Information Fusion 35 (2017) 148–165

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

A System-of-Systems perspective for information fusion system design and evaluation

Ali K. Raz [a,1,*], C. Robert Kenley [b,2], Daniel A. DeLaurentis [a,3,**]

[a] 701 W. Stadium Ave, Purdue University, West Lafayette, IN, 47907, U.S.A
[b] 315 N. Grant Street, Purdue University, West Lafayette, IN, 47907, U.S.A

ARTICLE INFO

Article history:
Received 25 November 2015
Revised 5 October 2016
Accepted 11 October 2016
Available online 12 October 2016

Keywords:
Information Fusion System
System-of-Systems
Model Based Systems Engineering
Design of Experiments

ABSTRACT

This paper provides a System-of-Systems (SoS) perspective for integrated design and evaluation of an Information Fusion System (IFS). IFS is comprised of distributed and heterogeneous systems that accomplish low-level and high-level information fusion (LLIF and HLIF) functionality. LLIF and HLIF functions are developed independent from one another but require collaboration to achieve the IFS mission objectives. The distribution and heterogeneity of systems, in addition to the multiplicity of LLIF and HLIF functions, creates an extensively large design space for the IFS. We apply a SoS engineering architecting process to obtain integrated architectures of IFS and propose guidelines to constrain an otherwise infinite design space of Information Fusion System-of-Systems (IF-SoS). Furthermore, we elaborate a multi-agent system modeling approach and pair it with Design of Experiments for objective evaluation of the IF-SoS design space. The statistical analysis, based on analysis of variance (ANOVA) and Tukey Honest Significant Difference (HSD) Range Tests, quantifies the impact of interactions between LLIF and HLIF design considerations on the IF-SoS performance. Furthermore, statistical evidence is provided to demonstrate that the interactions among JDL levels, in particular between LLIF and HLIF, are the most significant design considerations for fusion performance which necessitate an integrated design and evaluation of LLIF and HLIF—a manifestation of the SoS perspective for the IFS.

1. Introduction

Translating observation to information and extracting knowledge from information for enhancing situational awareness, reducing uncertainty, and improving decision making is a prime objective of information fusion. Information fusion as defined by the Joint Directors of Laboratories (JDL) Data Fusion task force is "a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats as well as their significance [1]." A system designed to accomplish these objectives is called an Information Fusion System (IFS). Fulfilling all the requirements of the above information fusion definition requires processing of information from multiple heterogeneous sources and systems. Therefore, an IFS is realized by the collaboration of multiple heterogeneous systems that individually satisfy certain attributes of the JDL information fusion definition and collectively fulfill the IFS objective. For instance, in an IFS, position estimation may be achieved by a radar, identity declaration by an Identify Friend or Foe (IFF) transponder system, and situation assessment by a command-and-control (C2) center.

It is not surprising to acknowledge that the constituent systems of IFS such as radar, IFF, and C2 systems are independently designed, developed, operated, and managed. For example, an IFS with a mission objective of tracking objects across a wide geographic span will likely include information sources and information processing systems distributed across the globe including ground, water, air, and space based resources. The design, development, operation and management of these systems is also expected to fall under different organizational entities (or agencies) with their own agendas, requirements and financial budgets. The IFS mission objectives are achieved when these independent systems collaborate with another and the information fusion functions are allowed to be distributed across these systems. It is this collaboration of distributed, heterogeneous, and independent

* Corresponding author.
** Corresponding author.
E-mail addresses: akraz@purdue.edu (A.K. Raz), kenley@purdue.edu (C.R. Kenley), ddelaure@purdue.edu (D.A. DeLaurentis).

[1] Ph.D. Candidate, School of Aeronautics and Astronautics
[2] Professor of Practice, School of Industrial Engineering
[3] Professor, School of Aeronautics and Astronautics

http://dx.doi.org/10.1016/j.inffus.2016.10.002
1566-2535/

systems spanning multiple organizational entities that corroborates the System-of-Systems (SoS) nature of the IFS.

In this paper, we formulate a system architecting process which takes into account the distribution, heterogeneity, and independence of IFS constituents systems and provides a SoS representation of the IFS. An Information Fusion System-of-Systems (IF-SoS) is represented by allocating functions to the physical systems across the architecture. The IF-SoS representation introduces the SoS paradigm for the IFS and enables application of the SoS analysis, tools, and methodologies for the IFS design and evaluation. This SoS paradigm also introduces guidelines for addressing the challenges arising from distribution of fusion functionality across multiple organizational entities and facilities objective evaluation of various IF-SoS architecture representations. We describe a SoS modeling methodology for the IF-SoS evaluation built upon a multi-agent system model to develop dynamic and executable models of the IF-SoS. We also provide a structured evaluation methodology that develops a design space of IF-SoS architecture representations resulting from multiple distinct allocated architectures. Utilizing the dynamic and executable models of the IF-SoS, we discuss a Design of Experiments (DoE) formulation for characterization of the IF-SoS design space.

1.1. Background and motivation

In order to achieve the stated objectives of information fusion definition, a Data Fusion Model (DFM) was developed by the JDL (now called the Data Fusion Information Group - DFIG) [2]. The JDL Data Fusion Model (JDL-DFM) provides a framework for describing an IFS while establishing a decomposition of activities and functions involved in the fusion of information from different heterogeneous sources. Fig. 1 provides an illustration of the JDL-DFM [2]. The JDL-DFM, starting with the information availability, decomposes the fusion processes into six different levels which are labeled as JDL Level 0 to 5. Each of these levels can be decomposed into functions, and further into algorithms that implement the desired objectives of an individual JDL level. Table 1 summarizes the scope of each of the six JDL levels along with examples of some of the functions that fabricate these levels. The JDL-DFM has received wide coverage in literature [3–6] in the past few decades and the summary of the JDL levels provided in Table 1 is derived from these references.

Stepping through the information fusion levels of the JDL-DFM, it is easy to overlook the shifting complexity and inference between the different JDL levels. Level 0 and Level 1 in the JDL-DFM consider characterization of the detectable entities and their present states. Whereas, JDL Level 2 and beyond attempt to infer the implications of future states of observed entities. Detection and estimation of entities versus characterization of situations and anticipations of their effects requires a paradigm shift in epistemology. Waltz and Llinas recognized the need to distinguish between the two and classified Level 0 and Level 1 as Low-Level Information Fusion (LLIF) and the remaining levels as the High-Level Information Fusion (HLIF) [7]. The early research and developments in information fusion typically focused on the LLIF developments [7] which left a gap between LLIF and HLIF [8]. Consequently, the recent trends for the IFS research are focused on HLIF developments [9,10]. The distinction between LLIF and HLIF, and a disparate treatment from the research community may incorrectly imply independence between the two disciplines. However, in "reality they are coupled" [11]. Achieving the information fusion capability and IFS objectives requires collaboration between LLIF and HLIF. Therefore, an integrated LLIF and HLIF design and evaluation methodology is required for the IFS.

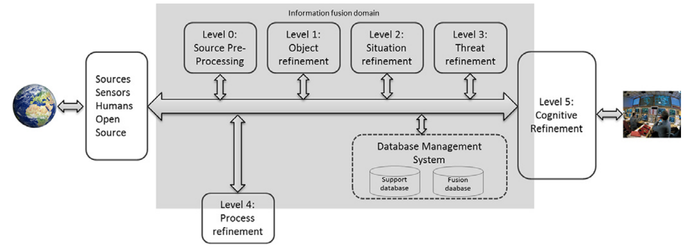


Fig. 1. JDL data fusion model [6].

The LLIF and HLIF integration and evaluation challenges for the IFS design are evident across various texts, e.g., Liggins et. al. [5], Hall et. al. [6], and Blasch et. al. [9]. Over the years, a number of frameworks such as the OODA (Observe-Orient-Act-Decide) [12], the Dual Node Network (DNN) architecture by Bowman and Steinberg [13], the state transition data fusion (STDF) model by Lambert [14], and Solano's Recombinant Cognition Synthesis [15] have been proposed to address some of the challenges of the IFS. A recent paper (2013) by Solaiman & Bossé et. al. provides a high level analysis of some these frameworks describing their applicability while highlighting the subsisting need for a holistic integration of all the JDL levels [16]. Fusion 'system design' still remains one of the grand challenges for the information fusion research community [11,17] and there is pertinent need to address this challenge as new capabilities such as multi-intelligence fusion [18] and information exploitation [19] (to name a few) are envisioned for IFS.

This paper addresses the well-founded need for an integrated design and evaluation of an IFS by taking into account the distributed environment and heterogeneity of the constituent elements of an IFS that perform the LLIF and the HLIF functions. From a Systems Engineering (SE) viewpoint, it can be observed that the JDL-DFM provides a basis for the development of an IFS, where the functionality associated with different fusion levels belongs to the individual elements of the IFS. However, the inherent

Table 1
JDL-DFM levels.

JDL-DFM levels	Scope	Example functions
Level 0: source pre-processing	Pixel and signal level data characterization. Signals and features that are determined by measurements and observations	Source detection (signal processing and detection theory). Bias corrections, coordinate transformations etc.
Level 1: object assessment	Object location, parametric and kinematic information. Establishment of tracks, IDs and classification. Combination of multi sensor data	Data alignment, data/object correlation, kinematic and attribute estimation, kinematic and attribute fusion
Level 2: situation assessment	Contextual interpretation of objects, events, situations and their relationships	Object aggregation, automated/rule-based reasoning, complex pattern recognition
Level 3: threat assessment	Future projections of current situations and consequence determination	Situation aggregation, automated/rule-based reasoning, complex pattern recognition
Level 4: process Refinement	Resource management and adaptive fusion control in support of mission objectives	Process control, sensor and network modeling, multi-objective optimization
Level 5: user Refinement	Fusion system and human interaction	Knowledge representation and information display

complexity in designing, developing and operating a set of components that engender a particular JDL-DFM level's functionality demands a system (or often a set of systems) on its own accord. It has been suggested that a single centralized system that accomplishes functionality of all the levels of the JDL-DFM is extremely difficult to build and will be prone to single point failures [20]. In fact, Waltz and Llinas claim that "there is no such thing as a data fusion system" [21] alluding that implementation of JDL-DFM levels cannot be achieved by a monolithic physical system. Similarly, Solano and Carbone advise against the design of a fusion system as a stand-alone system [22]. Furthermore, Roy et. al. recommends that a distributed system incorporating many 'heterogeneous components' is better suited for the development of an IFS [20].

