the following rational processes:

- Top-down/Bottom-up
  - Top-down looks for examples in input data that support or refute higher abstractions
  - Bottom-up looks for abstractions that are consistent/inconsistent with input data
- Deduction/Induction
  - Deduction "descends" the hierarchy from general principles to particulars
  - Induction "ascends" the hierarchy from particulars to generals
- Analysis/Synthesis
  - Analysis "takes wholes apart", differentiating high-level units into low-level "parts"
  - Synthesis integrally combines parts into higher-level wholes
- Generalization/Discrimination
  - Generalization identifies the common elements in a set of objects
  - Discrimination identifies the disparate elements in a set of objects
- Intuition/Expectation
  - An incompletely mapped bridging across many levels from lower to higher, often introspectively referred to as intuition
  - An incompletely mapped expectation or undefined bridging across levels from higher to lower
- Interpretation/Interpolation
  - Interpretation seeks the meaning or intent of some general high-level abstraction by defining it in terms of some particular context
  - Interpolation occurs when some general or higher-level category is inductively inferred or hypothesized from lower-level instances

3.6.4 Reasoning across fusion levels.

In reasoning about tactical situations, expert intelligence analysts routinely reason across fusion levels. For example, when reasoning at Level 1, the analyst may develop an estimate of the probabilities that an apparent radar return is caused by an actual target in the tactical environment, by weather conditions, or by anomalies in the radar equipment. The likelihood estimates may be influenced by his knowledge about the equipment, its state of operation (overheated?), or its state of repair. His judgments may be influenced by what he knows from independent sources about the weather in the target environment, and prior knowledge about interactions between known weather conditions and radar type. He is likely, however, to also consider information generally associated with Level 2 or 3 Fusion. What is the likelihood of there being an enemy platform in the apparent return area? If that likelihood is high, he may reduce his subjective likelihood estimate that the return is fallacious. Also, if the cost of finding an enemy platform in the return area is high, he may seek to bring additional sensors to bear on the area (Level 4 Fusion). Thus the analyst moves across fusion levels in reasoning about the situation.

Table -7. Categories of Reasoning

Declarative Existential	What things exist? e.g., tanks, troops
Declarative Relational	How do they interrelate? e.g., OPFOR, tank column
Procedural Tactical	What are they doing? e.g., shooting, retreating
Procedural Strategic	What is their objective? e.g., attacking, defending

Our reasoning agents currently operate by shifting among 4 levels of abstraction, corresponding roughly but not exactly to JDL fusion levels. For example, radar indicates an apparent return from a large object. The agent begins to assess the reliability of the event. Is it a real return, a weather related anomaly, or an artifact of the sensor system? Probabilities are estimated. A second return occurs. The agent adjusts probability estimates and notes that the probable large heavy object is moving toward the agent's location. Acoustic sensors report several explosions in the immediate vicinity of the agent's platform. The agent generates hypotheses to accommodate new information about the situation. The object may be a tank or other mobile weapons platform with projectile targeting capabilities. It may be a component of OPFOR, it may be shooting at the agent's platform, and it may be seeking to destroy the platform.

3.6.5 Ontologies and Data Models

A Data Model is the product of the database design process that aims to identify and organize the required data logically and physically. A data model says what information is to be contained in a database, how the information will be used, and how the items in the database will be related to each other. As database design is currently practiced, the items in the database (which correspond to entities in ontologies) are defined with sufficient precision to support formal and logical methods, but the relations are not. Database designers specify relations by using verb phrases such is "belongs to" or "can be derived from." These verb phrases are intended to be read and understood by humans, but they are not formally, axiomatically defined. Thus they cannot be operated upon by computational methods. In order to allow automated reasoning about entities, we propose to systematically express fusion-supportive relations in terms of predicate calculus. Predicate calculus is the branch of symbolic logic that deals with relations between propositions - especially the relation between subject and predicate of propositions. Symbols are used to represent the subject and predicate of the proposition, and the existential or universal quantifier is used to denote whether the proposition is universal or particular in application.

