Reclassifying Fusion: An Empirically Grounded Alternative to the JDL Level Hierarchy

Greg Passmore

Abstract

The multisensor data fusion community has spent over three decades cycling through numbered "level" schemes, from the original JDL model (1985) through the Revised JDL (Steinberg, Bowman, White, 1999), the DFIG extensions (Blasch et al., 2002, 2005), Revision II (Llinas et al., 2004), and Steinberg's ontological refinement (2022, 2023), producing at least ten distinct naming conventions that remain in concurrent use with no formal deprecation of any predecessor. The result is a field where "Level 2" can mean situation assessment, feature fusion, or decision fusion depending on which scheme an author happens to adopt. Drawing on an empirical catalog of 1,647 fusion methods across four volumes, spanning nearly five decades of literature from the late 1970s through 2026 and thousands of primary references, I identify seven root causes of the naming confusion (pipeline implication, inconsistent counting, semantic overloading of level numbers, orthogonality of the Dasarathy input output model, informality of the low/medium/high scheme, version fragmentation across revisions, and conflation of estimation with control). I then propose a two axis EntityType.Operation naming convention grounded in what the catalog actually reveals: methods group naturally by the type of entity whose state they estimate or control (Signal, Object, Relational, Predictive, Control, Human) and by the mathematical operation they perform (Detection, Estimation, Classification, Combination, Optimization). The convention is descriptive rather than ordinal, self documenting, free of implied processing sequence, tolerant of methods that straddle traditional level boundaries, and directly mappable to real system architectures. A full crosswalk table maps the proposed convention back to JDL, DFIG, Dasarathy, and the three tier low/medium/high scheme, demonstrating that EntityType.Operation subsumes rather than replaces the prior work.

Keywords: multisensor data fusion, JDL model, DFIG, fusion levels, naming convention, taxonomy, sensor fusion, state estimation, Dasarathy model, information fusion, EntityType.Operation, data fusion architecture, georectification

1. The Problem with Naming Fusion Levels

I recently completed a project that consumed the better part of two years: a four volume compendium cataloging 1,647 multisensor data fusion methods, drawing on nearly five decades of literature from the late 1970s through 2026 and thousands of primary references across signal processing, estimation theory, control, machine learning, image fusion, military ISR, and dozens of adjacent fields. The set covers everything from classical Bayesian filters and JDL era architectures through modern deep learning fusion, organized by mathematical family and application domain. After reading, deriving, and writing up that many methods, I can report with confidence that the single most persistent source of confusion in the field is not the math, not the algorithms, not even the implementations. It is the naming. Everyone uses the word "level" and everyone means something different by it.

A researcher in image fusion says "low level fusion" and means pixel combination. A researcher in military command and control says "Level 2 fusion" and means situation assessment. A third researcher says "Level 2" and means feature fusion. All three believe they are speaking the same language. None of them are. The result is a field where every paper must begin with a paragraph explaining which naming convention the authors have adopted, and where readers must mentally translate between schemes on every page.

I have read more fusion papers than any reasonable person should. The naming confusion is not a minor annoyance; it is a structural impediment to the field. Methods that solve identical mathematical problems get classified into different "levels" depending on which scheme the author prefers. Methods that solve fundamentally different problems get lumped into the same "level" because the scheme is too coarse. And every attempt to fix the naming has produced yet another scheme, compounding the very problem it aimed to solve.

What follows is my attempt to lay out the history, identify the root causes, and propose a convention that I can actually live with after doing the work.

2. A History of Level Schemes

A. The Original JDL Model (1985, 1987, 1990)

The naming problem originates with the Joint Directors of Laboratories (JDL) Data Fusion Group. Established in 1986 under the US Department of Defense, the JDL Data Fusion Working Group set out to codify the terminology of a field that barely existed yet. The result was a process model that defined four fusion levels plus supporting infrastructure:

- **Level 1: Object Refinement.** Estimation and prediction of entity states on the basis of observation to track association, continuous state estimation (kinematic filtering), and discrete state estimation (classification, identification).
- **Level 2: Situation Refinement.** Estimation and prediction of relations among entities, including force structure, communications, and physical context.
- **Level 3: Threat Refinement.** Estimation and prediction of effects on situations of planned or estimated actions by participants, including threat intent and capability.
- **Level 4: Process Refinement.** Adaptive data acquisition and processing to support mission objectives, including sensor management.

The model also included source preprocessing (before the levels), a database management system, and a human computer interface, all connected via a bus architecture.

The framework appeared in its canonical form in Waltz and Llinas (1990), *Multisensor Data Fusion* (Artech House), the book that defined the field for a generation. Hall and Llinas (1997) subsequently introduced the model to a broader audience through their *Proceedings of the IEEE* survey, "An Introduction to Multisensor Data Fusion."

The original JDL model had two significant problems. First, it contained no explicit signal level processing stage. Everything below object refinement was lumped into "source preprocessing," which gave no guidance on how to categorize the vast body of signal processing methods (beamforming, spectral estimation, CFAR detection) that constitute the front end of any real fusion system. Second, the terminology was military specific. "Threat Refinement" made perfect sense in a DoD context but was meaningless for medical imaging, autonomous vehicles, or environmental monitoring.

Despite these limitations, the JDL model succeeded in establishing the basic vocabulary. For better or worse, the field learned to think in numbered levels.

B. The Revised JDL Model (1998, 1999, 2001): Steinberg, Bowman, and White

The first major revision came from Alan Steinberg, Christopher Bowman, and Franklin White. Their work appeared initially at SPIE in 1999 (Proceedings of SPIE Vol. 3719) and was then published as Chapter 2 of the *Handbook of Multisensor Data Fusion* (CRC Press, 2001).