The notion of distributed heterogeneous components (usually complex enough to be classified as a system in their own right) coming together to enable new capabilities is being observed across different disciplines in science and engineering. Complications in a few high stakes endeavors that envisioned capabilities resulting from collaboration of heterogeneous systems (e.g., U.S. Coast Guard's Integrated Deepwater System and U.S. Army's Future Combat System [23]) and a growing interest in the capabilities that can only be made possible from such collaboration demands a new approach of designing, developing and evaluating these systems. This emerging approach is recognized in literature as the System-of-Systems Engineering (SoSE) [24,25].

In this paper, we introduce a SoS perspective for achieving the functionality of the JDL-DFM levels by establishing Information Fusion System-of-Systems (IF-SoS). The IF-SoS allocated architecture holistically integrates all of the JDL-DFM levels by simultaneously incorporating heterogeneous and distributed physical systems which implement the various JDL-DFM levels. The multiple JDL-DFM levels allocated to different systems along with the interconnection of systems in various networks creates an extensively large design space for IF-SoS allocated architectures. The IF-SoS evaluation methodology—built upon SoSE principles specifically designed to study and facilitate collaboration of distributed, independent and heterogeneous systems—then facilitates characterization of the IF-SoS design space. One of the key developments of the IF-SoS evaluation methodology is formulation of Design of Experiments (DoE) to quantify the impact of design decisions on the IF-SoS performance.

The remainder of this paper is structured as follows: Section 2 introduces SoS architecting methodology for representation and evaluation of the IF-SoS. Section 3 elaborates its application at a conceptual level for IF-SoS allocated architectures and Section 4 discusses conceptual modeling constructs for the evaluation of the IF-SoS. Section 5 describes an application example of the IF-SoS architecting process for a multi-sensor multi-target tracking example where Design of Experiment formulations are used to quantify the IF-SoS design space.

2. Methodology

The building foundation of this work is the SoS architecting process model illustrated in Fig. 2. This process model is derived

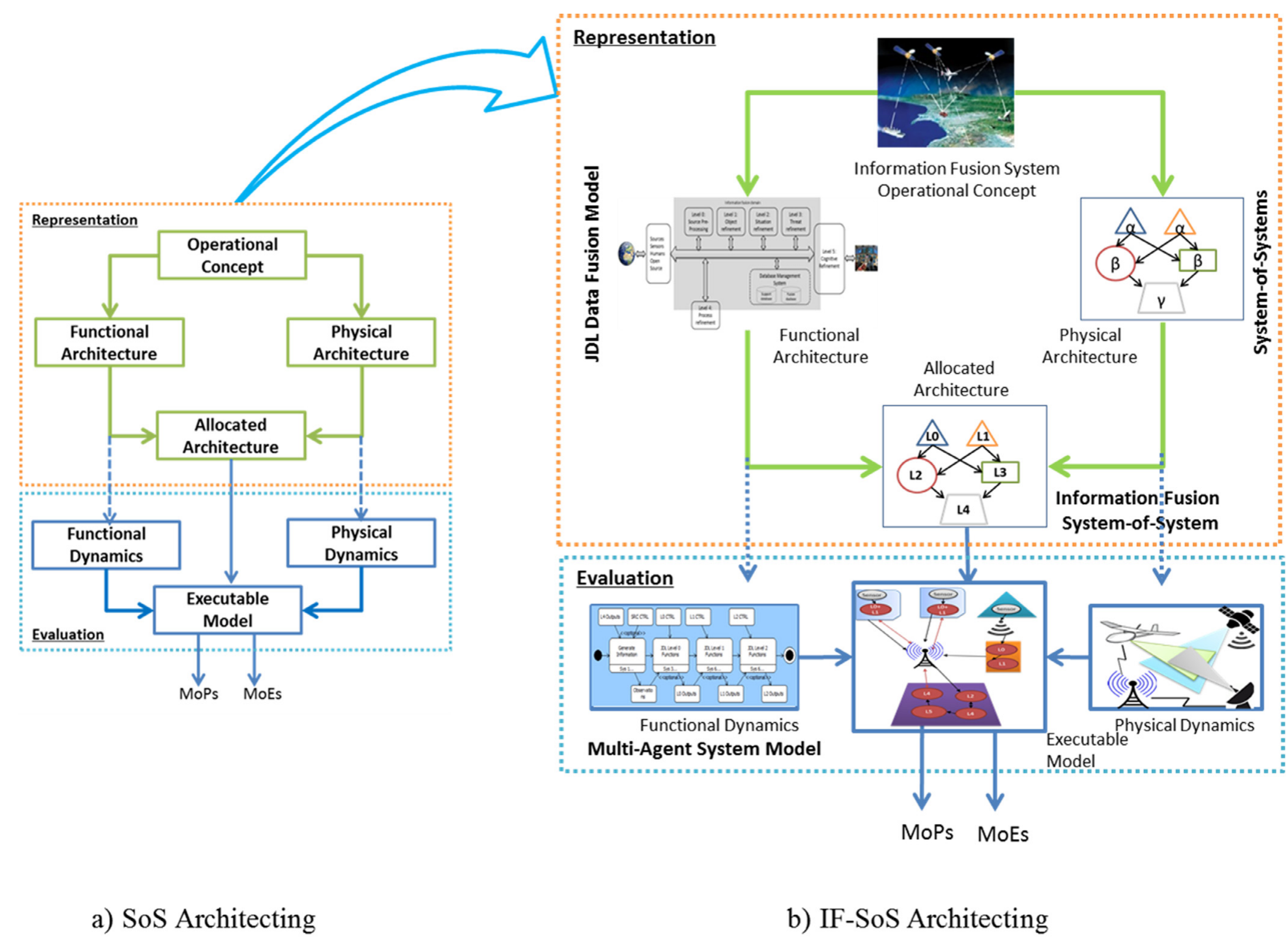


Fig. 2. Information Fusion System-of-Systems architecting process.

from system architecting process proposed by Levis and Wagenhals for developing Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) system architectures [26] and was later applied by Buede [27] as a key part of his overall process for engineering design of systems. Even though, this process was originally developed as a Systems Engineering process, it remains equally valuable to the System-of-Systems Engineering (SoSE). Sindiy in his PhD dissertation has discussed applicability of this process to the SoS [28] and Kenley et. al. have applied this process for synthesizing SoS architectures [29]. Building upon Kenley et. al. application, we demonstrate an application of this process to obtain a SoS representation and evaluation methodology for the IFS. Fig. 2 illustrates the SoS architecting process separated in two sections: 'Representation' and 'Evaluation' and the following paragraph provides a brief introduction to the constituent elements of this SoS architecting process.

In Systems Engineering, the mission statement of a system defines the 'Operational Concept' providing a vision of the system along with what is required of the system [27]. It is the operational concept that institutes whether a monolithic system, a SoS or an IFS is envisioned to achieve the stated mission objectives. A functional architecture is a set of activities or functions that need to be accomplished to meet the mission statement; whereas, the physical architecture is a representation of physical resources and their connectivity in support of the mission statement. The allocated architecture results from the allocation of a particular functionality from the functional architecture to the physical resource(s) of the physical architecture [26]. The functional and physical dynamics models along with the executable model facilitates simulating the allocated architecture to obtain performance metrics [29] i.e., Measures of Performance (MoPs) and Measures of Effectiveness (MoEs). The performance metrics then characterize attainment of the mission statement by the allocated architecture.

Given an operational concept that requires fusion of information, we discuss an application of the SoS architecting process depicted in Fig. 2 to obtain a SoS representation of the IFS and an evaluation model for the IF-SoS. For the IFS, the JDL-DFM provides a decomposition of different functions that facilitate fusion of information to realize the IFS operational concept and mission statement. Therefore, the functional architecture of the IFS is derived from the JDL-DFM [4,30]. On the other hand, the physical architecture of an IFS cannot be described by a physical architecture of a monolithic system, as highlighted in Section 1.1. Due to the distributed operational environment and the heterogeneity of elements, the physical architecture of an IFS is best described as a SoS. The allocation of the JDL-DFM functional aspects of the IFS to the physical architecture of the SoS then results in an allocated architecture which establishes the SoS representation of the IFS. This allocated architecture – called the IF-SoS – provides the basis for developing an evaluation methodology of the IFS that acknowledges distributed environment and heterogeneity of systems which constitute the IF-SoS. The SoS representation of the IFS qualifies the SoS tools and analysis techniques for the IF-SoS. For the IF-SoS evaluation, an Agent Based Model (ABM) is used where the individual agents model the IF-SoS functional and physical dynamics. The execution of multiple agents then provides the executable model for simulating IF-SoS allocated architectures, which is accomplished by Purdue University developed Discrete Agent Framework (DAF).

The next two sections detail application of the SoS architecting methodology for IF-SoS architecting as illustrated in Fig. 2.

3. Information Fusion System-of-Systems representation

3.1. IFS functional architecture: the JDL data fusion model (DFM)

The functional architecture of the IFS, i.e., the set of activities needed to be performed by the IFS, can be derived from the JDL-DFM illustrated in Fig. 1 [4,30].