IDEF-5 provides the basis for our efforts in this area, but we extend IDEF-5 formal relations to include Modal logic, a form of logic that deals with sentences qualified by modalities such as possibly, necessarily, contingently, actually, can, could, might, may, must, ought, and others. Whereas more traditional forms of first-order logic work only with assertoric sentences (such as "Socrates is mortal," "This dog is a terrier," "All lizards are reptiles," etc.), modal logic also deals with the logical relationships between problematic statements, such as "It's possible that it will rain on Thursday" or "I can choose to go to the movies tomorrow," and apodictic statements such as "Every planet must have an orbit in the form of a conic section" or "if you add 2 and 2, the answer is necessarily 4."

The basic set of modal operators is usually given to be possibility, actuality, and necessity, a sentence is said to be

- Actual if it is true;
- Possible if it might be true (whether it is actually true or actually false);

- Necessary if it could not possibly be false;
- Contingent if it is not necessarily true, i.e., is possibly true, and possibly false. A contingent truth is one which is actually true, but which could have been otherwise.

3.6.6 Inference Relations in Ontologies

A class-level ontology may explicitly represent inference relations. If we define a complete ontology as one that explicitly represents all subclasses, a complete ontology has, by virtue of its completeness, all inference relations. This is because if there were an inference relation that was non-explicit, it would mean an instance set had a relationship different from the class to which it belonged which would imply that instance set is actually a subclass. However, most ontologic representations are not complete. Thus, many inference and influence relations are hidden within a superclass and inter-superclass relations. Our interest is whether an ontology, that is complete in its superclasses, can be used to reveal and explicate all the hidden inference relations.

The general problem is that an ontologic representation consists of nodes that are classes, attributes of classes, and instance sets within the class (implicit subclasses). These could be the axes of a influences / is-influenced-by (causes / is-caused-by) matrix. But what are instance sets? These are indicated by the Information Elements (IE) taxonomy that accompanies the ontology class and class-relation model. They are in the type-of taxonomy for each IE. Since an IE corresponds to one or more classes, the type of the IE indicates the instance sets of the object classes.

Even given an identification of axis elements for the influence matrix, we still do not have a proper inference network because it is necessary to convert it to a form that is compatible with a Markov DAG. The relationship between influence, connection, and path matrices and inference nets has not been explored prior to this SBIR Phase I research. What is fairly mature is the relationship between graphs and probabilistic models. Therefore, we wanted to develop the relationship between matrices and graphs in the hope that that would then, by transitivity, relate our connection matrix technique with probabilistic models. Then we would be able to use the power of linear algebra to manipulate the structure of the model while using the power of probability to keep the functional model mathematically rigorous.

The leading proponent of inference networks is Judea Pearl. The key step in constructing an inference network is to design a Markov relative DAG that can be proven to be compatible with probabilistic model constructed using a chain rule that imposes intervening variables. Our work has shown that this DAG can also be represented as a connection matrix, which we notate as G_c.

Now, associated with G_c is also a path matrix, which indicates which nodes there are paths between (G_p) and a distance matrix (G_d), which states that shortest distance between two nodes. In all cases, the matrices have directed variants, notated G_c^d, G_p^d, and G_d^d. It is easy to show that Gp is derived from Gc as follows:

Step 1: Perfom routine triangulation of G_c on the ith row as row 1 for which path connectivity is of interest. Call this matrix G_c^i.

Step 2: Then the nodes to which i has a path are those in the following with non-zero elements:

$$\begin{bmatrix} 1 \\ g_{12}^i \\ g_{12}^i g_{23}^i \\ g_{12}^i g_{23}^i g_{34}^i \\ \vdots \\ \prod_{\alpha=1}^{j-1} g_{\alpha\alpha+1}^i \end{bmatrix} G_c^i$$

where j is a row in the vector.