The revision made several important changes:

- **Added Level 0: Sub Object Assessment.** Signal and pixel level processing finally had a home. Level 0 covered estimation and prediction of signal and object observable states on the basis of pixel and signal level data association. CFAR detection, beamforming, image preprocessing, and spectral estimation all fell here.

- **Renamed Level 3** from "Threat Refinement" to **"Impact Assessment."** The change broadened the scope beyond military threat analysis to encompass any estimation of effects on situations from planned or estimated actions.
- **Redefined data fusion itself** as a state estimation problem. Fusion became, in Steinberg and Bowman's formulation, the process of combining data to estimate entity states, where an entity could be any aspect of a universe of discourse at any degree of abstraction.

The state estimation framing was the revision's most lasting intellectual contribution. It gave the field a unified mathematical lens: every fusion problem, at every "level," is fundamentally about estimating the state of some entity given uncertain observations. The question is merely what kind of entity (signal, object, situation, impact, resource) and what kind of state variables are involved.

Steinberg and Bowman also introduced a useful structural distinction: **attribution based fusion** (Levels 0 and 1, dealing with properties of individual entities) versus **relation based fusion** (Levels 2 and 3, dealing with relationships among entities). The mathematics changes when you cross that boundary. Attribution based fusion typically involves state spaces that are continuous and comparatively low dimensional. Relation based fusion involves combinatorial state spaces (which entities are related, and how?) and demands fundamentally different inference machinery.

C. The DFIG Model (2002, 2005, 2006): Blasch and Colleagues

Erik Blasch and Susan Plano proposed Level 5, "User Refinement," in their 2002 SPIE paper (SPIE Vol. 4729, Aerosense 2002). The motivation was straightforward: the JDL model, even in its revised form, treated the human operator as external to the fusion process. The human computer interface sat outside the level hierarchy. Blasch argued that the human's cognitive processing of fused information (trust calibration, attention allocation, workload management, situation awareness construction) was itself a form of fusion that deserved explicit representation.

The ISIF Data Fusion Information Group (DFIG) subsequently adopted this extension and added Level 6, "Mission Management," producing a seven level model (Levels 0 through 6). The DFIG model also introduced the distinction between **Low Level Information Fusion (LLIF)**, covering Levels 0 and 1, and **High Level Information Fusion (HLIF)**, covering Levels 2 through 6. The model was mapped explicitly to Boyd's Observe Orient Decide Act (OODA) loop, giving it a decision theoretic grounding that the original JDL model lacked.

The DFIG model appeared across multiple SPIE and FUSION conference publications between 2002 and 2010, with Blasch, Steinberg, Das, Llinas, Chong, Kessler, Waltz, and White all contributing to various refinements.

Adding Levels 5 and 6 solved a real problem (the human was indeed missing from the model) but created a new one: the level count was now seven, the numbering was becoming unwieldy, and the boundary between Level 4 (Process Refinement) and Level 6 (Mission Management) was not always clear. More fundamentally, the model was drifting from a fusion taxonomy into a systems engineering framework. Mission Management is not fusion in the state estimation sense; it is control. The conflation of estimation and control within a single numbered hierarchy would become a recurring source of confusion.

D. JDL Revision II (2004): Llinas, Bowman, Rogova, Steinberg, Waltz, and White

The 2004 FUSION conference saw a second major revision by the original architects: James Llinas, Christopher Bowman, Galina Rogova, Alan Steinberg, Edward Waltz, and Franklin White. Published as "Revisiting the JDL Data Fusion Model II" in the FUSION 2004 Proceedings, the paper addressed the internal structure of fusion processing within each level.

Key contributions included:

- **Elaborated Level 2 and Level 3 internal structure.** The authors described how situation and impact assessment require co processing of abductive (hypothesis generating) and deductive (hypothesis testing)

inference. A fusion node does not simply run a pipeline; it iterates between generating explanations for observed data and testing those explanations against further evidence.
- **Reliability weighted fusion.** The revision emphasized that fusion must account for the reliability and relevance of each source, not just its information content. A highly precise but unreliable source can be worse than a noisy but reliable one.
- **Ontology based partitioning.** The paper began moving toward partitioning fusion by entity type rather than by sequential level number. Objects, situations, threats, and processes differ not in their position in a pipeline but in the type of state variables they involve.

Revision II was more a deepening than a restructuring. The level numbers remained. But the intellectual direction (toward ontological partitioning, away from sequential pipeline thinking) planted the seeds for later developments.

E. Dasarathy's Input Output Model (1997)

Belur Dasarathy took an entirely different approach. His 1997 paper, "Sensor Fusion Potential Exploitation: Innovative Architectures and Illustrative Applications" (Proceedings of the IEEE, Vol. 85), characterized fusion not by what level of abstraction it operates at, but by what goes in and what comes out.

Dasarathy defined five categories based on input output type:

Category	Input	Output	Example
DAI DAO	Data	Data	Pixel level image fusion, signal combination
DAI FEO	Data	Features	Feature extraction from raw sensor data
FEI FEO	Features	Features	Feature refinement, feature level combination
FEI DEO	Features	Decisions	Classification from fused feature vectors
DEI DEO	Decisions	Decisions	Voting, Bayesian decision combination

The scheme is orthogonal to JDL. It does not ask "what level of entity are you estimating?" but rather "what representation does your method consume and what representation does it produce?" A single system might contain DAI DAO processing at its front end, DAI FEO processing in its middle, and DEI DEO processing at its output, regardless of whether any of those stages correspond to JDL Level 0, 1, 2, or 3.