In this paper, we utilize the Integrated Definition for Function Modeling (IDEF0) diagram for developing the functional architecture of the IFS based on the JDL-DFM levels. IDEF0 diagrams are a powerful SE tool for representing functional architectures [27] and are defined by the National Institute of Standards and Technologies (NIST) [31]. Given the IDEF0 popular and valuable application for system design in the SE community, Steinberg and Snidaro advocate its usefulness for IFS design [32]. The fundamental constructs of an IDEF0 diagram for functional decomposition are the functions and inputs, controls, outputs, and mechanisms (ICOM). For representing functional architectures, only the inputs, outputs, and controls are used. Mechanisms are used to represent the systems of the physical architecture that are allocated to the functions when defining the allocated architecture. The controls of a function determine constraints and considerations for transforming inputs into outputs. Hence, the controls account for the algorithms/ methods used by a function as well as the contextual information available for functional execution. For example, controls of LLIF functions (JDL L0 and L1) may include sensor registration, tracking, and sensor fusion algorithms along with the contextual information such as the weather data for electro-optical target trackers [33,34].

A complete functional architecture of the IFS will depict the inputs, outputs and controls of the necessary JDL functions required for achieving the IFS mission objective. However, it is important to note that the necessary JDL-DFM functions, their inputs, outputs, and controls may differ based on the mission objective(s) and remain particular for a given operational concept. Therefore, a complete functional architecture, depicting sufficient functional decomposition for functional allocation remains application specific and may not necessarily include all JDL levels. Nonetheless, a general framework for representing IFS functional architecture, based on information sources and the six JDL-DFM levels described in Table 1, is illustrated in Fig. 3. This abbreviated representation illustrates that observations are generated by the generate information function which are then processed by the by JDL L0 and the JDL L0 outputs by the JDL L1 function. The JDL L1 outputs are consumed by all the remaining JDL-DFM levels along with their predecessor outputs. The multi-interactions between the JDL levels in this functional architecture comes from the feedback of JDL L4 and the JDL L5 functions whose outputs provide additional inputs for different functions. In Section 5, we apply the IDEF0 framework illustrated in Fig. 3 to generate JDL-DFM functional architecture for a multi-sensor multi-target tracking application. Moreover, it is important to note that alternative functional architecture depicting more complex interactions between JDL levels, such as the JDL L5 (user) interaction with all fusion levels as suggested by Blasch and Plano [35], can be easily constructed with the IDEF0 diagrams.

Even though, the JDL-DFM elicits the functions required for achieving the IFS mission objectives, it lacks delineation of coupling and interdependence between different functions and system elements. The information fusion capability and the IFS mission objectives are only achieved when the information is exchanged between different JDL-DFM levels. This includes information exchange and interactions among LLIF/HLIF levels and between LLIF & HLIF levels. Recognition of information exchange and coupling between different levels is facilitated by understanding of the IFS physical architecture. However, it was discussed earlier in Section 1.1 that a single system that performs both LLIF and

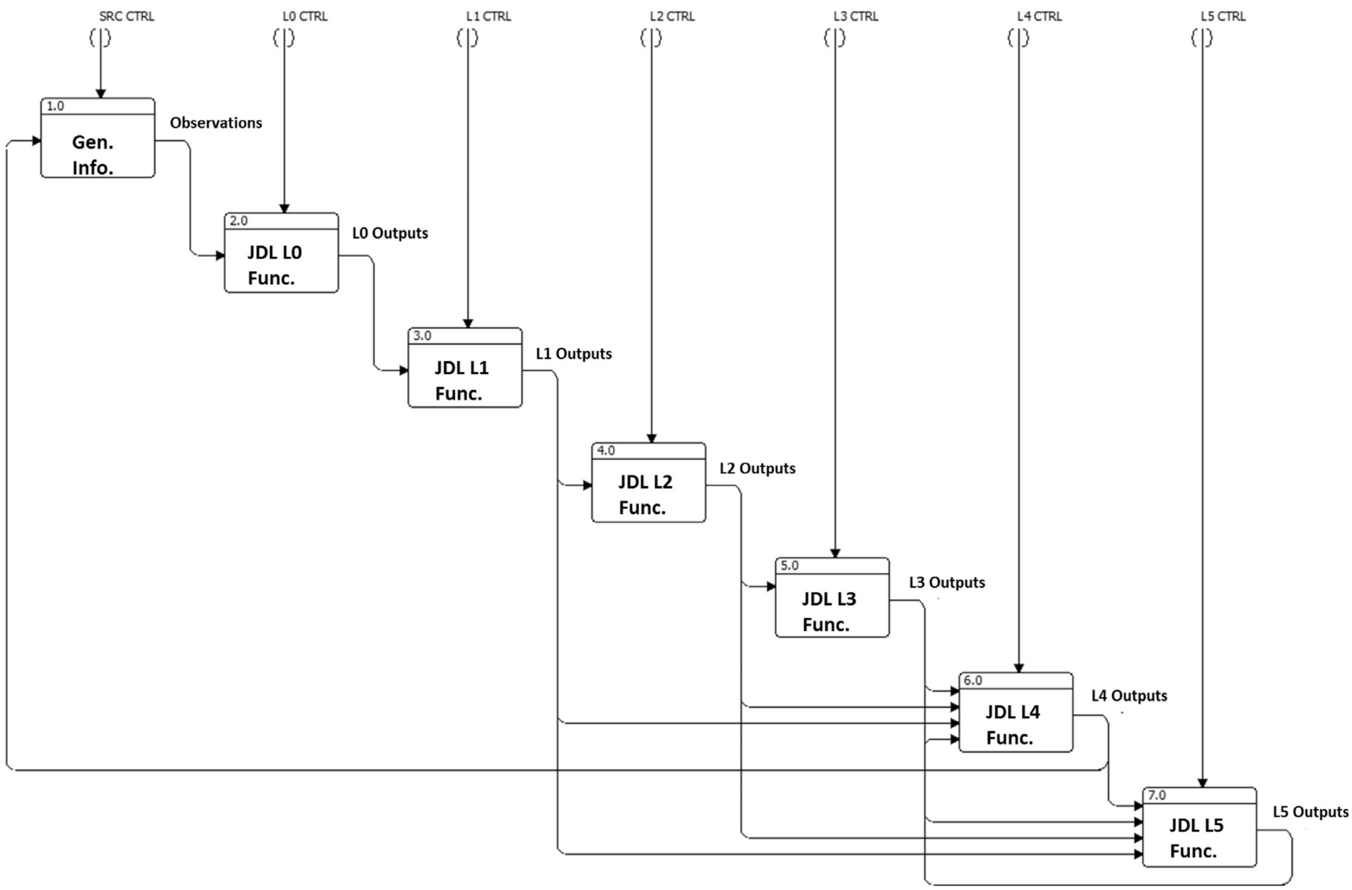


Fig. 3. IDEF0: JDL-DFM Functional architecture framework.[4]

HLIF functions is deemed impractical, concluding that the IFS physical architecture cannot be described by a monolithic system. An effective allocation of the JDL-DFM levels will require that different functions are allocated to distributed heterogeneous systems while the coupling between levels remains preserved. The physical architecture of a SoS is inherently built around this notion and is discussed in the next section.

3.2. IFS physical architecture: Systems-of-Systems

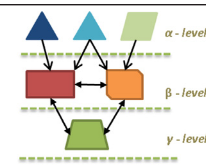
The physical architecture of the IFS is a System-of-Systems (SoS) which is a special system consisting of a collection of independent heterogeneous systems that interact on numerous levels in various networks for novel purpose. There are many different definitions of what is a SoS and which complex systems should not be confused with a SoS [24,36]. Jamshidi [25] has provided a detailed literature survey and analysis of different SoS definitions. The most common themes across different definitions of SoS include artifacts of independent operation of the member systems, synergy, collaboration and interoperability of otherwise distributed member systems along with evolutionary development and recognition of emergent behaviors [25,36].

In order to facilitate representation of a SoS, DeLaurentis et. al. [37] have developed a lexicon that provides an effective hierarchical decomposition for establishing physical architecture(s) of a SoS. This lexicon decomposes the member systems of the SoS into various hierarchical levels denoted as α, β and γ levels and so forth. In this lexicon, α-level systems are the base level systems which cannot be further decomposed; whereas, β-level systems are a collection of α-level systems organized in a network, and subsequently, the γ-level systems are a collection of networked β-level systems. From an IFS perspective, the information sources and their support systems can be viewed as the α-level entities and a network of these sources as a β-level system which partially enables some of the LLIF and/or the HLIF fusion capabilities. The γ-level systems, which include the networked β-level systems, then provide the remaining information fusion capabilities. Furthermore, the inclusion of connectivity between physical systems and information flow is implicit in the lexicon as the hierarchical levels by definition include layered networked systems.

Table 2 provides a brief explanation of different hierarchical levels along with an illustration of the SoS decomposition into these levels. In this example, similar shapes with differing colors indicates presence of similar (but not identical) systems and different shapes depict the heterogeneity of systems. The connectivity of physical systems and information flow is depicted by black arrows. The example SoS decomposition illustrates that independent systems at each level interact with heterogeneous systems operating at the same level and also across SoS level(s). This decomposition provides an effective way to capture the heterogeneity of member systems along with their independence, distribution and interactions in a SoS. The SoS physical architecture representation results from the allocation of independent systems (physical resources) to the SoS hierarchical levels and the identification of the interconnections between independent systems.

[4] Fig. 3 is developed using Vitech's Model Based System Engineering (MBSE) software called CORE. Using MBSE tools for architecture development ensures logical consistency throughout the architecture development process. (http://www.vitechcorp.com/products/core.shtml)

Table 2
SoS hierarchical distribution / physical architecture.

SoS level	Description	SoS physical architecture
α	Base level of entities that cannot be further decomposed	
β	Collection of α-level systems organized in a network	
γ	Collection of β-level systems organized in a network	

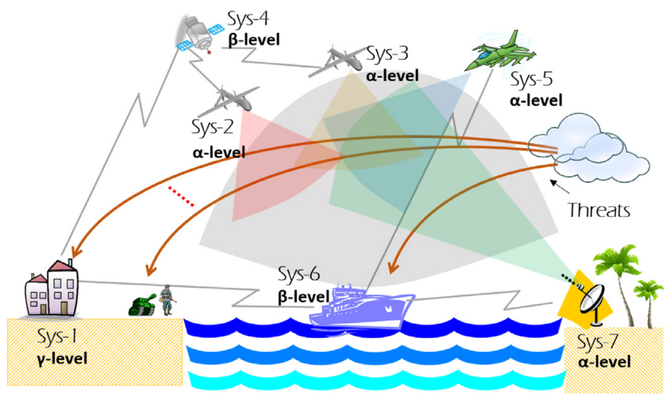


Fig. 4. Multi-sensor multi-target tracking System-of-Systems.