While a simple result, the advantage is to provide mathematical rigor to the otherwise verbal descriptions of graph properties. For example, acyclicity can now be stated mathematically, without visual reference to the graph as:

$$\text{G acyclic} \Leftrightarrow G_p^d \text{ is strictly non-symetric, that is,}$$

$$\underset{i,j}{\forall} \ni i \neq j, g_{ij} \neq g_{ji}$$

This is significant because causal diagrams that are acyclic are Markovian given their error terms are jointly independent [1]. An even more compact example is Theorem 1.2.8 from [71] stated as:

> *Theorem 1.2.8 (Observational Equivalence). Two DAGs are observationally equivalent if and only if they have the same skeletons and the same sets of v-structures, that is, two converging arrows whose tails are not connect by an arrow (Verma and Pearl 1990).*

Restated using our matrix notation:

$$G_c^d \underset{O}{\approx} H_c^d \Leftrightarrow \underset{i,j}{\forall} \left(\left(g_{ij} = h_{ij}\right) \vee \left(g_{ij} = h_{ji}\right)\right) \wedge$$
$$\underset{j}{\forall} \ni \left(\left(\sum g_{ij} > 1\right) \vee \left(\sum h_{ij} > 1\right)\right), \sum g_{ij} = \sum h_{ij}$$

The mathematical formulation of the Bayes network theory also has computer science advantage in that the connection matrix is directly storable and retrievable via a structure like the C2IEDM-RT's associative and multi-hypothesis structures. This ability could conceivably be useful in the future in deriving inference paths and, therefore, node triggers on-the-fly in the run-time system as new unforeseen situations are encountered.

4 CONCLUDING REMARKS

The E-2C aircrew have been increasingly tasked with a wider variety of more demanding missions. The result of this progression has been operator task saturation and overload. Multiple missions are conducted in parallel, with mission changes or priority changes occurring while the E-2C is on station. Interpreting the tactical situation and responding to changes is still a manual process. Currently, the operator must mentally fuse all information (Tracks, Intelligence – pre-briefed and real-time, Identifications, Threats – AIR/SURF/SUB, Alerts, Geography, Atmospherics, Stationing Considerations, System Performance – RADAR/IFF/ESM/CEC/LINK4/11/16, SATCOM Data, Voice Communications, etc.)

The next-generation E-2E Advanced Hawkeye (AHE) will possess new radar, Theatre Missile Defense (TMD) capabilities, multi-sensor integration, and a Northrop Grumman Navigation Systems tactical cockpit. The new radar is being developed under the E-2C Radar Modernization Program (RMP) and will be a solid state radar with improved performance in the presence of land clutter and casual and intentional Electromagnetic Interference (EMI). AHE also includes CNIR the Identification Friend or Foe (IFF) system, SATCOM Data Links (TDDS, TIBS, TOP, OBP), data Links 4/11/16, Electronic Support Measures (ESM), Cooperative Engagement Capability (CEC), upgrades to the communications and navigation systems, and possibly an Infra-Red Search and Track (IRST) system used primarily to detect and track Theater Ballistic Missiles (TBM's). New fusion software in the AHE configuration includes Multi-Sensor/Multi-Source Integration (MSI) software used to fuse Radar, Identification Friend or Foe (IFF), Cooperative Engagement Capability (CEC), Electronic Support Measures (ESM), Link 4, Link 11, Link-16, and SATCOM data feeds (data received via the MATT terminal from TDDS, OBP, TOP, and TIBS).

Algorithms and software in-progress or under consideration include:

- cross cueing of multiple onboard sensors;
- advanced resource management techniques;
- organic resource requests from external sources;
- Classification and/or Combat Identification techniques;
- Automated Tactical Decision Aids (i.e. Air Tasking Order, Rules of Engagement, Order of Battle, EPL/NERF, etc.),
- advanced sensor suite architectures required to support next generation Airborne Early Warning requirements (RADAR, IFF, ESM,, links, CEC, etc.).
- access to mission information is planned for the cockpit crew as part of the tactical cockpit modifications