Dasarathy's model has three virtues that the JDL model lacks. First, it is not hierarchical: there is no implication that DAI DAO must precede FEI DEO. Second, it is descriptive rather than prescriptive: it tells you what a method does rather than where it belongs in an abstract taxonomy. Third, it is agnostic to application domain: the categories apply equally to military ISR, medical imaging, autonomous driving, and environmental monitoring.

The limitation is granularity. Five categories is not enough to distinguish, say, Kalman filtering (which is arguably DAI FEO or FEI FEO depending on how you define "feature") from particle filtering (same ambiguity) from graphical model inference (also ambiguous). The scheme tells you the input output type but nothing about the entity being estimated or the mathematical operation being performed.

F. The Low, Medium, and High Abstraction Model

The image fusion community, largely independently of the JDL tradition, developed its own three level scheme:
- **Low level fusion** (also called data level, signal level, or pixel level fusion): combination of raw sensor data before any feature extraction. Examples include Laplacian pyramid fusion, wavelet based image fusion, and averaging of registered images.

- **Medium level fusion** (also called feature level fusion): combination of features extracted from individual sensors. Examples include concatenation of feature vectors from different modalities, followed by joint classification.
- **High level fusion** (also called decision level fusion): combination of decisions or classifications made independently by each sensor's processing chain. Examples include voting, Dempster Shafer combination of evidence, and Bayesian decision fusion.

The scheme is intuitive and easy to teach. It also maps roughly onto the first three categories of the Dasarathy model (DAI DAO $\approx$ low, FEI FEO $\approx$ medium, DEI DEO $\approx$ high). But "roughly" is the operative word. The boundaries are imprecise. Is a method that takes raw pixels and produces a fused feature map "low level" or "medium level"? The answer depends on whether you define the level by the input, the output, or the internal representation.

More critically, the three level scheme covers only what the JDL model would call Levels 0 and 1 (and perhaps the front end of Level 2). It has nothing to say about situation assessment, impact prediction, resource management, or human interaction. For the image fusion community, this omission is unproblematic; for the broader data fusion community, it renders the scheme incomplete.

The collision between these two uses of the word "level" is a major source of cross community confusion. When an image fusion researcher says "Level 2 fusion," they typically mean feature level combination (medium level in the three tier scheme). When a JDL trained researcher says "Level 2 fusion," they mean situation assessment. The same two words, completely different meanings.

G. Steinberg's Ontological Refinement (2022, 2023)

Alan Steinberg returned to the problem in his 2022 FUSION conference paper, "An Ontology for Multi Level Data Fusion" (FUSION 2022, Linköping, Sweden), and his subsequent 2023 article in the ISIF *Perspectives on Information Fusion* (IPIF). Steinberg argued that the real organizing principle behind the JDL levels was always entity type, not processing sequence:

- **Levels 0 through 2** can be distinguished by the types of state variables being estimated: signal and feature parameters versus individual metric and kinematic variables versus relational variables.
- **Level 3** estimates or predicts courses of action, events, and impacts. Because these generally concern projected entity states and relationships, many versions of fusion models refer to a blended "Level 2/3."
- **Level 4** (Process Refinement) and the proposed Levels 5 and 6 relate to resource management rather than estimation proper.

Steinberg made the explicit argument that the word "levels" is misleading because it implies a sequential pipeline: first do Level 0, then Level 1, then Level 2, and so on. The original JDL architects never intended this reading (the bus architecture was supposed to allow arbitrary inter level communication), but the numbered hierarchy inevitably suggests sequence to anyone encountering it for the first time. Steinberg proposed partitioning by entity type (signal, object, situation, scenario) as a more accurate reflection of how fusion systems actually work.

The 2023 IPIF article also drew a sharp distinction between **data fusion** (estimation) and **resource management** (control). Steinberg argued that lumping both into a single numbered hierarchy had been a persistent source of confusion: Level 4 Process Refinement and Level 6 Mission Management are not fusion problems in the same sense that Level 1 Object Assessment is a fusion problem. They are control problems that use fusion outputs.

H. Other Schemes

Several additional models deserve mention for completeness:

- **The Omnibus Model (Bedworth and O'Brien, 1999).** An activity based, cyclic model that described data fusion as a four stage loop: sensing, signal processing, situation assessment, and decision making. Unlike the JDL model, the Omnibus model specified process sequence and emphasized the cyclic nature of operations. It employed general (non military) terminology but did not support easy decomposition into independently testable modules.

- **The Salerno SA Model.** John Salerno proposed a situational awareness reference model that combined the JDL Data Fusion model with Endsley's three level SA model (Perception, Comprehension, Projection). The result mapped fusion levels onto cognitive SA levels, adding a "Resolution" level for determining actions.
- **Boyd OODA Derived Models.** John Boyd's Observe Orient Decide Act loop influenced multiple fusion models, most notably the DFIG model. The OODA mapping was attractive because it connected fusion (Observe, Orient) to decision making (Decide, Act), but it also blurred the line between estimation and control.
- **The Luo Kay Taxonomy.** Referenced as Method 1470 in the compendium's military and defense fusion section (Book 4, Section 2), Luo and Kay proposed a taxonomy that organized sensor fusion by the functional role of each processing stage within a robotic system. Their scheme distinguished sensor modeling, signal level fusion, feature level fusion, and symbol level fusion, with explicit attention to the physical configuration of the sensor suite.

