29th International Command and Control Research & Technology Symposium

24-26 September 2024

London, UK

Submission number 038

Concept Paper

Topic 6: Emerging Concepts and Theories – System Architecture

Paper Title: **An Open Architecture for Data Fusion**

Authors**:**

Steven W. Mitchell	Tod Schuck
Lockheed Martin	*Lockheed Martin*
steve.mitchell@lmco.com	tod.m.schuck@lmco.com

An Open Architecture for Data Fusion

Steven W. Mitchell
Lockheed Martin
steve.mitchell@lmco.com

Tod Schuck
Lockheed Martin
tod.m.schuck@lmco.com

Abstract

While the mathematics of fusion is quite general, every practical fusion engine must be tuned to a particular application to achieve acceptable performance with real data on tactical hardware. Tuning is typically achieved for a specific set of sensors and for specific track types of interest. For example, the fusion engine for a fifth-generation fighter is tuned for that aircraft's sensors and for the types of air-to-air tracks it needs to engage. The result is a fusion engine which may perform very well in a bounded subset of the multidimensional data space generated by the parameters of the sensor measurements and the track attributes, while performing less efficiently or even failing in other parts of that data space.

A joint all-domain command and control system requires addressing a much larger subset of the potential multidimensional data space than that processed by contemporary practical fusion engines. At the same time, there is pressure to re-use existing fusion engines in which a great deal has already been invested.

This paper describes how multiple sensor fusion engines can be combined in an open architecture to provide the best overall estimate of the fused sensor track picture while managing the overall computational requirements. It describes how the component fusion engines can be calibrated, the open multi-fusion-engine framework adjusted to ensure that each component engine is fed the data for which it is best suited, and how the outputs from the various fusion engines can be correlated to deliver the final fused track picture.

1 Introduction

While the mathematics of fusion is well published and generally understood, every practical fusion engine must be tuned to a particular application to achieve acceptable performance on anything larger than a trivial dataset. As a result of real-world effects, a fusion engine that may perform very well in a portion of the multidimensional data space {position x velocity x acceleration x radar signature x SIGINT parameters x …} generated by the parameters of the sensor measurements and track attributes, may perform less well or even poorly in other parts of that data space. A universal high-performance fusion engine does not exist except for very focused applications [1].

Tuning is typically achieved over a specific set of sensor inputs and a specific set of track types of interest. For example, the fusion engine of a fifth-generation fighter might be tightly tuned for that aircraft's sensors, mission computer, and for the types of air-to-air tracks it needs to detect, track, and engage. That tuning process requires deep understanding of the fusion technology, sensor capabilities, and the specific application, and typically takes significant investments to succeed. Within the portion of interest of the multidimensional data space the fusion engine may perform phenomenally well. Outside that portion it may be inefficient, or even completely fail.

Unlike a fifth-generation fighter, a C4I system for Joint All-Domain Command and Control (JADC2) is necessarily concerned with a much larger portion of the multidimensional sensor and track data space. Typically it must ingest data from a range of distributed sensors of various types, and must track a wide variety of entities across multiple domains. Due to the contested nature of the communications and cyber environment, the fusion component of the JADC2 system may need to function distributed across multiple C2 nodes, maintaining a coherent tactical picture (CTP) when possible and resynchronizing that picture after disruptive events. All of this must be accomplished with the limited computational resources that can be deployed to the tactical edge.

Both the cost of developing and tuning fusion engines and the shrinking timelines available for deployment at the speed of relevance demand re-use of existing high-performance fusion engines, despite the bounds on their applicability. These conflicting requirements drove the development of the open architecture for fusion management discussed here.

This paper is organized as follows: Section 2 provides background on the concept of "what is fusion?", modern fusion engines, the compromises required to build efficient implementations to support real-time tactical applications, and the conflict between those compromises and the needs of JADC2. Section 3 describes an open architecture for efficiently executing multiple fusion engines in parallel while managing computational requirements in tactical applications. Section 4 describes how practical tactical fusion engines can be characterized, while Section 5 discusses how that characterization can facilitate re-use of existing high-performance fusion engines in that open architecture fusion management framework. Section 6 describes potential on-going work to implement and test this approach.