Additional data relevant to fusion is under consideration such as the Air Tasking Order, Special Instructions, Airspace Restrictions, Rules of Engagement, Battlefield Situation Reports, and intelligence briefings prior to takeoff. The MSI/Fusion system should provide one displayed track for each representative target in the battlespace, resulting in a stable surveillance picture with no duel tracks or multiple depictions of the same track. The MSI/Fusion system should result in more accurate and continuous tracking than each sensor or source of information can produce individually. Also, the identification picture should be enhanced by accurately associating track attribute data as received by all the sensors and sources. The MSI/Fusion system will build a solid foundation for automated target identification in support of a Combat ID decision. Furthermore, the data presented to the operator should be intuitive, and it should allow the operator to display threat information and alerts based on individual mission priorities. Finally, the MSI/Fusion system should provide an uncluttered intuitive display, resulting in increased Situational Awareness and reduced operator workload. Decreased operator workload means more time for tactics and execution of mission requirements as demanded by the Battle Group or Theater Commanders. Increased Situational Awareness means more timely and accurate decisions through early identification of hostile forces and optimal use of friendly forces.

5 REFERENCES AND ENDNOTES

[1] "A Simplified Turn Logic", *Note N-1090*, Fleet Computer Programming Center, Pacific, San Diego, CA 1963

[2] "Communities of Interest in the Net-Centric DoD Frequently Asked Questions (FAQ), Draft", OASD (NII) / DoD CIO, 13 March 2004

[3] "DoD Net-Centric Data Management Strategy: Metadata Registration", memorandum, OASD (NII) / DoD CIO, 3 April 2003

[4] "The DoD Net- Centric Data Strategy And Discovery & Mediation Enterprise Services", briefing by OASD (NII) / DoD CIO, 3 Nov 2003

[5] A metric space, A, is one in which is defined a function * * * x, y * A: (1) *(x, y) > 0, (2) *(x, y) = 0 iff x and y are identical, (3) *(x, z) < *(x, y) + *(y, z). Epstein; Linear Functional Analysis: An Introduction to Lebesgue Integration and Infinite Dimensional Problems; W. B. Saunders Company; 1970

[6] Barr, Arron, and Feigenbaum, Edward A.; The Handbook of Artificial Intelligence, Vol. I., William Kaufman, Inc., Los Altos, CA, 1982.

[7] Bar-Shalom, Y., Fortmann, T.; Tracking and Data Association; Academic Press, Inc., San Diego, CA; 1988

[8] Bar-Shalom, Y., X. Rong Li, Thiagalingam Kirub, Estimation with Applications to Tracking and Navigation, John Wiley and Sons, 2001

[9] Bassett, Dave (MAJ); "FCS One Team-The Army/Defense/Industry, Unit of Action Network Systems Integration Update"; briefing at the Combatant Command Information Superiority Conference (CISA); PACOM, HI; May 2004. (In questions and discussions between Associate Investigator and briefer, the "common object model" was said to be C2IEDM.)

[10] Borgida, A., Mylopoulos, J., and King, H.; Generalization/Specialization as a Basis for Software Specification" in On conceptual Modelling: Perspectives from Artificial Intelligence, Database, and Programming Languages; ed. by Brodie, M., Mylopoulos, J. and Joa Chim, W.; Springer-Verlag; 1984

[11] C2IEDM documentation and model at Multilateral Interoperability Program: http://www.mip-site.org/

[12] CADM documentation and model at https://pais.osd.mil/enterprisearchitectures)

[13] Carey, S., Kleiner, M., Hieb, M., Brown, R; "Standardizing Battle Management Language – Facilitating Coalition Interoperability"; *MSIAC's M&S Journal Online*; Winter 2004

[14] Charniak, E. and McDermott, D.; Introduction to Artificial Intelligence; Addison-Wesley Publisheing Company, Inc.; 1985.

[15] Community Imagery Needs Forecast (CINF), Air Sovereignty Operations Center (ASOC), and others

[16] CORBA is the acronym for Common Object Request Broker Architecture, OMG's open, vendor-independent architecture and infrastructure that computer applications use to work together over networks.