3. Comparison of Naming Schemes

Scheme	Year(s)	Levels or Categories	Partitioning Criterion	Strengths	Weaknesses
Original JDL	1985, 1987, 1990	4 (L1 through L4) plus preprocessing	Abstraction of entity under refinement	First standard; widely adopted; clear hierarchy	No signal level; military terminology; implies sequence
Revised JDL (Steinberg, Bowman, White)	1998, 1999, 2001	5 (L0 through L4)	Entity state type (attribution vs relation)	Added signal level; state estimation framing; broadened terminology	Still implies pipeline; still military influenced; Level 4 mixes estimation and control
DFIG (Blasch et al.)	2002, 2005, 2006	7 (L0 through L6)	Entity type plus human and mission management	Includes human; maps to OODA; comprehensive	Seven levels is unwieldy; blurs fusion and control; Level 4/5/6 boundaries unclear
JDL Revision II (Llinas et al.)	2004	5 (L0 through L4), elaborated internals	Entity type with inference co processing	Deepened L2/L3; reliability weighting; ontological direction	Did not resolve level count; added complexity without simplifying naming
Dasarathy I/O Model	1997	5 (DAI DAO, DAI FEO, FEI FEO, FEI DEO, DEI DEO)	Input and output representation type	Orthogonal to JDL; descriptive; non hierarchical; domain agnostic	Too coarse; ambiguous for many methods; ignores entity semantics

Low/Medium/ High	Various	3 (low, medium, high)	Abstraction of representation	Simple; intuitive; easy to teach	Imprecise boundaries; covers only signal/object scope; collides with JDL numbering
Steinberg Ontological (2022, 2023)	2022, 2023	Partitioned by entity type	Ontological: signal, object, relation, scenario	Principled; avoids pipeline implication; separates estimation from control	Not yet widely adopted; does not prescribe operations
Omnibus (Bedworth, O'Brien)	1999	4 cyclic stages	Functional activity sequence	Cyclic; general terminology; process oriented	Does not support modular decomposition; less influential
Salerno SA Model	2008	4 SA levels mapped to fusion	Cognitive SA (perception, comprehension, projection, resolution)	Links fusion to human cognition; multi domain	Heavily tied to SA framework; less useful for signal level
Luo Kay Taxonomy	1990s	4 (sensor modeling, signal, feature, symbol)	Functional role in robotic system	Grounded in physical sensor configuration; practical	Robotic scope; limited adoption outside that community

4. The Core Confusion

The naming mess has identifiable root causes. Let me enumerate them.

"Levels" implies sequence. The most damaging consequence of numbering fusion categories 0, 1, 2, 3, 4 is that every new reader interprets the numbers as steps. First you do Level 0 signal processing, then you pass the output to Level 1 object estimation, then the object estimates flow to Level 2 situation assessment, and so on. The JDL architects explicitly stated that the bus architecture was meant to allow arbitrary inter level communication, that processing need not follow the numerical order. But a bus architecture diagram does not survive contact with a numbered list. People read "Level 0, Level 1, Level 2" and think "step one, step two, step three." Thirty five years of published diagrams showing left to right arrows from L0 to L4 have cemented the pipeline interpretation.

Different schemes count differently. The original JDL model has four levels (1 through 4). The revised JDL has five (0 through 4). The DFIG has seven (0 through 6). The low/medium/high scheme has three. Dasarathy has five categories that are not levels at all. Saying "there are N levels of fusion" is meaningless without specifying which scheme you are using, and most authors do not specify.

The same number means different things. "Level 2" in the JDL tradition means situation assessment: estimating relationships among entities, assessing force structures, understanding group behaviors. "Level 2" in the image fusion community (mapping from the three tier low/medium/high scheme) typically means feature level fusion: combining feature vectors extracted from different sensors. A paper that says "we propose a Level 2 fusion method" could be describing a Bayesian network for situation assessment or a neural network for fusing image features. The reader has no way to know without further context.

Dasarathy is orthogonal, not hierarchical. Dasarathy's input output model classifies methods by representation type, not by abstraction level. A DAI DAO method and a DEI DEO method might both operate within JDL Level 1 (one fusing raw radar returns, the other combining track classifications). Dasarathy and JDL are measuring different things. Trying to map one onto the other produces at best an approximate correspondence and at worst outright confusion.

The low/medium/high scheme is informal. It has no canonical reference, no formal definition of where "low" ends and "medium" begins. Different authors draw the boundaries differently. Some equate "low level" with pixel level and "medium level" with feature level. Others use "low level" to encompass everything up to and including feature extraction. The scheme survives because it is easy to say in conversation, not because it is precise.

Each revision created version fragmentation. Every attempt to improve the naming (adding Level 0, adding Level 5, adding Level 6, elaborating Level 2/3 internals, proposing ontological partitioning) produced a new variant that coexisted with the old ones. The literature now contains papers using the original JDL model, the 1999 revision, the 2001 handbook version, the DFIG version, Revision II, and various hybrids. No version has been formally deprecated. All of them are cited. The result is not one naming scheme with known limitations but half a dozen naming schemes with overlapping scope and inconsistent terminology.

Estimation and control are conflated. The original Level 4 (Process Refinement) and the DFIG additions (Level 5 User Refinement, Level 6 Mission Management) are not estimation problems. They are control problems. Sensor management decides where to point the radar, not what state the target is in. Mission planning decides what objectives to pursue, not what situation currently exists. As Steinberg argued in 2023, lumping estimation and control into a single numbered hierarchy conflates two fundamentally different kinds of computational problems. An extended Kalman filter (estimation) and a partially observable Markov decision process (control) are not different "levels" of the same activity; they are different activities entirely.

Georectification is invisible. Every real fusion system depends on georectification: the process of projecting sensor measurements into a common earth referenced coordinate frame so that observations from different platforms, different sensors, and different modalities can be compared, associated, and combined. Without georectification, there is no fusion. An infrared image from a UAV at 15,000 feet and a SAR image from a satellite at 400 kilometers cannot be overlaid until both have been mapped to the same geographic grid. A radar detection in range, azimuth, and elevation cannot be correlated with an ESM bearing until both have been transformed into a shared geodetic frame. Yet no numbered level scheme gives georectification a home. It is not Level 0 (it operates on metadata and platform state, not on signal content). It is not Level 1 (it is a prerequisite for object estimation, not object estimation itself). It is not sensor management (Level 4), not situation assessment (Level 2), and not user refinement (Level 5). Georectification is infrastructure that every level depends on and no level owns. The numbered schemes simply have no place for it, which is remarkable given that in operational ISR systems, georectification errors are often the dominant source of fusion failure.