2 Background

The goal of data and information fusion has been defined as "the process of combining incomplete and imperfect pieces of mutually complementary data in such a way that a better understanding of an underlying real-world phenomenon is achieved" (Wolfgang Koch, Fusion 2011 Tutorial Workshop). Specifically, the overarching goal of sensor fusion is to attain an accurate characterization of detected objects in the joint battlespace so that high-confidence, timely application of military options and weapon resources can occur [2]. This problem is exacerbated by the raw speed of modern warfare where the time between a weapon engagement decision and target impact is exceedingly compressed. Improving the tactical picture is the same as improving information processing; thus, this is the purpose of performing fusion on all the sensor information that is available that can help develop the tactical picture.

A solid approach to consider for a fusion process and architecture design evaluation is to examine what Hall and Steinberg composed in 2001 in their seminal paper "Dirty Secrets in Multisensor Data Fusion" [3], which discusses a series of dictums about information fusion. For the purposes of our paper, some of these include:

A. There is no substitute for a good sensor

B. Downstream processing cannot fix upstream errors or failures

C. Sensor performance information is critical for sensor fusion applications

D. There is no one "right" fusion algorithm or approach

From a systems process perspective, a reliable method to achieve successful track fusion is to fully profile the performance of the sensors and sources and the type of data that will be presented for fusion. [4]. A lot of the challenge is to quantify the "known unknowns" and design in provisions for "unknown unknowns". So the design for resiliency in a fusion architecture is extremely important. [5]. This may not be possible with a single fusion engine (FE) or approach.

2.1 Modern Fusion Engines

A helpful visualization of a successful modern track and classification fusion process is shown in Figure 1[1].

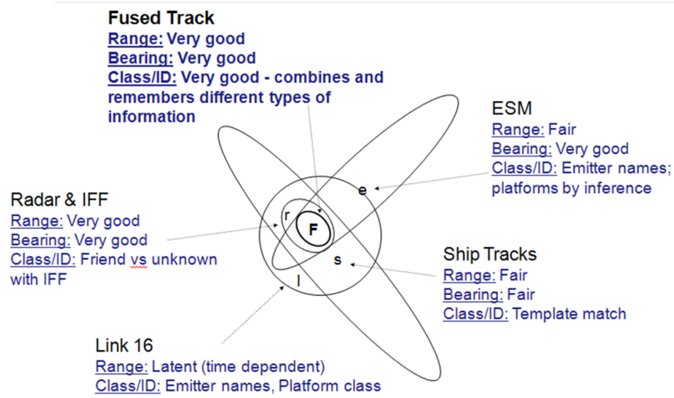

Figure 1 – the expected result of a resilient and robust modern fusion capability

In this illustration, the uncertainty in an object's location – denoted by the central ellipse labeled "F" – is reduced significantly via the combining or fusion of disparate sets of sensor information. There is a spatial orthogonality to the kinematic and classification information allowing for optimization of the intersections of the uncertainty ellipses. There is also modality diversity between sets of information like Identification Friend-or-Foe (IFF) which is cooperative, Electronic Warfare (EW) which is unintentionally cooperative, and radar which is non-cooperative that enable more complex automated decisions based on the information characteristics present and that the error trends differ [6].

Specifically, for track and ID fusion, Figure 2 is an excellent way to distinguish it relative to the other fusion levels and how multi-modal orthogonal data works when combined to improve tracking and situational assessment (SA) [7].

[1] Figure generated at LM for use in a variety of tutorials

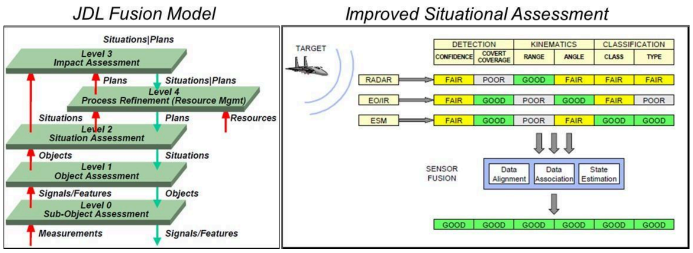

Figure 2 - JDL Fusion Levels and Orthogonal Data for Improved Tracking and Situation Assessment