[17] DDB-2600-4537-90; Military Intelligence Integrated Data System (MIIDS) and Integrated Data Base (IDB) Definition and Specification; August 1990

[18] Defense Data Architecture (DDA), documentation and models available at the Net Centric XML repository available at http://diides.ncr.disa.mil/

[19] Delong, P., "Interoperability and Sensor Fusion", *Naval Engineers Journal*, Vol 115, Numbers 2, 3, and 4, 2003.

[20] Department of Defense Architecture Framework, Version 1.0, OASD (NII) / DoD CIO, 9 Feb 2004

[21] Department of Defense Discovery Metadata Specification (DDMS), Version 1.0, Deputy Assistant Secretary of Defense (Deputy Chief Information Officer), 29 September 2003

[22] Department of Defense Net-Centric Data Strategy, OASD (NII) / DoD CIO, 20 April 2003

[23] Dickman, Bruce; "Leveraging MIP to Support JC2 Foundation Strategy"; briefing to the Office of the Assistant Secretary of Defense (OASD) for Network Infrastructure and Integration (NII); Arlington, VA; May 2004

[24] Discussions between proposed associate investigator and Dr. Frank Perry, a principal in GCCS development.

[25] Documentation and model at http://www-datadmn.itsi.disa.mil/datadmn/dda/c2core.html.

[26] DoD Architecture Framework, Version 1.0, O (NII) 8 February, 2004

[27] DoD Data Administration", DODI 8320.1, OASD (NII) / DoD CIO, 26 Sept 1991

[28] DoD Joint Technical Architecture, Version 5.0, dated 4 April 2003.

[29] DoD XML Registry at http://diides.ncr.disa.mil/

[30] Dr's Hans Keithley and Dale Walsh, respectively.

[31] E. Waltz and J. Llinas. Multisensor Data Fusion. Artech House, Norwood, MA 1990

[32] Endsley, Mica R., "Toward a theory of situation awareness in dynamic systems. Human Factors, 37(1), 32-64, 1995.

[33] Final Report of the CNO NTDS Functional Allocation Study Group (Final Report of the NTDS Functional Allocation (Intra-Ship) Study); 5 March 1981

[34] *Functional Description Document for Track and Relational Data Synchronization*, Version 1.0, August 26, 1998, Prepared for Defense Information Systems Agency (DISA) by FGM, Inc

[35] Genesereth, M., "Knowledge Interchange Format" draft proposed American National Standard (dpANS), NCITS.T2/98-004

[36] Hierarchical defaults are used throughout to accommodate incomplete INTEL data, an almost fundamental characteristic of INTEL databases.

[37] In the sense that the recursion is applied to the last cycle "state estimate". There is no process noise because the process model for identification has no assumptions analogous to the assumption of constant velocity. The model is static, not dynamic.

[38] Information Integration for Concurrent Engineering (IICE), IDEF5 Method Report, Armstrong Laboratory (AL/HRGA), Wright-Patterson Air Force Base, Ohio, 21 Sept 1994

[39] Joint Battle Management Command and Control (JBMC2) Roadmap, Office of the Under Secretary of Defense (Acquisition Technology and Logistics) and US Joint Forces Command (USJFCOM), Draft Version 1.0, 12 April 2004.

[40] Jonker, R., Volgenant, A., "A Shortest Augmenting Path Algorithm for Dense and Sparse Assignment Problems," *Computing*, Vol. 38, 1987

[41] Keithley, H., "An Evaluation Methodology for Fusion Processes Based on Information Needs", in Handbook of Multisensor Data Fusion, ed. By Hall, D. and Llinas, J., CRC Press, New York, 2001

[42] Keithley, H., Multi-INT Fusion Performance, Joint C4ISR Decision Support Center, OASD(C3I), Washington, D.C., 2001

[43] Mahler, R.; An Introduction to Multisource-Multitarget Statistics and it Applications; Lockheed Martin; 2000

[44] McDaniel, D., "Multi-Hypothesis Database for Large-Scale Data Fusion", Proceedings of the Fifth International Conference on Information Fusion, International Society of Information Fusion, Sunnyvale, CA, 2002

[45] McDaniel, D., and Schaefer, G.,"Real-Time DBMS for Data Fusion", *Proceedings of the National Symposium on Sensor and Data Fusion*, Infrared Information Analysis Center, 2003

[46] McDaniel, D., *Electronic Warfare IDentification (EWID) Small Business Innovative Research Final Report*, Space and Naval Warfare Systems Command, 1996.