5. What the Compendium Reveals

After cataloging 1,647 methods across four volumes (564 in Book 1, 531 in Book 2, 341 in Book 3, and 211 in Book 4), I have the advantage of empirical observation. I did not set out to disprove any naming scheme. I set out to catalog every significant fusion method I could find. The naming problem revealed itself through the data.

Methods naturally group by the type of entity whose state they estimate. Kalman filters, particle filters, and data association algorithms cluster together because they all estimate the state of individual objects (position, velocity, identity). Bayesian networks, factor graphs, and situation templates cluster together because they all estimate relationships and group behaviors. CFAR detectors, beamformers, and spectral estimators cluster together because they all operate on raw signals. The entity type (signal, object, relation, scenario, resource) is the most natural organizing dimension.

Methods also group by the type of operation they perform. Within any entity type, you find detection methods (is something there?), estimation methods (where and what is it?), classification methods (what type is it?), combination methods (how do I merge multiple answers?), and optimization methods (how do I improve the process?). A CFAR detector and a track detection algorithm both perform detection, but on different entity types (signals and objects, respectively). A Kalman filter and a Bayesian network both perform estimation, but on different entity types (objects and relations, respectively).

The operation by entity matrix is more useful than any single "level" hierarchy. If I place entity types on one axis and operation types on the other, I get a two dimensional grid that accommodates every method in the compendium. CFAR detection is Signal × Detection. Kalman filtering is Object × Estimation. Bayesian situation assessment is Relational × Classification. Game theoretic threat prediction is Predictive × Estimation. POMDP sensor management is Control × Optimization. No single "level" number captures these distinctions.

Many methods straddle multiple "levels." A particle filter does not care whether it is estimating an object state (JDL Level 1) or a situation state (JDL Level 2). The mathematics is the same: a sequential Monte Carlo approximation to a Bayesian posterior. The particle filter is a method; "Level 1" and "Level 2" are labels that depend on what state vector you plug into it. Similarly, a factor graph can perform object level data association (Level 1), situation assessment (Level 2), or distributed detection (Level 0), depending on how you define the graph's variables and factors. The method is invariant to the level; the level is determined by the application context.

The 31 sections of Book 4 alone span every supposed "level." Book 4 (Methods 1437 through 1647) covers 31 sections ranging from Cognitive and Biological Fusion Models (Section 1) through CFAR Variants and Adaptive Detection (Section 8) through Graphical Models and Probabilistic Inference (Section 28) through Coordinate Systems and Measurement Models (Section 31). These 211 methods span what any naming scheme would call Levels 0 through 5. Yet the methods themselves do not organize neatly into a level hierarchy. A copula based dependence model (Section 14) can fuse signals, features, or decisions depending on the application. A convex optimization method (Section 24) can optimize a sensor management problem (Level 4), a detection threshold (Level 0), or a classifier ensemble (Level 1). The method does not know what level it belongs to.

Critical prerequisites have no level. Georectification is the clearest example. Before any fusion can occur, sensor measurements must be projected into a common earth referenced coordinate frame. The compendium addresses this through Coordinate Systems and Measurement Models (Section 31, Methods 1646 through 1647) and Robust Registration and Model Fitting (Section 26, Methods 1622 through 1625), but these methods do not belong to any single JDL level. A georectification pipeline transforms raw sensor geometry into earth coordinates (touching Level 0), enables track correlation across platforms (enabling Level 1), and constrains situation assessment by defining spatial relationships (feeding Level 2). In practice, georectification is the single largest engineering effort in many operational fusion systems, consuming more development time than the tracking filters and classifiers combined, yet the level schemes treat it as invisible infrastructure.

The compendium's structure reflects entity type and operation, not levels. When I organized Book 4, the 31 sections emerged naturally from the methods themselves: sections on detection (CFAR, sequential detection, distributed detection), sections on estimation (filtering, smoothing, interpolation), sections on combination (classifier ensembles, aggregation operators), sections on signal processing (beamforming, spectral estimation, pyramid fusion, wavelet fusion), sections on optimization (convex, metaheuristic), sections on uncertainty representation, sections on registration and coordinate geometry, sections on inference (graphical models). These sections map to entity types and operation types, not to JDL levels.

6. A Proposed Convention

I propose a naming convention grounded in what the 1,647 method catalog actually reveals. The convention has two orthogonal axes.

Axis 1: Entity Type (What Is Being Estimated or Controlled?)

Following Steinberg's ontological direction but grounding it in empirical observation from the compendium:

Signal Fusion. The entity is a raw signal, pixel, waveform, or spectral component. The state variables are signal parameters: amplitude, phase, frequency, spatial location within a sensor array, pixel intensity across registered images. Methods include CFAR detection (Methods 1520 through 1525), beamforming and array processing (Methods 1526 through 1530), spectral estimation (Methods 1531 through 1537), pyramid decomposition (Methods 1576 through 1579), wavelet and directional transform fusion (Methods 1580 through 1584), sparse representation fusion (Methods 1585 through 1588), and edge preserving filter fusion (Methods 1589 through 1591). In JDL terms, this is Level 0.