Referencing Figure 2, the Joint Directors of Laboratories (JDL) Level 1 (L1 – track and ID) *Object Assessment* is also called *Object Refinement*. This processing level combines information from the results of level 0 processing within sensors. This level is about fusing the attributes of objects to detect, locate, characterize, track, and classify these objects. This level of processing involves information assignment/correlation/fusion (per Figure 1) and taxonomic classification such as F-16, CG-47, etc. This means assigning an identity to an object from a taxonomy—a set of mutually exclusive possibilities with probabilities associated with its uncertainty. Generally, to fuse information at this level, multiple hypotheses must be tracked due to uncertainties in predicted positions of objects and their identity.

Automating the fusion and declaration decision process flexibly across L1, as information becomes available and without throwing away valuable information and context, is challenging. L1 fusion performance is further compounded by the challenges with real-time distribution of fusion capabilities across a large System of Systems (SoS) enterprise covering thousands of miles of land and ocean and extending from sub-surface to space domains. This point is illustrated in the conceptual Coalition and Joint Multi-Domain C2 enterprise shown in Figure 3.

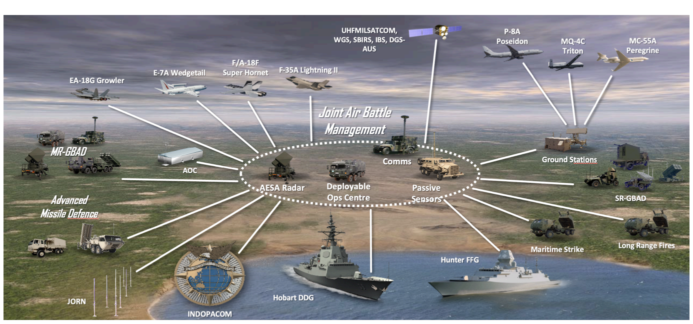

Figure 3 – C/JADC2 Network-Centric SoS OV-1

The challenge, as shown in Figure 3, is to develop a L1 fusion architecture and a set of algorithms that integrates many disparate sensor and source reports in four warfare areas: Space, Air, Ground, and Sea-Surface, so warfighters at all levels of the chain of command can readily understand the current tactical situation and make decisions based on that understanding. Further, this L1 fusion system must work within a dynamic and heterogeneous networked SoS structure as information sources will provide diverse quality and types (e.g., tracks, Associated Measurement Reports (AMRs), measurements) of information across multiple information dimensions and security levels. Figure 3 is also an illustration of the challenges for a successful JADC2 deployment. This is considered further in section 2.3.

2.2 Algorithms Constrained to Achieve Performance

With a distributed, multi-sensor data fusion architecture, the problem of constrained computational nodes is evidenced. Referring to Figure 3, there are varying degrees of tactical hardware capabilities across the SoS. Yet, for most of these sensor and processing nodes, the need for "real-time" computational capabilities is the same.

The computational challenge for the heterogeneous platforms in Figure 3 requires a scalable selection of data fusion capabilities. The computational platform and sensor variation on a *Hobart* class DDG is very different from a E-7A *Wedgetail*. Both are very different from an unmanned MQ-9B. Real-time processing at the edge is still required regardless of the platform/node capabilities.

Recently, the development of the Internet-of-Things (IoT) and the concept of "smart cities" has ushered in an architectural concept similar to Figure 3 where multiple sensors and sources contribute to a fused product. Instead of a 5th generation stealth fighter communicating to an Operations Center, the IoT might connect a smart coffee maker with a refrigerator that monitors food consumption. Information could be fused there in order to automatically purchase more coffee through a smart Internet device (or potentially the fusion resides in the smart Internet device).

A major concern in this multi-sensor fusion IoT architecture is *energy efficiency* [8] [9] [10] [11]. This is also a major concern for tactical edge-based fusion and communications due to the need to minimize Space, Weight and Power (SWaP) for deployable, portable, and packable computational form factors. As the discussion in section 2.3 evolves, the necessity of a fusion architecture to scale the types and numbers of fusion engines becomes clear.