[47] McDaniel, D., Tooey, M., "Advanced Combat Direction System (ACDS) Block 1 Tactical Information Integration and Control Group (TIICG) Algorithms Timing Analysis", *Technical Repor*t, American Defense Systems, Inc., San Diego, CA, 1987

[48] Meaning the operator selection imposes a constraint on the normal ranking logic.

[49] Multi-INT Fusion Performance, Joint C4ISR Decision Support Center, OASD (C3I), Washington, D.C., 2001

[50] Multilateral Interoperability Program Available: http://www.mip-site.org/

[51] Multisensor Data Fusion; Edward Waltz & James Llinas; 1990

[52] Multi-Source Integration System Engineering Team (MSI SET), prepared for Combat Systems Functional Allocation Board (CSFAB), 3 November 1997.

[53] Naval C4ISR Operational Architecture, Space and Naval Warfare Systems Command (SPAWAR), 1997

[54] Naval Ocean Systems Center (NOSC); ACDS Block 1 Algorithms Analysis Report; QuesTech, Inc. CDRL under contract N66001-85-D-0204, DO 912; 31 August 1987.

[55] Naval Ocean Systems Center (NOSC); ACDS Block 1 Algorithms Analysis and Modelling Report; QuesTech, Inc. CDRL under contract N66001-85-D-0204; September 1988.

[56] Naval Ocean Systems Center (NOSC); ACDS Block 1 System Acceptance Test (SAT) Specification; QuesTech, Inc. CDRL under contract N66001-85-D-0204; May 1989.

[57] Naval Surface Warfare Center Dahlgren Division (NSWCDD), Open Architecture Document Set, Program Executive Office, Integrated Warfare Systems, Version 1.0 Draft, Program Manager, Open Architecture (OA), 7 May 2004

[58] Naval Warfare Tactical Data Base (NWTDB) Management Plan; September 1992

[59] Naval Warfare Tactical Data Base (NWTDB) Standards Manual, Volume 1: Tactical Information Management; July 1993

[60] NAVSEA 0967-LP-027-8602; Combat Direction System Model 5 Systems Engineering Handbook, Volume 1, CDS Role in Battle Force Combat Control and Systems Engineering Introduction; February 1985

[61] NAVSEA 0967-LP-027-8627F; System Specification for CV/CG Block 1 Advanced Combat Direction System (ACDS), Revision F, Change 1; 1 November 1994

[62] Nichols, M., "A Survey of Multisensor Data Fusion Systems", in Handbook of Multisensor Data Fusion, ed. By Hall, D., and Llinas, J., CRC Press, 2001

[63] NTDS Functional Allocation Study, Chief of Naval Operations, 1980

[64] OAML-DBD-43; Data Base Description for Historical Temporal Shipping (HITS); April 1993

[65] Ontology and Fusion (McDaniel, D., "Multi-Hypothesis Database for Large-Scale Data Fusion", Proceedings of the Fifth International Conference on Information Fusion, International Society of Information Fusion, Sunnyvale, CA, 2002)

[66] Open Architecture (OA) Functional Architecture Definition Document (Version 2.0), Program Executive Office, Integrated Warfare Systems, Program Manager, Open Architecture (OA); 1333 Isaac Hull Avenue SE, Washington Navy Yard, DC 20376-2301 November 2003

[67] Open Architecture (OA) Functional Architecture Definition Document (Version 2.0), Program Executive Office, Integrated Warfare Systems, Program Manager, Open Architecture (OA); 1333 Isaac Hull Avenue SE, Washington Navy Yard, DC 20376-2301 November 2003

[68] Open Architecture Volume 4, Open Architecture Computing Environment Technologies and Standards, Version 1.0 ; May 2004.