In order to describe a physical architecture in the information fusion application context, consider an example of a multi-sensor multi-target tracking system depicted in Fig. 4. In this example, physical systems such as radars, manned and unmanned aircraft, satellite, ship, and a command and control center are used to detect threats. In the SoS lexicon, both the different aircraft and the radar can be considered as α-level systems (i.e., the base entities that cannot further be decomposed), the satellite and the ship as β-level systems as they enable a network of α-level systems, and finally the command and control center as the γ-level system. Identification of distribution of these physical resources along with their connectivity constitutes the SoS physical architecture. However, in order to achieve the objective of the multi-sensor multi-target tracking system, these physical systems are required to accomplish certain JDL levels and functions as determined by the functional architecture. Allocation of the functions to the physical resources to meet the mission objectives instigates the allocated architecture which is described next.

3.3. IFS allocated architecture: Information Fusion System-of-Systems

This section describes the final step of the Information Fusion System-of-Systems representation that brings together the functional and physical architectures, instigating formation of an allocated architecture of the IF-SoS. As discussed earlier in Sections 3.1 & 3.2, the JDL-DFM levels define the functions that needs to be accomplished by the IF-SoS and the SoS hierarchical levels identifies the interconnected physical resources/systems that constitute the IF-SoS. The allocation of the JDL-DFM levels (functional architecture) to the SoS hierarchy (physical architecture) then establishes the allocated architecture of the IF-SoS. The allocated architecture of the IF-SoS identifies which systems are responsible for what levels (i.e., functions) of the JDL-DFM. In essence the allocated architectures establishes the roles and responsibilities of individual systems that collectively constitute the IF-SoS. The IF-SoS allocated architecture can be viewed as an enterprise architecture conglomerating information fusion functions and systems. From a service-oriented architecture [38] point of view, the JDL levels can be considered as services, the physical resources as means to provide these services, and the IF-SoS allocated architecture as an enterprise architecture which integrates services and systems to achieve mission objectives.

Fig. 5 provides an example of two notional IF-SoS allocated architectures depicting a conceptual illustration as well as a Model-Based Systems Engineering (MBSE) implementation in CORE which ensure functional and physical architecture consistency. The two IF-SoS allocated architectures, albeit benign prima facie, differ from each other with contrasting considerations in systems and networks that constitute the allocated architecture. In the IF-SoS allocated architecture 1, two distinct information sources are illustrated at the α-level. One set of information sources require a system with JDL-DFM Level 0 functionality at the α-level (shown as a blue triangle in the IF-SoS allocated architecture 1); while the other has the JDL-DFM Level 0 allocated as part of the information source itself (shown as green parallelograms). For example, one could consider the information sources connected to the JDL-DFM Level 0 system (blue triangle) as providing raw infra-red heat signatures which could be meaningless without JDL-DFM Level 0 functionality. An example of the information sources shown as green parallelograms can be sensors that provide processed information (e.g., with bias correction and coordinate transformations etc.) such as radar measurements. A network of these independent sources then constitutes the β-level of the IF-SoS where the JDL-DFM Level 1 and Level 2 functionality is accomplished by two similar but independent systems. Finally, at the γ-level of the IF-SoS, Level 3 and 5 are accomplished by one independent system and Level 4 by another one. On the other hand, in IF-SoS allocated architecture 2, both the Level 0 and Level 1 functionality is allocated at the α-level. For example, the information sources shown as green parallelograms for the IF-SoS allocated architecture 2 can considered to be sophisticated radars systems that have the capability to provide tracks and classification of detected objects. A network of these independent sources that already have information at Level 1 processing then constitutes the β-level of the IF-SoS allocated architecture 2, where all the remaining levels of JDL-DFM are accomplished. Since all of the JDL-DFM levels are accomplished between the two-levels of SoS distribution, a γ-level is not required in the IF-SoS allocated architecture 2.

A parallel analogy of these allocated architecture can be easily drawn to the multi-sensor multi-target SoS depicted in Fig. 4. In reference to IF-SoS allocated architecture 1, consider that the information generation and JDL L0 is allocated to the α-level systems shown in Fig. 4, the β-level systems (i.e., the satellite—Sys-6 and the ship—Sys-7) are allocated JDL L1 and L2 functionality, and the remaining functionality (JDL L3, L4, and L5) is accomplished by independent systems at command and control center. Now for the IF-SoS allocated architecture 2, consider that the α-level systems are allocated information generation, JDL L0 and L1 functionality and the staellite (Sys-4) and the ship (Sys-6) are not present. In this case, the α-level systems directly communicate with systems at the command and control center which provides the remaining JDL functionality as shown by IF-SoS allocated architecture 2 in Fig. 5 (it should be noted that an example of information sources

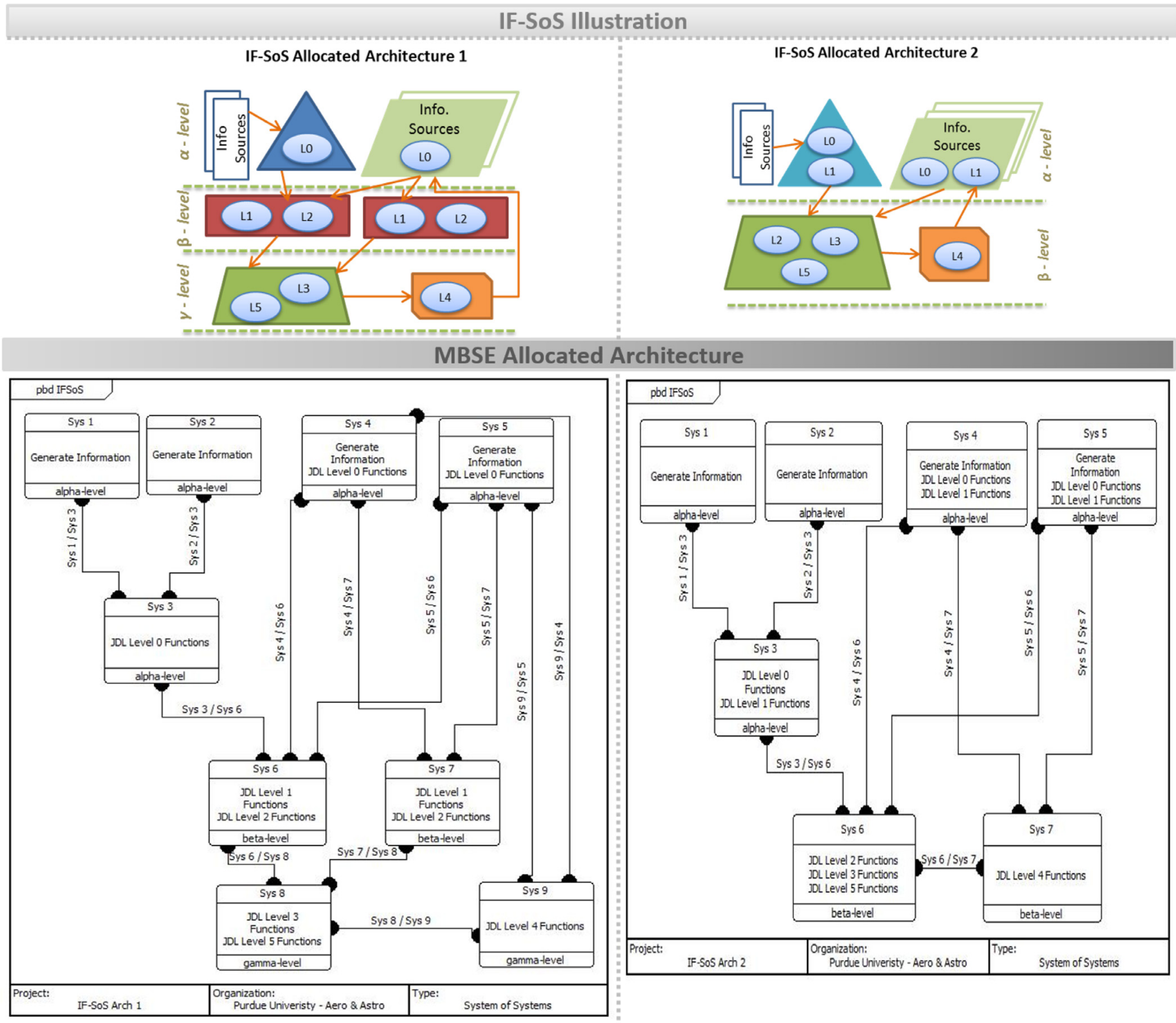


Fig. 5. Notional IF-SoS allocated architectures.

with a separate α-level system with JDL L0 i.e., the blue triangle is not shown in Fig. 4)

The IF-SoS allocated architectures depict the allocation of the JDL-DFM levels to different systems along with the information flow between different systems. Hence, the IF-SoS allocated architectures depict the information flow between the JDL-DFM levels. It was discussed earlier that the JDL-DFM Levels 0 and 1 are classified as LLIF and the remaining levels as HLIF. An important contribution of the IF-SoS allocated architecture is that it directly facilitates recognition of the coupling between LLIF and HLIF which remains inconspicuous in the functional architecture alone. For example, in IF-SoS allocated architecture 2, all of the LLIF levels are allocated at the α-level of the IF-SoS and all of the HLIF levels are allocated at to the β-level. On the other hand, in IF-SoS allocated architecture 1, α- and γ-levels remain dedicated to LLIF and HLIF respectively; whereas, the β-level signifies that some LLIF and HLIF can co-exsist at the same IF-SoS level.