2.3 Breadth of the Fusion Problem in JADC2

In a real-world, networked, real-time, tactical and operational fusion enterprise (as shown in figure 3), the following are considerations for this enterprise [12]:

1. Data type processing (measurements, AMRs/tracklets, tracks)
2. Sensor and source report dimensionality (1D, 2D, 3D) including
    a. Electronic Intelligence (ELINT)
    b. Electronic Support Measures (ES or ESM)
    c. Communications Intelligence (COMINT)
    d. RADAR (both active and passive)
    e. Air, Ground, and Surface Moving Target Indicators (AMTI, GMTI, SMTI)
    f. Photonics trackers (Infrared Search and Track (IRST))
3. Association and dissociation of tracks
4. Track maintenance and evaluation including intermittent sensor reporting
5. Out-of-time order measurements and track updates
6. Time-late measurements and tracks
7. Data integrity (data ringing)
8. Non-organic and organic information combining
9. Environmental affects (e.g., noise, jamming, fading)
10. Organic and networked information processing across representative nodes

Also, these systems characteristics are necessary to fully evaluate SoS applicability of fusion engines and may not be available:

A. Source code and documentation
B. Maturity of fusion product
C. Fielded systems using the fusion product
D. Ease of use via open interface design and extensibility
E. Resiliency to unexpected inputs and corrupted information
F. Stability of networked operations in corrupted environments
G. Scalability of the system especially for distributed fusion operations
H. Distributed design for organic and networked sensors/sources and C2 nodes
I. Communications bandwidth usage and availability
J. Information (sensor and source) availability

We envision that a practical fusion enterprise will require multiple fusion engines/processes whose performance will be optimized for various modalities and environments.

3 An Open Architecture for Managing Fusion

We have seen that a practical JADC2 system needs to accommodate multiple fusion engines. The compelling arguments for modular open architectures are well known [13], so an open fusion architecture can provide optimum flexibility and speed to deployment for the system developer, the acquisition agent, and the end user.

Let the multidimensional data space of interest be MDS. For a given fusion engine (FE_i), call the portion of the MDS where it performs well $MDSP_i$. The individual fusion engines which this architecture combines may be tuned for a discrete partitioning of the multidimensional data space of interest to the joint commander,

$$\cup\ MDSP_i = MDS,\ \cap\ MDSP_i = \emptyset$$

they may be tuned for portions that are disjoint but do not cover that space,

$$\cup\ MDSP_i < MDS,\ \cap\ MDSP_i = \emptyset$$

or they may be tuned for portions which overlap.

$$\cap\ MDSP_i \neq \emptyset$$

In what follows, the collection of fusion engines is referred as a “constellation”.

The open architecture for managing fusion wraps a fusion management framework around this constellation of fusion engines to provide good quality, high performance fusion over the union of their tuned regions $\left\{\cup\ MDSP_i\right\}$.

The Open Fusion Architecture service resource flow in Figure 4 describes how multiple sensor fusion engines are combined in an open architecture to provide the best overall estimate of the fused sensor track picture while managing the overall computational requirements. The basic concept is that the *Sensor Measurement Routing Service* directs incoming sensor measurements from any source to the fusion engine(s) which can best process them. Each fusion engine processes the sensor measurements it receives as it would in a traditional single-fusion-engine deployment. The fusion engines generate track reports and provide them to the *Select Best Fused Track Estimate Service.* When this service receives multiple estimates of the same track, it selects the best estimate using an appropriate metric. Then it delivers the combined fused track picture to the (Common Track Picture) *CTP Service*.

option where there are not sufficient high-performing fusion engines to cover the MDS.

Because this architecture is open, the constellation of fusion engines can easily be changed over time as new engines become available that improve overall performance, either by improving coverage of the MDS or by replacing outdated engines. When the constellation is changed, only the *Sensor Management Routing Service* needs to be updated.

If the portions of the MDS for which the individual fusion engines are disjoint, the total computational complexity of this process is bounded by the computational complexity of the least efficient component FE_i, scaled by the frequency of tracks in $MDSP_i$. If there is overlap in the tuned portions, the growth in complexity should be linear in the size of the overlap, assuming a uniform distribution of tracks of interest.