Object Fusion. The entity is an individual object (target, platform, emitter, person). The state variables are kinematic (position, velocity, acceleration), attributive (type, class, identity), and existential (does this track correspond to a real entity?). Methods include Kalman filtering and its extensions (Book 1 and Methods 1494 through 1510), particle filtering (Book 1), data association (Methods 1471 through 1481), track management and scoring (Methods 1482 through 1493), out of sequence processing (Methods 1511 through 1514), and distributed and federated estimation (Methods 1515 through 1519). In JDL terms, this is Level 1.

Relational Fusion. The entity is a relationship, group, structure, or situation. The state variables describe how objects are related: spatial proximity, communication links, command hierarchies, behavioral patterns, force structures. Methods include graphical models and probabilistic inference (Methods 1630 through 1635), copula based dependence modeling (Methods 1557 through 1560), information theoretic methods applied to relational structures (Methods 1592 through 1600), and the many Bayesian network and factor graph methods cataloged in Books 1 through 3. In JDL terms, this is Level 2.

Predictive Fusion. The entity is a future state, intent, plan, course of action, or impact. The state variables describe what will happen or what an adversary intends to do. Methods include game theoretic models (Method 1611 and Book 2), simulation based approaches, threat assessment frameworks from the military and defense section (Methods 1450 through 1470), and predictive inference methods. In JDL terms, this is Level 3.

Control Fusion. The entity is a resource, sensor, or process whose allocation or configuration must be optimized. The state variables describe the controllable parameters of the fusion system itself: where to point sensors, how to allocate bandwidth, which processing chains to activate. Methods include POMDP formulations (Book 2), reinforcement learning for sensor management (Book 2), convex optimization for resource allocation (Methods 1612 through 1618), metaheuristic optimization (Methods 1619 through 1621), and control theoretic estimation methods (Methods 1636 through 1641). In JDL terms, this maps to Level 4 and DFIG Level 6.

Human Fusion. The entity is the human operator's cognitive state: attention, workload, trust, situation awareness, and information need. The state variables describe what the user knows, what the user needs to know, and how the system should present information to optimize cognitive performance. Methods include cognitive and biological fusion models (Methods 1437 through 1449), attention models, Shapley value attribution for explainability (Book 2), and visualization optimization. In DFIG terms, this is Level 5. The original JDL model had no corresponding level.

Axis 2: Operation Type (What Mathematical Operation Is Being Performed?)

Following Dasarathy's insight that the type of processing matters independently of the level, but with finer granularity:

Detection. Is something there? Binary or M-ary hypothesis testing, with the goal of determining whether an entity of interest exists in the data. Includes CFAR detection (Signal.Detection), track initiation (Object.Detection), anomaly detection in relational structures (Relational.Detection), and change detection in situational patterns (Predictive.Detection).

Estimation. Where is it, and what are its state variables? Continuous state estimation using Bayesian filtering, least squares, maximum likelihood, or their approximations. Includes beamforming for angle of arrival (Signal.Estimation), Kalman filtering for kinematic states (Object.Estimation), Bayesian network inference for relational states (Relational.Estimation), and predictive simulation (Predictive.Estimation).

Classification. What type is it? Discrete state estimation, assigning an entity to one of a finite set of categories. Includes modulation classification from spectral features (Signal.Classification), target identification from fused attributes (Object.Classification), situation type recognition (Relational.Classification), and intent classification (Predictive.Classification).

Combination. How do I merge multiple estimates, features, or decisions into one? Explicit fusion of parallel information streams. Includes pixel averaging and wavelet coefficient selection (Signal.Combination), track to track fusion and covariance intersection (Object.Combination), evidence combination via Dempster Shafer or classifier ensembles (cross cutting, applicable at multiple entity levels), and decision voting (applicable wherever decisions are made).

Optimization. How do I improve the process? Tuning parameters, allocating resources, selecting configurations. Includes adaptive threshold optimization (Signal.Optimization), sensor management (Control.Optimization), mission planning (Control.Optimization), and cognitive load management (Human.Optimization).

The Naming Convention

I propose referring to methods by **EntityType.Operation** notation. Examples:

Method	Conventional Name	Proposed Convention
Cell Averaging CFAR	Level 0 detection	Signal.Detection
Capon Beamformer	Level 0 estimation	Signal.Estimation
Laplacian Pyramid Fusion	Low level image fusion	Signal.Combination
Extended Kalman Filter	Level 1 tracking	Object.Estimation
Global Nearest Neighbor	Level 1 data association	Object.Detection
Random Forest target classification	Level 1 classification	Object.Classification
Covariance Intersection	Level 1 track fusion	Object.Combination
Bayesian Network situation assessment	Level 2 fusion	Relational.Estimation
Factor Graph group detection	Level 2 fusion	Relational.Detection
Game theoretic threat assessment	Level 3 fusion	Predictive.Estimation
Course of action classification	Level 3 fusion	Predictive.Classification
POMDP sensor management	Level 4 fusion	Control.Optimization
Reinforcement learning for tasking	Level 4/6 fusion	Control.Optimization
Shapley attribution for explainability	Level 5 user support	Human.Optimization
Attention model for display design	Level 5 user refinement	Human.Estimation

Mapping Compendium Sections to the Convention

The 31 sections of Book 4 map as follows:

Section	Methods	Primary Convention
1. Cognitive and Biological Fusion Models	1437 through 1449	Human.Estimation, Human.Combination
2. Military and Defense Fusion Methods	1450 through 1470	Predictive.Estimation, Relational.Estimation, Control.Optimization
3. Advanced Data Association	1471 through 1481	Object.Detection
4. Track Management and Scoring	1482 through 1493	Object.Estimation, Object.Detection
5. Advanced Filtering and Smoothing	1494 through 1510	Object.Estimation
6. Out of Sequence Processing	1511 through 1514	Object.Estimation
7. Distributed and Federated Estimation	1515 through 1519	Object.Combination
8. CFAR Variants and Adaptive Detection	1520 through 1525	Signal.Detection
9. Beamforming and Array Processing	1526 through 1530	Signal.Estimation
10. Spectral Estimation Methods	1531 through 1537	Signal.Estimation
11. Classifier Combination and Ensemble Methods	1538 through 1546	Object.Combination, Relational.Combination
12. Robust and Nonparametric Detection	1547 through 1551	Signal.Detection, Object.Detection
13. Sequential Detection and Change Detection	1552 through 1556	Signal.Detection, Relational.Detection
14. Copula Based Dependence Modeling	1557 through 1560	Relational.Estimation, Signal.Combination
15. Distributed Detection Theory	1561 through 1575	Signal.Detection, Object.Detection
16. Pyramid Based Image Fusion	1576 through 1579	Signal.Combination
17. Wavelet and Directional Transform Fusion	1580 through 1584	Signal.Combination
18. Sparse Representation and Matrix Decomposition Fusion	1585 through 1588	Signal.Combination
19. Edge Preserving Filter Fusion	1589 through 1591	Signal.Combination
20. Information Theoretic Fusion Methods	1592 through 1600	Signal.Estimation, Object.Estimation, Relational.Estimation
21. Distributed Source Coding for Fusion	1601 through 1605	Signal.Combination
22. Advanced Uncertainty Representation	1606 through 1609	Cross cutting (applicable to all entity types)
23. Aggregation Operators and Game Theoretic Fusion	1610 through 1611	Relational.Combination, Predictive.Estimation
24. Convex Optimization for Fusion	1612 through 1618	Control.Optimization, Signal.Optimization
25. Metaheuristic Optimization	1619 through 1621	Control.Optimization
26. Robust Registration and Model Fitting	1622 through 1625	Signal.Estimation, Object.Estimation
27. Robust Statistics for Fusion	1626 through 1629	Cross cutting (applicable to all entity types)
28. Graphical Models and Probabilistic Inference	1630 through 1635	Relational.Estimation, Object.Estimation
29. Control Theoretic Estimation	1636 through 1641	Control.Optimization, Object.Estimation

30. Spatial Interpolation and Regression	1642 through 1645	Signal.Estimation, Object.Estimation
31. Coordinate Systems and Measurement Models	1646 through 1647	Cross cutting (infrastructure)

Notice that several sections span multiple entity types and operations. Sections 22 (Uncertainty Representation), 26 (Robust Registration and Model Fitting), 27 (Robust Statistics), and 31 (Coordinate Systems and Measurement Models) are genuinely cross cutting: they provide mathematical infrastructure used by methods at every entity level and for every operation type. Georectification lives here, in the cross cutting infrastructure layer, where it belongs. A georectification pipeline that transforms sensor measurements into WGS 84 coordinates is not Signal Fusion (it does not process signal content), not Object Fusion (it does not estimate object states), and not Control Fusion (it does not manage resources). It is coordinate infrastructure that every entity type and every operation type depends on. The EntityType.Operation convention accommodates this by labeling the application context: the same projective geometry used for image registration is Signal.Estimation; the same geodetic transform used for multi platform track alignment is Object.Estimation; the underlying coordinate mathematics is cross cutting infrastructure. The convention handles this gracefully by allowing a method to carry multiple labels. An M estimator (Section 27) used for robust track estimation is Object.Estimation; the same M estimator used for robust signal processing is Signal.Estimation. The method is the same; the label describes the application context.

Why EntityType.Operation Is Better Than Numbered Levels

Five reasons.

It is descriptive. "Signal.Detection" tells you what the method does. "Level 0" tells you where someone put it in a taxonomy. A researcher encountering "Object.Estimation" for the first time can guess, correctly, that the method estimates the state of an individual object. A researcher encountering "Level 1" for the first time has to look up which scheme is being used.

It is self documenting. The convention serves as its own legend. There is no need for a footnote explaining "we use the revised JDL model with the DFIG Level 5 extension but without Level 6." The label itself contains its meaning.

It does not imply sequence. Signal.Detection and Relational.Estimation have no inherent ordering. A system might perform Relational.Estimation (situation assessment) first to generate context, then use that context to improve Signal.Detection (adaptive CFAR with situation dependent thresholds). The convention permits this. Numbered levels discourage it.

It accommodates methods that straddle categories. A particle filter used for target tracking is Object.Estimation. The same particle filter used for situation state estimation is Relational.Estimation. A Dempster Shafer combination rule used for target classification is Object.Combination. The same rule used for situation classification is Relational.Combination. The method gets the label that matches its application context, rather than being forced into a single "level" that may not fit.

It maps naturally to system architecture. A real fusion system has signal processing modules, object tracking modules, situation assessment modules, resource management modules, and display modules. Each module performs specific operations on specific entity types. The EntityType.Operation convention maps one to one onto this architecture. Numbered levels do too, approximately, but the convention makes the mapping explicit and precise.

7. Mapping to Existing Schemes

For researchers accustomed to existing naming conventions, the following table provides a crosswalk:

Proposed Convention	JDL Revised (Steinberg, Bowman, White)	DFIG (Blasch et al.)	Dasarathy (Approximate)	Low/Med/High
Signal Fusion	Level 0 (Sub Object Assessment)	Level 0	DAI DAO	Low
Object Fusion	Level 1 (Object Assessment)	Level 1	DAI FEO, FEI FEO	Low to Medium
Relational Fusion	Level 2 (Situation Assessment)	Level 2	FEI DEO	High
Predictive Fusion	Level 3 (Impact Assessment)	Level 3	DEI DEO	High
Control Fusion	Level 4 (Process Refinement)	Level 4, Level 6	(not covered)	(not covered)
Human Fusion	(not covered)	Level 5	(not covered)	(not covered)

The mapping is approximate by necessity. The Dasarathy categories are defined by input output type, not entity type, so the correspondence is loose. DAI FEO (data in, feature out) could be Signal Fusion (feature extraction from raw signals) or Object Fusion (feature extraction from preprocessed data); the Dasarathy label does not distinguish between these. Similarly, "High" in the three tier scheme encompasses both Relational and Predictive Fusion with no differentiation between them.