[69] Open Systems Joint Task Force Available: http://www.acq.osd.mil/osjtf/

[70] OPNAVINST 3430.23B; Tactical Electronic Warfare Reprogrammable Library (EWRL) Support Program; 12 June 1992

[71] Pearl, J., *Causality*, Cambridge University Press, NY, NY, 2000

[72] Pope, A., The CORBA Reference Guide: Understanding the Common Object Request Broker Architecture, Addison Wesley

[73] Program Performance Specification for AN/SLQ-32(V) Integrated Operational Program, Volume 1, Digital Processor Unit Operationa Program; March 1985

[74] Rawlings, T., "Report on Parameter Search Methods for EW SSI", Block 1 Combat Direction System Engineering Notebook, Hugest Aircraft Company, Fullerton, CA, 1986

[75] Rawlings, T., "Track Position Correlation System", Block 1 Combat Direction System Engineering Notebook, Hugest Aircraft Company, Fullerton, CA, 1986

[76] see the ontology track in *Proceedings of the Fifth International Conference on Information Fusion*, International Society of Information Fusion, Sunnyvale, CA, 2002

[77] see the ontology track in Proceedings of the Fifth International Conference on Information Fusion, International Society of Information Fusion, Sunnyvale, CA, 2002

[78] Sensor-track to sensor-track correlation is not performed in this elementary stage experiment.

[79] Simpson, P., Artificial Neural Systems; Pergamon Press: Oxford, 1990

[80] STANAG No. 5616, NATO Standardization Agency (NSA) Standardization Agreement (STANAG), Tactical Data Exchange – LINK 16, Edition 3

[81] Stankovic, J., Hyuk Son, S, Hansson, J., "Misconceptions About Real-Time Databases", *IEEE Computer*, June, 1999

[82] System Requirements Document (SRD) for Common Link Integration Processing (CLIP), Program Executive Office (PEO) for C4I and Space, 02 August 2004

[83] Terrain tailoring of probability density functions could be a simple as shorelines (e.g., WVS) , could consider elevation (e.g., DTED), or could use terrainability data (e.g., those generated by US Army Topographic Engineering Center).

[84] The logical constraints play a role similar to "half-order theory" in mass spectroscopy [barr].

[85] The TimesTen Team, "High Performance and Scalability through Application-Tier, In-Memory Data Management", Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt, Morgan Kaufmann, 2000

[86] This consideration is deferred in this elementary stage experiment.

[87] U.S. Army Topographic Engineering Center; Request for Proposal for War Breaker Multiple Intelligence Correlator (MICOR); USATEC DACA76-93-R-0020; November, 1993.

[88] Ultimately, the project settled on a non-linear Kalman, called the Multi-State Space (MSS) tracker that, in effect, used a Cartesian motion model and a polar observation equation. This was verified to perform roughly equivalently to the iterated extended Kalman described in many references such as Gelb, A. (ed); Applied Optimal Estimation; The Analytic Sciences Corporation; 1974.

[89] Updated Data Distribution Service Final Adopted Specification; dated 7 July 2003; http://www.omg.org/docs/ptc/03-07-07.pdf.

[90] US Joint Forces Command (USJFCOM) and Office of the Under Secretary of Defense (Acquisition Technology and Logistics); "Joint Battle Management Command and Control (JBMC2) Roadmap"; RAND Corporation, Arlington, VA; April 2004

[91] Vaccaro, D.; Electronic Warfare Receiving Systems; Artech House, Inc.; 1993.

[92] Walker, R., Loaiza, F., Simaitis, E., "Development of a Far-Term Data Model for the Global Command and Control System (GCCS)", IDA Paper P-3047, Institute for Defense Analyses, Alexandria, VA, 1995.

[93] Wallenius, K., "Support for Situation Awareness in Command and Control," www.nada.kth.se/~klasw , 2004

[94] Wiley, R., *Electronic Intelligence: The Analysis of Radar Signals*, Artech House, Norwood, MA, 1993

[95] www.timesten.com."