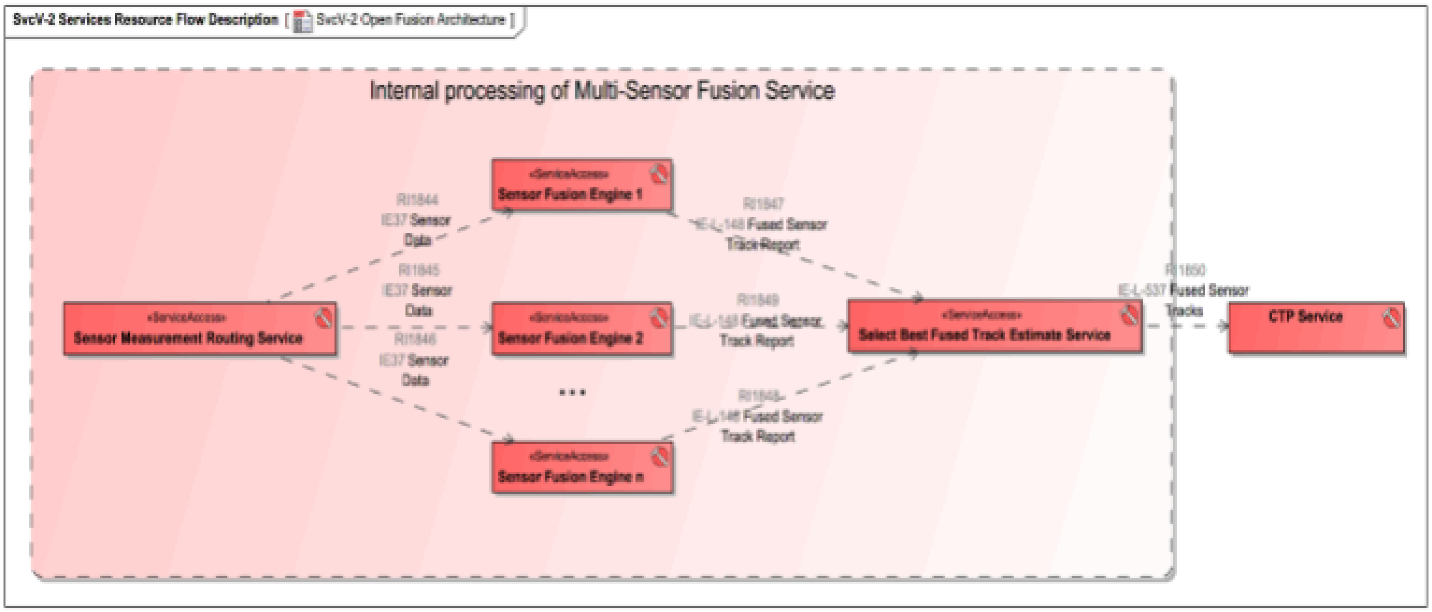

*Figure SEQ Figure * ARABIC 4: Open architecture for management of multiple fusion engines*

This architecture requires that the individual Sensor Fusion Engines in the constellation be characterized against a set of standardized scenarios to determine the portion of the multidimensional data space of interest ($MDSP_i$) in which they perform well. That information is used by the *Sensor Measurement Routing Service* to ensure that each fusion engine is fed the data for which it is best suited. Where the incoming sensor data falls into an area that overlaps the $MDSP_i$ for more than one fusion engine, the *Sensor Measurement Routing* will feed it to each of the appropriate fusion engines in parallel.

For the case where the tuned regions of the available high-performance fusion engines do not cover the MDS, lower performing untuned engines can be added to the managed fusion engine constellation to address sensor measurements in the remaining data space of interest. This will incur an overall performance penalty, but is an

4 Characterization of a fusion engine

The Single Integrated Air Picture (SIAP) program [14] has established quality measures for tracking and classification. It should be noted that there are other sources that can also contribute additional quality metrics. However, these metrics have been established as the reference for quantifying kinematics and classification/ID performance. The cited reference provide formulaic methods to quantify these measures. These equally apply to L1 fusion performance. These baseline metrics are defined at a high level for both centralized and network-based fusion architectures as follows:

Completeness – All kinematic (e.g., position, velocity) and attribute (e.g., identification, classification) information is incorporated in a System Track.

Correctness – All kinematic and attribute information from a given real-world object is assigned to a System Track.