The characteristics of the two IF-SoS allocated architectures shown in Fig. 5 remain distinct from one another. Theses characteristics begin to diverge from the acquisition of different systems and information sources, their design and requirements, the network considerations for enabling these allocated architecture, and continue to be disparate all the way to the end performance of the IF-SoS (in Section 5.3, the performance impact of different IF-SoS allocated architectures is established by DoE). Furthermore, it should be noted that the JDL-DFM levels are considered to be a high-level abstraction (not be confused with HLIF) of the information fusion functions [6]. These levels can be further decomposed into functions as briefly discussed in Section 3.1. Depending upon the abstraction of allocation, it might be desirable to allocate functions of one JDL-DFM level across different SoS levels. For example, some functions of JDL-DFM Level 1 (e.g., state estimation, filtering etc.) can be allocated to the α-level of the IF-SoS, whereas some other functions of Level 1 (e.g., correlation, classification etc.) can be allocated to the β-level of the IF-SoS. Nevertheless the abstraction level, it is evident that the allocation of the JDL-DFM levels to independent distributed systems can result in an incomprehensibly vast design space for the IF-SoS. On the one hand, this extended design requires a highly flexible evaluation mechanism for evaluation of the allocated architectures of the IF-SoS. On the other

Table 3
IF-SoS design space constraints.

Architecture	Design space constraints considerations
Functional (JDL-DFM limitations)	• Development maturity of JDL-DFM functions, methods and algorithms • Necessary functional flow of the JDL-DFM (based on functional implementation)
Physical (SoS limitations)	• SoS Types (Directed, Acknowledged, Collaborative, Virtual) [36,39] • System(s) geographical distribution and available communication technologies
Allocated (IF-SoS limitations)	• IF-SoS ROPE (Resources, Operation, Policy, Economics) [37] • IF-SoS Taxonomy (Autonomy & Control, Connectivity, System-Human Integration) [40] • Enterprise Architecture Principles [18,22] and Information Management [9]

hand, the lack of uniqueness of the IF-SoS allocated architectures necessitates establishment of guidelines that govern the allocation of the JDL-DFM levels to the SoS physical architecture. The following paragraphs briefly highlight these allocation guidelines while the flexible evaluation approach is detailed in Section 4.

IF-SoS allocation guidelines:. The infinitely many options of allocating the JDL-DFM levels to the SoS physical architecture may appear to be only limited by the resources of the SoS physical architecture (i.e., the total number of systems in the SoS). Although, innate constraints due to the available (or acquirable) resources cannot be undermined, there are additional guidelines that remain particular to the functional, physical and allocated architectures of the IF-SoS. Limitations and design consideration applicable at both the functional and the physical architectures determine the feasibility of the IF-SoS allocated architectures, bounding an otherwise infinite design space. Consider the SoS illustrated in Fig. 4 and it can be easily comprehended that the JDL functional architecture and the SoS physical architecture constrains which functions maybe allocated to what systems, hence limiting the allocated architecture design space. Furthermore, additional considerations such as the fusion databases in enterprise architectures [18] and the information management considerations [9] impose further limitations on the IF-SoS allocated architectures. Some of these IF-SoS design space constraining considerations are discussed in [41] and are highlighted in Table 3. While a comprehensive discussion of implications of the JDL-DFM and the SoS constraints on the IF-SoS design space will be pursued in future, it is important to note that despite being extensively large, the design space of the IF-SoS remains finite. In future, we will develop a rule set for allocating IF-SoS architectures under the constraints imposed the JDL-DFM functional, SoS physical and the IF-SoS allocated architecture considerations listed in Table 3.

4. Information fusion System-of-Systems evaluation

Albeit the IF-SoS allocation guidelines that attempt to restrict the infinite dimensionality of the IF-SoS allocated architectures, allocation of the functional JDL-DFM to the physical resources of the SoS can result in an extensively large design space. Effective evaluation of this extended design space resulting from the non-uniqueness of the IF-SoS allocated architectures becomes of an imperative importance for the IF-SoS. System designers and decision makers cannot finalize an IF-SoS configuration without cognizance of the possibilities in this design space and characterization of their implications. Evaluation of this large design space requires a highly flexible evaluation model. In particular, it requires such a model that allows:

(i) flexibility in JDL-DFM levels implementation,
(ii) capability of representing systems with varying functionality of one or more JDL-DFM levels and
(iii) adaptability of linking multiple systems in SoS hierarchies.

The SoS architecting process of Fig. 2 divides the SoS evaluation in two major sections: a) Functional and Physical Dynamics and b) Executable Model. The following paragraphs elaborate on these two sections where an Agent-based modeling (ABM) approach is discussed for modeling the functional and physical dynamics model, and a simulation tool called Discrete Agent Framework (DAF) is described to obtain an executable model of the IF-SoS. Furthermore, in Section 5, DoE formulations are presented which leverage ABM and DAF to quantify the impact of various design decisions on the IF-SoS.

4.1. Dynamics model

The purpose of a 'Dynamics Model' is to characterize "static manner aspects of the dynamic behavior of the model" [26]. The static aspects of the IF-SoS allocated architecture that comprise its dynamic behavior appertain to the individual system design of the member systems and their collaboration with one another. The system architecting process in its original form as proposed by Levis and Wagenhals [26] along with its refinements [38] and extension to SoSE [29] describes dynamics model as a singular item. However in this paper, we have introduced functional and physical dichotomy of the dynamics model. The functional dynamics, models the behavior of a system resulting from its individual design, while the physical dynamics accounts for the interactions and collaborations of one system with the others.

4.1.1. Functional dynamics

The functional dynamics of the IF-SoS directly result from the functional architecture which is comprised of the JDL-DFM levels. Each JDL-DFM level can be decomposed into one or many functions which are accomplished by several implementation methods (i.e., algorithms). For example, Hall and McMullen state that the JDL-DFM level 1 can be partitioned into four functions: (1) data alignment, (2) association, (3) tracking, and (4) identification [4]. Furthermore, each of these functions can be accomplished by several different implementation methods, for instance there are a number of algorithms to perform tracking functionality of Level 1 fusion (e.g., Kalman Filters, particle filters, Probability Hypothesis Density (PHD) filters etc.).

In the IF-SoS, a number of different JDL-DFM levels can be allocated to a system, subject to the JDL-DFM allocation feasibility guidelines. Therefore, the individual system design then becomes a function of the JDL-DFM level(s) and the level's functions allocated to a particular system along with the method used to implement the JDL-DFM level(s). This decomposition of an individual system of IF-SoS into functional dynamics is illustrated in Fig. 6.

In Section 5.3, the impact of different implementation methods and system allocations is quantified using an example of a multi-sensor multi-target tracking system by utilizing Analysis of Variance (ANOVA) and range tests in the DoE framework.

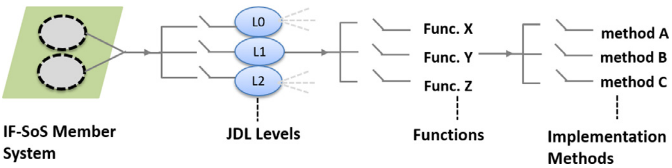


Fig. 6. Elements of the functional dynamics model of the IF-SoS.

4.1.2. Physical dynamics

The physical dynamics models the physical parameters and the collaboration between the constituent systems of the IF-SoS. The physical parameters necessitate deliberations of an individual system's operational and external environment; while taking into account the information exchange required for enabling collaboration between different systems. The physical parameters of an individual system may include its location, motion, and environmental considerations etc. For example, the ability of a sensor to generate a measurement depends on the sensor's location, it's relative motion, and weather conditions; meanwhile the IF-SoS objectives are dependent on the sharing of these measurements from one system to the others. The physical dynamics, therefore, elucidates the physical properties and communication considerations of all systems in the IF-SoS.

Similar to the functional dynamics parameters, Section 5.3 provides quantification of variation in the physical dynamics design decisions of a multi-sensor multi-target tracking system example.

4.1.3. Modeling functional and physical dynamics

All the various considerations of functional and physical dynamics combined with numerous possibilities of the IF-SoS allocated architectures demand a highly flexible modeling language. For the IF-SoS, it is imperative that the dynamic modeling language provides the flexibility of accommodating the various functional and physical dynamic considerations while allowing for specification of different IF-SoS allocated architectures.

In this paper, we utilize Agent-Based Model (ABM) as a dynamic modeling tool for the IF-SoS which is built for providing this very flexibility. In ABM, agents are heterogeneous autonomous decision making entities [42] and a system behavior can be represented by single or a collection of multiple agents. ABM provides a flexible framework for describing system behavior and rules of interconnections [42]. Joslyn and Rocha describe an agent as an autonomous entity that interacts with its operational environment and makes independent decisions based on its own desire and objectives [43]. Kenley et. al. describe the agent structure for modeling dynamic behavior of different functions in a SoS [29] and a similar decomposition can be used for creating agents for the IF-SoS.

Fig. 6 serves as a baseline criteria for developing agents for the IF-SoS. For the IF-SoS, each JDL-DFM level can be modeled as an individual agent. The functional dynamics then determine the agent functionality such that any given function is allowed to be accomplished from a variety of implementation methods. The final selection of the JDL-DFM function for an agent remains a function of the IF-SoS allocated architecture and the final selection of the a method for delivering that specific functionality becomes a decision criteria of the agent initialization routine as part of its instantiation. A systems' functionality, specified in the IF-SoS allocated architecture, is then described by a collection of one or more agents as shown in Fig. 6. The operational environment of co-located agents provides the physical parameters of an individual system while the rules of interconnections (input/output) between different systems accounts for information exchange and communication consideration of the physical dynamics.

Therefore, an agent-based design of JDL levels accords a flexible implementation of IF-SoS functional and physical dynamics which is required for measuring an IF-SoS allocated architecture ability to achieve the mission requirements. Consider for example the SoS concept illustrated in Fig. 4 and the two IF-SoS allocated architecture shown in Fig. 5. In IF-SoS allocated architecture 1, the α-level systems, such as a radar, are allocated L0 functionality where as in IF-SoS allocated architecture 2, the α-level systems were allocated JDL L1 functionality in addition to JDL L0 functionality. An ABM design of JDL levels provides an ability to flexibly obtain an individual system representations from multiple agents. Once JDL levels are implemented as independent agents, the α-system representation of IF-SoS architecture 1 is given by one (JDL L0) agent while the IF-SoS architecture 2 representation is obtained by assigning two agents (JDL L0 and JDL L1) to α-level systems.