The Dasarathy input output axis and the Operation axis of the proposed convention have a relationship worth noting:

Operation	Most Common Dasarathy Mapping
Detection	DAI DAO (raw data in, detection decision out, though Dasarathy would call this DAI DEO)
Estimation	DAI FEO or FEI FEO (data or features in, refined features or states out)
Classification	FEI DEO (features in, class decision out)
Combination	Varies: DAI DAO for signal combination, FEI FEO for feature combination, DEI DEO for decision combination
Optimization	(not directly covered by Dasarathy)

The two axis convention subsumes, rather than replaces, the prior schemes. If you prefer JDL numbering, read the entity axis (Signal = L0, Object = L1, Relational = L2, Predictive = L3, Control = L4). If you prefer Dasarathy, read the operation axis through the lens of input output types. If you prefer low/medium/high, Signal is low, Object is medium, Relational and above are high. The convention does not invalidate any prior work; it provides a common framework into which all prior schemes can be translated.

8. Closing

I do not claim this convention is the final word. Naming conventions are tools, and tools get replaced when better ones appear. What I claim is that EntityType.Operation is the convention I can actually live with after reading, cataloging, and analyzing 1,647 methods across 1.19 million words and roughly 5,300 pages of multisensor data fusion literature.

The convention describes what the methods do rather than assigning arbitrary numbers. It accommodates every method in the compendium, including the ones that straddle multiple JDL levels and the ones that JDL never covered at all. It separates estimation from control, which is a fundamental distinction that the numbered schemes persistently blur. And it avoids the single worst failure of every prior naming scheme: it does not pretend that fusion happens in a sequential pipeline from Level 0 to Level N.

The field has spent thirty five years arguing about what the levels mean. Perhaps it is time to stop numbering them and start naming them instead.

References

1. Waltz, E., and Llinas, J. *Multisensor Data Fusion.* Norwood, MA: Artech House, 1990.
2. Hall, D. L., and Llinas, J. "An Introduction to Multisensor Data Fusion." *Proceedings of the IEEE*, Vol. 85, No. 1, pp. 6–23, January 1997.
3. Dasarathy, B. V. "Sensor Fusion Potential Exploitation: Innovative Architectures and Illustrative Applications." *Proceedings of the IEEE*, Vol. 85, No. 1, pp. 24–38, January 1997.
4. Steinberg, A. N., Bowman, C. L., and White, F. E. "Revisions to the JDL Data Fusion Model." *Proceedings of SPIE*, Vol. 3719, Sensor Fusion: Architectures, Algorithms, and Applications III, pp. 430–441, 1999.
5. Steinberg, A. N., and Bowman, C. L. "Revisions to the JDL Data Fusion Model." In Hall, D. L., and Llinas, J. (Eds.), *Handbook of Multisensor Data Fusion*, Chapter 2. Boca Raton, FL: CRC Press, 2001.
6. Blasch, E., and Plano, S. "JDL Level 5 Fusion Model: User Refinement Issues and Applications in Group Tracking." *Proceedings of SPIE*, Vol. 4729, Signal Processing, Sensor Fusion, and Target Recognition XI, pp. 270–279, 2002.
7. Llinas, J., Bowman, C., Rogova, G., Steinberg, A. N., Waltz, E., and White, F. E. "Revisiting the JDL Data Fusion Model II." In *Proceedings of the 7th International Conference on Information Fusion (FUSION 2004)*, Stockholm, Sweden, 2004.
8. Blasch, E., and Plano, S. "DFIG Level 5 (User Refinement) Issues Supporting Situational Assessment Reasoning." In *Proceedings of the 8th International Conference on Information Fusion (FUSION 2005)*, Philadelphia, PA, 2005.
9. Bedworth, M. D., and O'Brien, J. "The Omnibus Model: A New Model for Data Fusion?" In *Proceedings of the 2nd International Conference on Information Fusion (FUSION 1999)*, Sunnyvale, CA, 1999.
10. Salerno, J. "Measuring Situation Assessment Performance Through the Activities of Interest Score." In *Proceedings of the 11th International Conference on Information Fusion (FUSION 2008)*, pp. 1–8, 2008.
11. Steinberg, A. N. "An Ontology for Multi Level Data Fusion." In *Proceedings of the 25th International Conference on Information Fusion (FUSION 2022)*, Linköping, Sweden, pp. 1–8, 2022.
12. Steinberg, A. N. "JDL Model." *ISIF Perspectives on Information Fusion (IPIF)*, Vol. 6, No. 1, pp. 36–50, June 2023.
13. Endsley, M. R. "Toward a Theory of Situation Awareness in Dynamic Systems." *Human Factors*, Vol. 37, No. 1, pp. 32–64, 1995.
14. Blasch, E., Steinberg, A. N., Das, S., Llinas, J., Chong, C. Y., Kessler, O., Waltz, E., and White, F. "Revisiting the JDL Model for Information Exploitation." In *Proceedings of the 16th International Conference on Information Fusion (FUSION 2013)*, Istanbul, Turkey, 2013.
15. Luo, R. C., and Kay, M. G. "Multisensor Integration and Fusion in Intelligent Systems." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, pp. 901–931, 1989.