Clarity – No ambiguous or spurious tracks are generated, and tracks contain no ambiguous or conflicting identification (ID) .

Continuity – Track associations are maintained if an object continues to be observed by one or more sensors .

Accuracy – Estimated position and velocity (or angular) accuracies are maintained in System Track.

Latency – Measured from track initiation, update, and error. This is the time difference from a detection (or similar threshold) to the status of initiation, update, and an established number of standard deviations in error.

It is expected that these metrics are expanded to account for a fully distributed L1 fusion architecture. However, the core metrics are sufficient and extensible to a distributed architecture.

A challenge for a distributed architecture is its ability to rank the requirements for these metrics and establish the scope and performance priorities. So for an enterprise fusion problem like JADC2, the following requirements and performance rankings should be considered:

Essential. Indicates a requirement which the JABMS **must** be capable of meeting, to the full extent specified, to achieve the **minimum** level of functionality and performance of the service. Failure to meet any Essential requirement will exclude a FE or multiple FE capabilities from contention.

Very Important. Indicates a requirement in which the JABMS function is considered to be **Essential**, but the performance to which this function is achieved may be considered to be of Important value. Failure to meet a *Very Important* requirement does not (individually) make a FE non-compliant, but in the aggregate may have a large impact on architecture possibilities.

Important. Indicates a requirement which the JABMS should be capable of meeting, to the full extent specified, to achieve the intended level of functionality and performance of the service.

Desirable. Indicates a requirement which the JABMS does not need to meet in order to achieve the intended functionality and performance of the service, but which provides additional benefit, such as in future anticipated needs.

Information. Indicates text which has been provided for information only, in order to provide context for a section of the specification. It may also guide a requirement which the JABMS should demonstrate adherence to. Compliance to information as design principles will necessarily be assessed more subjectively than a simple 'pass/fail' for each; however, the relative level of compliance to Design Principles may still influence architecture development.

Referencing section 3, the open architecture characterization of a given data fusion engine is shown in Figure 5 via an Internal Block Diagram (ibd).

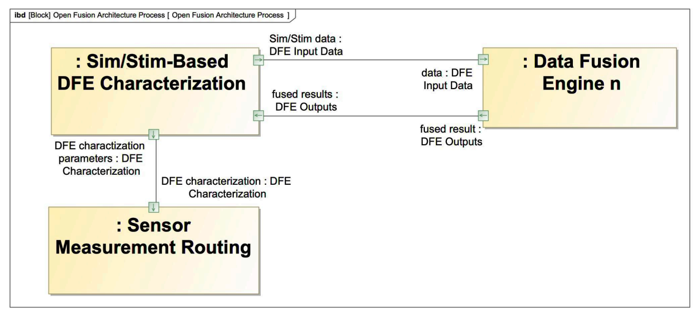

Figure 5: Open Architecture data fusion engine characterization

The characterization of a given fusion engine (labeled as the Data Fusion Engine (DFE) in Figure 5) is componentized across the architecture to automatically evaluate the performance of each one. This design has been incorporated in an existing data fusion test bed. The DFTB was designed for automated fusion engine evaluation across multiple programs.

5 Utilizing Fusion Engine Characterization for Multi-Engine Management

The mapping of each FE_i into the $MDSP_i$ can be used in multiple ways to construct the *Sensor Measurement Routing Service.* If the $MDSP_i$ is characterized as a collection of intervals in MDS, those intervals could be expressed in simple logic or fuzzy logic conditions that would be used to route the incoming sensor measurements to the appropriate FE_i. If the $MDSP_i$ is described as a continuous and differentiable bounding surface, a cross product in MDS would indicate which side of each bounding surface a sensor measurement lies. If the set of all $\{MDSP_i\}$ is captured in a deep learning network, querying the compressed parametric representation of that network with the sensor measurement would identify the appropriate FE_i.

The critical point is that each of these mathematical techniques is of low computational complexity, with a scaling constant that is proportional to the dimensionality of the MDS. These computationally inexpensive operations induce minimal overhead in the routing process, which implies that the overall computational complexity of the entire fusion system comprising the management framework and the constellation of fusion engines is bounded by the computational complexity of the individual FE_i, scaled by the relative frequency of sensor measurements occurring in the respective $MDSP_i$. For highly tuned tactical fusion engines, this computational complexity previously has been minimized to the maximum extent practical by source programs of record, so the net effect is that the combined fusion system captures those prior investments to deliver highly efficient sensor data fusion across $\left\{ \cup\ MDSP_i \right\}$ in exchange for the cost of FE_i characterization.