An agent-based implementation of an IF-SoS member system captures the static aspects of its dynamic behavior. However, the holistic dynamics of the entire IF-SoS allocated architecture are only revealed when the multiple-agents are allowed to exchange information and interact with one another which is facilitated by the executable model.

4.2. Executable model

Creating a SoS representation of an IFS as an IF-SoS allocated architecture is a top down decomposition; whereas, obtaining an agent based representation of the member systems in the IF-SoS allocated architecture is a bottom up composition. The executable model brings the two together to provide a realization of the IF-SoS which can be objectively evaluated via pre-specified MoPs and MoEs.

The executable model of an agent-based system, often called an agent-based simulation (ABS), is a framework that allows the information exchange and interaction between multiple agent-based systems. Purdue University has developed a simulation framework – called the Discrete Agent Framework (DAF) – for simulating agent-based models. Built upon object-orientated programming principles, the core value proposition of DAF is to provide a flexible, extensible and a modular simulation framework for analyzing SoS behavior. Recently, DAF utility is demonstrated by its application to a wide range of SoS in various application domains: e.g., Command Control Battle Management and Communication Systems (C2BMC) [29,44], Littoral Combat Systems [45] and Air Transportation System [46] (an extended description of DAF can also be found in these references).

From an IF-SoS allocated architecture evaluation perspective, once agent-models are developed for different JDL-DFM levels, DAF facilitates creation and simulation of different IF-SoS allocated architectures. The modularity, extensibility and flexibility of DAF simplifies creation of multiple different IF-SoS architectures, aiding in the evaluation of the extended design space of the IF-SoS based on pre-defined domain specific MoPs and MoEs. In the following section, we demonstrate applicability of DAF for evaluating IF-SoS design space for a multi-sensor multi-target tracking system.

5. IF-SoS application

In this section, we demonstrate an application of the IF-SoS architecting process and it's manifestation of analytical capabilities that aid the design space characterization of the IF-SoS using a multi-sensor multi-target tracking example such as the one illustrated in Fig. 4. We begin with an information fusion mission objective of tracking multiple detected objects using different sensors and apply the methodology introduced in this paper to develop representation and evaluation models of the resulting IF-SoS. The development of these artifacts introduces a plethora of design variables which require a structured methodology for analysis of their IF-SoS performance implications. We utilize DoE techniques to quantify the impact of various design variables on the IF-SoS end performance, identifying the most significant considerations for achieving the IF-SoS mission objectives.

5.1. IF-SoS representation

5.1.1. Mission objective and scenario description

The mission objective for the IFS considered in this section is to detect and track objects (targets) traversing a wide geographical region using two passive and two active sensor resources. The scenario considered in this example utilizes three different targets, modeled with the identical dynamics but different trajectories. The passive and active sensors resources are positioned at fixed locations such that the three targets are simultaneously in all sensors Field of Regard (FoR), but not necessarily in their Field of View (FoV). The IFS mission objective is to established accurate tracks of all detected objects, where a track at time t is the state estimate of a target given by $\mathbf{x}(t) = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]$ providing target position (x, y, z) and it's velocity $(\dot{x}, \dot{y}, \dot{z})$ in Earth-Centered,Earth-Fixed (ECEF) coordinates.

5.1.2. Functional architecture and JDL-DFM functions

In order to accomplish the mission objective of tracking multiple objects using multiple sensors with the above constraints, we consider a subset of LLIF and HLIF functions. The scope and functionality of these LLIF and HLIF function is summarized in Table 4.

The functional architecture of the JDL-DFM functions utilized for achieving the IFS mission objective is developed using the IDEF0 diagram described earlier in Section 3.1. The input, output, and controls required for each function described in Table 4 are determined along with the required functional flow to meet the IFS mission objectives. The resulting functional architecture which identifies the inputs, outputs and controls is constructed via the IDEF0 diagram in Vitech's MBSE software CORE. This functional architecture, illustrated in Fig. 7, specifies that *Sen* function requires sensor commands to produce target measurements based on passive or active sensor control. The produced observations are processed by the *L0* function to produce measurements which are processed by the *L1-SE* function to output state estimates of the detected objects. The state estimates are input to the *L1-TF* function which produces the system level tracks. The system level tracks are consumed by the *L23-AE* function to determine tracking priorities. The *L4-ST* function then determines the sensor FoV commands based on the tracking priorities from *L23-AE* and the system level tracks from *L1-TF*.

It is important to reiterate that the functional architecture illustrated in Fig. 7 only takes into account a subset of the JDL LLIF and HLIF levels and functions. For example, the JDL L5 representing human interaction with the IF-SoS along with other LLIF and HLIF functions such as object classification functions and multi-intelligence fusion are not modeled for the particular application discussed in this section. The necessary and sufficient assumptions

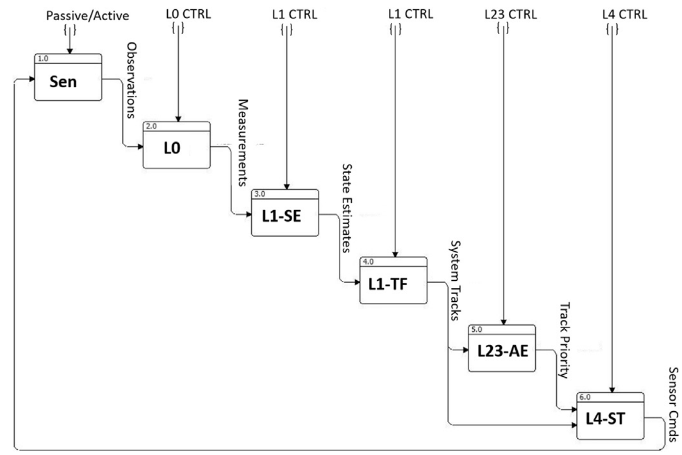


Fig. 7. IDEF0 - functional architecture of JDL-DFM functions.

for the JDL-DFM functions not included in this functional architecture are discussed later in the functional dynamics model in Section 5.2. However, it should be noted that the IDEF0 methodology and the analytical framework discussed in this section remains scaleable to accommodate all the various JDL levels and their functions in higher fidelity applications.

5.1.3. Physical architecture

For the multi-sensor multi-target tracking problem, the two passive and the two active sensors are considered as the α-level resource systems (i.e., base entities that cannot be further decomposed). In addition, it is assumed that LLIF and HLIF information processing systems capable of performing assigned JDL-DFM functions are also available which can be operated the α-, β- and γ-levels of the SoS. Aggregation of various information processing systems at different SoS hierarchic levels and sensor systems along with the information exchange between them produces the physical architectures. For achieving the IFS mission objective for the multi-sensor multi-target tracking problem discussed in this section, a number of SoS physical architectures can be conceived and are discussed in the following section along with the allocated JDL-DFM functionality.

5.1.4. IF-SoS allocated architectures

The objective of creating IF-SoS allocated architectures is to achieve the IFS mission objective by allocating JDL-DFM functions to the SoS physical resources such that the functional flow (determined by the functional architecture) is not violated. Given the α-level resources of two active and two passive sensors, a number of different IF-SoS allocated architectures can be produced for accomplishing the IFS mission objectives using the fan-in tree for LLIF functions and fan-out tree for HLIF functions as described in the data fusion dual node network (DNN) framework [13]. These allocation strategies and the resulting IF-SoS allocated architectures are discussed next. However, it should be noted that in creating

Table 4
JDL-DFM functions for multi-sensor multi-target tracking example.

JDL-DFM	Function	Abbev.	Scope & description
Source	Observation generation	*Sen*	Generate target observation based on active or passive control
Level 0	Measurement generation	*L0*	Passive and Active sensor measurement generation
Level 1	State est.	*L1-SE*	Generate state estimate (tracks) of detected objects based on sensor measurements
Level 1	Track fusion	*L1-TF*	Fuse multiple tracks corresponding to the same detected object to provide a single system level track
Level 2 & Level 3	Assessment & evaluation	*L23-AE*	Assess intent of the detected objects based on available system track and assign tracking priorities for tracking multiple objects
Level 4	Sensor tasking	*L4-ST*	Guide the sensors by providing field of view (FoV) commands for continuation of measurement generation based on tracking priority

the IF-SoS allocated architectures, complete flexibility of function-resource allocation is assumed which maybe difficult to attain especially when the function-resource allocation space spans multiple geographic regions and organizational entities. In the context of this example, the flexibility assumption facilitates identification of multiple possibilities for achieving the same IFS mission objective.

LLIF allocation strategy:. The LLIF allocation strategy allocates the JDL-DFM level 0 and level 1 functionality to different SoS hierarchies indicating the variations in individual system design and capabilities. Distribution of LLIF functions for multi-sensor multi-target tracking is a well studied problem in literature and multiple formulations have been studied in detail. We describe three different LLIF allocation strategies motivated by tracking architectures described by Bar-Shalom et. al. [47] and Chong et. al. [48]. The following paragraphs describe these strategies and the resulting LLIF allocated architectures are depicted in Fig. 8.

Distributed fusion strategy (LL1):. In this allocation strategy, LLIF functions i.e., *Sen, L0, L1-SE* and *L1-TF* are distributed across systems operating at α, β and γ levels such that independent state estimates of passive and active sensors are produced. The physical SoS architecture in this strategy provides the track-to-track fusion system with two independent tracks for any detected object, one based on passive sensors information and the other based on active sensors information.