6 Future Implementation and Test of the Open Fusion Architecture

In the near future the authors plan on applying the methodology described in Section 4 above to characterize a few existing high-performance fusion engines. Depending on the results of that characterization, and in particular on the degree of overlap in the $MDSP_i$ of the individual FE_i, a *Sensor Measurement Routing Service* and a *Select Best Fused Track Estimate Service* will be implemented. These services will be combined with the FE_i as depicted in Figure 4, and the resulting data fusion ensemble will be integrated into a prototype JADC2 system currently under development.

This prototype JADC2 system will then be exercised on a suite of scenarios similar to those used to characterize the individual FE_i as well as with live and recorded sensor data collected during system of systems integration. The performance of the prototype JADC2 system will be measured and compared with the performance of the prototype JADC2 system prior to integration of the data fusion ensemble. The results of this effort will be published in an appropriate forum.

References

[1] Liggins II, M., D. Hall, and J. Llinas, eds., *Handbook of Multisensor Data Fusion, Theory and Practice*, 2nd ed. (multiple chapters), CRC Press, 2009.

[2] Schuck, T., J. B. Hunter, and D. Wilson, "Developing Information Fusion Methods for Combat Identification", in Liggins II, M., D. Hall, and J. Llinas, eds., *Handbook of Multisensor Data Fusion, Theory and Practice*, 2nd ed. CRC Press, 2009.

[3] Hall, D., and A. Steinberg, "Dirty Secrets in Multisensor Data Fusion"; Technical Report; Pennsylvania State University Applied Research Laboratory: University Park, PA, USA, 2001.

[4] Schuck, T., M. Friesel and J. B. Hunter, "Information Properties as a Means to Define Decision Fusion Methodologies in Non-Benign Environments," Sixth International Conference of Information Fusion, 2003. Proceedings of the, Cairns, QLD, Australia, 2003, pp. 479-484, https://doi: 10.1109/ICIF.2003.177485.

[5] Urken, A. B., B. Nimz, and T. Schuck, "Designing Evolvable Systems in a Framework of Robust, Resilient and Sustainable Engineering Analysis." *Advanced Engineering Informatics*, 26(3), 553–562, 2012.

[6] Ibid., Schuck, Hunter and Wilson (2009).

[7] Steinberg, A., C. L. Bowman, and F. E. White, "Revisions to the JDL model," Joint NATO/IRIS Conf., 1998.

[8] Pourghebleh, B., N. Hekmati, Z. Davoudnia, and M. Sadeghi, M. "A Roadmap Towards Energy-Efficient Data Fusion Methods in the Internet of Things. *Concurrency Comput. Pract. Exp*. 34 (15), e6959. 2022. https://doi:10.1002/cpe.6959.

[9] Dung, D., et al., "Overlapping Coalition Formation for Efficient Data Fusion in Multi-Sensor Networks", The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA.

[10] Senel, N., et al., "Multi-Sensor Data Fusion for Real-Time Multi-Object Tracking", Manufacturing Processes and Systems, a section of *Processes* 2023, 11(2), 501; https://doi.org/10.3390/pr11020501.

[11] Krishnamurthi, R., et al., "An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques", State of the Art Sensor Technologies a section in *Sensors* 2020, 20(21), 6076; https://doi.org/10.3390/s20216076.

[12] Ibid. Schuck, Friesel, and Hunter (2003).

[13] Dunn, S., J. Laroche, and P. Mitchell., "Open System Architectures for the ADF: Opportunities and Challenges." *Australian Defence Force Journal*, no. 204 (2018): 41–52.

[14] Votruba, P., R. Nisley, R. Rothrock, and B. Zombro., "Single Integrated Air Picture" System Engineering Task Force Arlington Virginia, Technical Report,

Approved for public release, October 29, 2001; https://apps.dtic.mil/sti/pdfs/ADA397225.pdf.