Hybrid fusion strategy (LL2):. In the hybrid fusion strategy slight variation in the SoS physical architecture results in one hybrid state estimate based on passive and active sensor information and the other state estimate based on a single active sensor information. The implications of the SoS physical architecture variation leads to different design considerations and requirements for α and β systems. In the Hybrid Fusion strategy, one of the α-level system with the *Sen+L0* functionality now has additional requirements to perform *L1-SE* functionality and the β-level *L1-SE* system has to be able to process both passive and active measurements.

Independent fusion strategy (LL3):. In this strategy, the *L1-TF* system receives three independent tracks, one based on the passive sensor information and the others based on independent active sensor information. In this LLIF allocation strategy, the *L1-SE* functionality for active sensors is allocated to the α-level systems; whereas, the same functionality for processing passive sensor information is allocated to the β-level systems.

HLIF allocation strategy:. The HLIF allocation strategy is built on a similar philosophy as of LLIF allocations. The following paragraphs describe three different HLIF strategies and the resulting HLIF allocated architectures are depicted in Fig. 9.

Global awareness and global tasking (HL1):. In this allocation strategy all the HLIF functions are allocated to the γ-level of SoS. This provides the HLIF systems with centralized awareness of all sensors information along with a complete picture of what objects are being tracked by all resources. Furthermore, the allocation of *L4-ST* functionality at the gamma level allows for centralized tasking of individual sensor.

Global awareness and local tasking (HL2):. In this strategy, the *L23-AE* functionality is allocated at the γ-level providing for centralized awareness but the sensor tasking functionality (i.e., *L4-ST*)

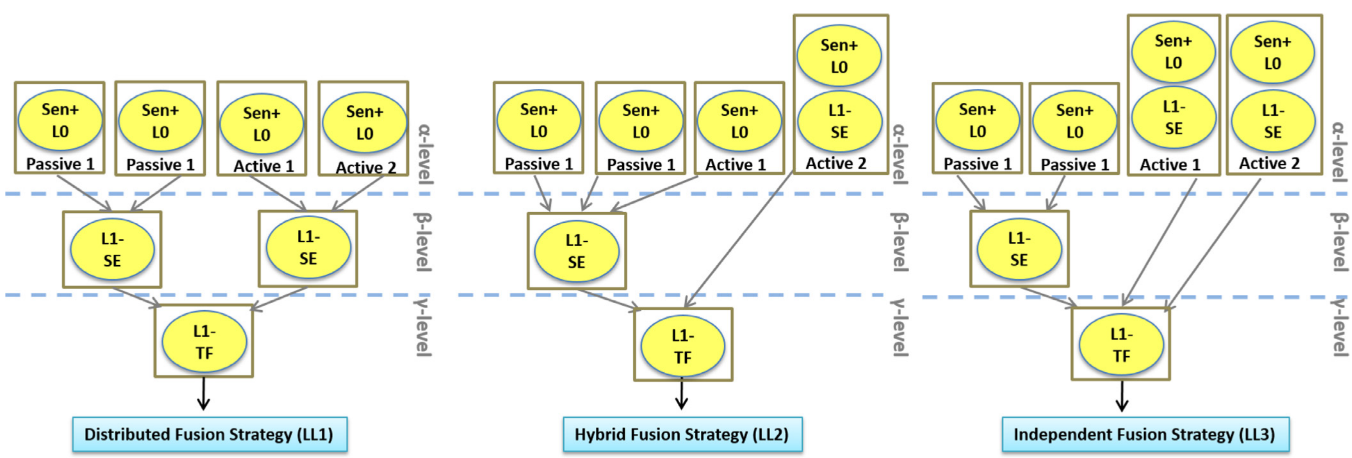


Fig. 8. LLIF allocation strategies and architectures.

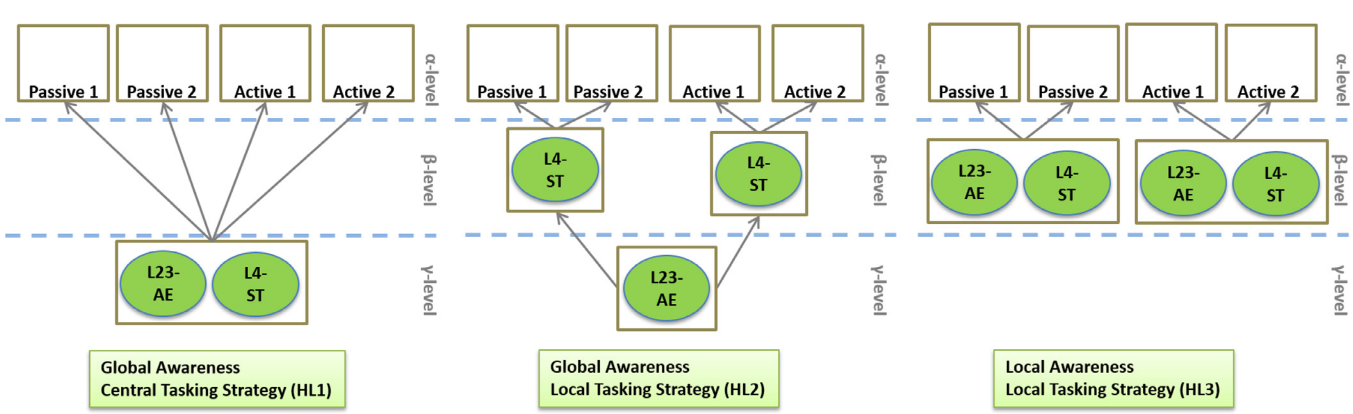


Fig. 9. HLIF allocation strategies and architectures.

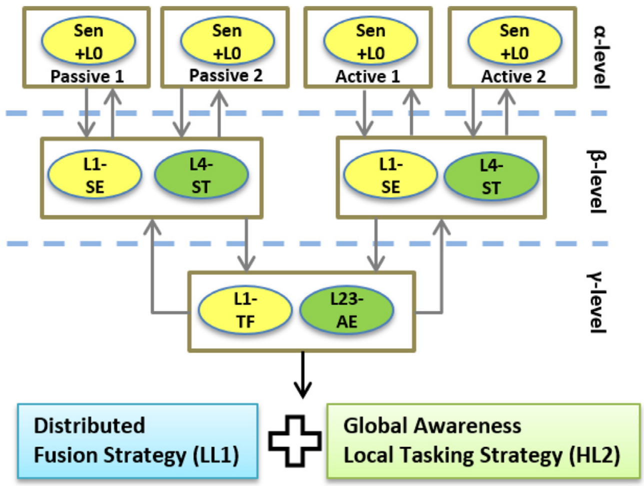


Fig. 10. IF-SoS allocated architecture (LL1 + HL2 strategy).

is distributed to β-level SoS. In this setup, the γ-level systems maintain awareness of all objects being tracked by all sensors but are not responsible for tasking all sensors.

Local awareness and local tasking (HL3):. In this strategy both the *L23-AE* and *L4-ST* functionality are independently allocated to the β-level systems for passive and active sensors set. In this allocation strategy, independent systems determine their own priorities and sensor tasking solely based on independent awareness.

Resulting IF-SoS allocated architectures:. The IF-SoS allocated architecture is given by integrating any LLIF allocated architecture with any HLIF allocated architecture, insofar the IDEF0 functional architecture illustrated in Fig. 7 is preserved. For example, the mission objective of tracking multiple objects with multiple sensors can be achieved when hybrid fusion strategy (LL2) is integrated with global awareness and global tasking (HL1) or the IFS mission objective can also be achieved if the distributed fusion strategy (LL1) is integrated with global awareness and local tasking (HL2) strategy. The resulting IF-SoS allocated architecture from integrating the latter two combinations of LLIF and HLIF allocations strategies is illustrated in Fig. 10.

In order to visualize the IF-SoS allocated architecture shown in Fig. 10, refer to the multi-sensor multi-target tracking system shown earlier in Fig. 4. In the resulting IF-SoS—given the physical resources of Fig. 4 and allocation strategy of Fig. 10—the α-level systems (Sys-2,3,5, and 7) will generate processed measurement information. The β-level systems (Sys-6 and 7) will provide independent state estimates of the detected object(s) based on the information from their locally connected α-level systems. Furthermore, the β-level systems will also be responsible to task their α-level systems to continue tracking of the detected objects. Finally, the γ-level system will receive state estimates from both β-level systems, fuse the state estimates, perform assessment and evaluation of the detected objects, and communicate tracking priorities to the β-level systems.

It can be clearly seen that based on the different LLIF and HLIF allocation strategies a total of $3^2 = 9$ different IF-SoS allocated architectures can be created which remain capable of achieving the mission objective. Although these allocated architectures conform to the same functional architecture, design considerations for individual systems become remarkably different based on the allocated functionality and the physical architecture akin to LLIF/HLIF allocation strategies. It becomes imperative to characterize how the differences in allocated architectures impact the end performance of achieving the mission objective. This leads to the evaluation section of the IF-SoS architecting process which is discussed next for the multi-sensor multi-target tracking example.

5.2. IF-SoS evaluation

5.2.1. Dynamics model

The salient features of the IF-SoS which influence its behavior reside in its functional and physical dynamics model. The JDL-DFM functions described in Table 4 are modeled as an independent agent. As discussed in Section 4.1.3, agents are autonomous decision making entities which provide a flexible implementation of the functional and the physical dynamics of an IF-SoS. The details of these agent models of the JDL-DFM functions elaborating the considered functional and physical dynamics are described below.

Sen+L0 agent:. The *Sen* and *L0* functionality is molded as a single agent called Sen+L0 which produces measurements when an object lies within a sensor's FoV. A sensor agent's ability to accurately generate passive or active measurement constitute its functional dynamics. Each Sen+L0 agent can be configured as either a passive or active sensor sensor. Passive sensors are limited in their measurement generating abilities and provide a two-dimensional measurement for object azimuth and elevation but have a wider field of view. Active sensors, on the other hand, may have limited field of view but provide target range measurement in addition to azimuth and elevation [47]. The physical dynamics of a sensor agent models its ability to share the measurements with other agents and/or systems in the IF-SoS.

The variable design options modeled for the Sen+L0 Agent include:

- Varying measurement communication rates for active sensors, qualitatively labeled as low, medium and high. Passive sensors are modeled with constant measurement communication rates.
- Varying measurement accuracy of sensor agents, qualitatively labeled as low, medium and high. Variations in measurement accuracy of active sensors is modeled independently, whereas passive sensor accuracy varies together. This means that the two active sensors can independently operate at different measurement accuracy (i.e., one active sensor provides low measurement accuracy and the other provides high accuracy), but both passive sensors will always operate at the same selected level.

L1-SE agent:. The objective for the state estimation functionality is to establish state estimates (3D tracks) of the detected objects. These state estimates are based on active or passive measurements. The functional dynamics of the L1-SE agent are modeled by an Extended Kalman Filter (EKF) [47] to process sensor measurements for establishing tracks based on multi-sensor measurements. Since the passive sensors generate a two-dimensional measurement, a triangulation method [49] is used for passive sensor range measurement. Also it is important to note in a multi-sensor multi-tracking problem, data association remains a prerequisite problem for state estimation; we consider ideal data association solution for all multi-sensor multi-threat measurements. The physical dynamics of the L1-SE agent reflect its ability to transmit state estimates with other other agents and/or systems; the state estimation transmission rate is assumed constant.

The L1-SE functionality is modeled as a static agent providing state estimates using the EKF at a constant communication rate. No variable design options are modeled for L1-SE agent.

L1-TF agent:. The L1-TF agent provides track-to-track fusion of multiple tracks of the same detected object to establish a single system level track of the detected object. Similar to the data association problem, track-to-track correlation remains a prerequisite problem for track fusion which is modeled to provide an ideal solution. The track-to-track fusion (T2TF) is an extensively studied problem in literature and a number of different algorithms are available for its implementation.

The variable design options of the L1-TF agent include:

- Three different track fusion algorithms: 1) the Covariance Intersection (CI) [50], 2) the Covariance Intersection with Memory (CIM)[51], and 3) the Maximum Likelihood (ML) fusion [47] (the mathematical and implementation details of these algorithms can be found in the associated references).

The physical dynamics of L1-TF agent are modeled by a constant system track communication rate.

L23-AE agent:. The purpose of the L23-AE agent is to provide situational awareness and impact evaluation. The modeled L23-AE functionality determines the number of tracked objects based on the system level track and projects the impact location(s). The expected impact location is then evaluated against a pre-defined geographic area (e.g., an area of interest or defended area boundary) to determine object tracking priorities.

In this paper, all objects are targeted towards a hypothetical area of interest, and therefore are assigned the same tracking priorities by the L23-Agent which are communicated at a constant rate.

L4-ST agent:. Given the tracking priorities and system-level tracks, the L4-ST agent generates sensor tasking commands to direct sensors FoV to generate new measurements. Typically, the sensor tasking solutions are obtained by solving a large-scale optimization problem. The constraints and complexity of optimization formulation directly impact the time it takes to reach an optimal solution. Utilizing the mixed-fidelity model integration benefit of agent-based modeling approach, we simulate varying sensor tasking solution availability.

The variable design options of the L4-ST agent include:

- The sensor tasking solution availability, qualitatively described as high, medium, and low. This reflects the functional and physical dynamics of L4-ST agent, accounting for the time L4-ST agent takes to compute a solution and to communicate the solution to other agents.

5.2.2. Executable model and measure of performance:

The executable model provides the simulation capability of allocated architectures based on the dynamic model and produces metrics for evaluation of both the allocated architectures and the dynamic model considerations. We utilize DAF to provide the simulation of the multi-sensor multi-target tracking system based on the agent descriptions and the IF-SoS allocated architectures described earlier in this section. An object-oriented software model of each agent is developed in DAF and all the variable design options (specific to each agent as discussed the previous section) are parameterized for agent instantiation. In order to create multiple IF-SoS architectures, independent agents as determined by different LLIF (Fig. 8) and HLIF (Fig. 9) allocation strategies are connected together. The behavior of each individual agent is then initialized with the desired functional or physical dynamics given by it's parameterized variable design options.

The ability of an IF-SoS allocated architecture to achieve the mission objective based on specified functional and physical dynamic model considerations is evaluated using the system-level tracking error of all detected objects. The system-level tracking error is defined as the median position error of all detected objects, measured from L1-TF functionality, over mission duration. Mathematically, let $x_\tau(t) = (x_\tau, y_\tau, z_\tau)$ be the true position of a detected object and $x_s(t) = (x_s, y_s, z_s)$ its system-level position estimate at time t, the position error at time t is then given by:

$$Err_{pos} = \sqrt{(x_\tau - x_s)^2 + (y_\tau - y_s)^2 + (z_\tau - z_s)^2} \qquad (1)$$

Although, the recent trends in information fusion community have developed MoEs for fusion system performance evaluation [52–54], the above tracking MoP is a suitable metric for IF-SoS performance characterization based on the mission objective described in Section 5.1.1. Nonetheless, it is important to acknowledge that the general IF-SoS framework developed in this paper and the ensuing performance characterization analysis is fully extensible to include MoEs and other MoPs for different IF-SoS applications and mission objectives.

5.3. Design of experiments and results analysis

The existence of multitude of variable options for the IF-SoS design is evident from the preceding discussions. These variable design options, starting from the IF-SoS allocated architectures to the parametrization of functional and physical dynamics model, render an extensively vast design space for the IF-SoS. The evaluation challenge for the IF-SoS is to characterize how the variations in these design options (allocation strategies, functional dynamics, and physical dynamics) impact the end IF-SoS performance.

In this paper, we discuss Design of Experiments (DoE) for characterizing performance implications of different IF-SoS design variables which includes both the LLIF and the HLIF considerations. A few authors have explored the analytical potential of DoE for IFS performance evaluation, but only included limited LLIF design considerations in the experimental formulations, e.g., Sambhoos et. al. [55], Llinas et. al. [56], and Raz & DeLaurenits [51].

DoE provides a methodological approach for characterization of the IF-SoS design space, utilizing statistical analysis tools to identify the key decision variables. Introducing DoE, Rekab & Shaikh state that *"The objective of experimental design is to provide the researcher or a practitioner with a statistical method that determines which input variables are most influential on the output and where to set the influential input variables so that the output is either maximized, minimized, or nearest to a desired target value"* [57].

5.3.1. Experimental factors

One of the first steps in the DoE is to identify explanatory factors that are to be included in the experiment along with the different treatment levels of each factor [58]. From the IF-SoS perspective, the DoE factors are the IF-SoS design variables which constitute the various allocation strategies, functional dynamics, and physical dynamics. The factor treatment levels are the various design options considered for each variable. For example, the LLIF allocation strategy is one factor in DoE and the three implementation options (LL1, LL2, and LL3) shown in Fig. 8 are the three treatment levels of this factor. Similarly, when different fusion methods are available for implementing L1-TF functionality, the fusion method becomes a DoE factor, and the different algorithms its treatment levels. In addition to the LLIF and the HLIF allocation strategies, the factors for the DoE and their treatment levels can be derived from the bulleted items in Section 5.2.1. Table 5 summarizes the list of all factors considered in this experiment along with the treatment levels of each factor. It is worthwhile to note that all factors in this experiment are considered to be fixed categorical factors which implies that a given treatment level of any factor at any time remains exactly the same and no ordinal relationship with the factor's other treatment levels is assumed.

Table 5
DoE factors and treatment levels.

Factors	ID	L1	L2	L3
LLIF allocation stg.	X1	LL1	LL2	LL3
HLIF allocation stg.	X2	HL1	HL2	HL3
Pass. sen accuracy	X3	High	Med	Low
Act. sen 1 accuracy	X4	High	Med	Low
Act. sen 2 accuracy	X5	High	Med	Low
Act. sen rate	X6	High	Med	Low
Fusion method	X7	CI	CIM	ML
Sen. Tsk. Sol. Avail.	X8	High	Med	Low

5.3.2. Experimental design and data collection

Once the factors and their treatment levels are identified, the next steps in the DoE are to identify the underlying design for conducting the experiment and analyzing the resulting data. "The selection of an experimental design depends upon a number of different variables such as the total number of factors, treatments levels of each factor, available resources to conduct the experiment and the required inference regarding factors" [51]. In this paper, we use 3^k full factorial design (FFD) which represents a design space characterized by k factors evaluated at three levels each. In a FFD, all possible combinations of every factor at all treatment levels are evaluated, resulting in a total of 3^k experimental evaluations. For the purpose of IF-SoS performance evaluation, based on the experimental layout identified in Table 5, the FFD is given by $3^8 = 6561$ evaluations.

In DoE, a matrix called the experimental design matrix identifies all the different factor treatment level combinations which are to be included in the experiment. For 3^8 FFD, the experimental design matrix contains 6561 rows, each identifying a unique factor level combination for which response value of the experiment is to be collected. Hence, the experimental design matrix provides a complete set of data required for statistical analysis. Table 6 provides a snap shot of the experimental design matrix for the IF-SoS performance evaluation. The data collection process for populating the last column, i.e., response value Y, using the IF-SoS position error MoP is discussed in the following paragraph.

The factor level combination identified in the experimental design matrix corresponds to a specific setting of the IF-SoS allocated architecture, functional dynamics and physical dynamics. For each row of the experimental design matrix, an independent DAF simulation is instantiated based on the corresponding factor treatment levels to simulate the IF-SoS mission. Three detectable targets are simulated and the position error (Eq. 1) of each target is calculated for the mission duration. The IF-SoS MoP is then calculated as the mean of the median position error of the three targets. Each experimental condition is replicated three times with independent DAF instantiations to account for any variability in the IF-SoS MoP and all replicates are used for the statistical analysis. The entire data set which contains a total of $3^8 \times 3 = 19,683$ IF-SoS MoP values is used for IF-SoS statistical analysis.

5.3.3. Statistical analysis

Once the entire experiment is conducted and the response values for all 3^8 experimental conditions are obtained, the next step in the experimental design is to perform statistical analysis on the data. Statistical analysis identifies the significant factors and quantifies contribution of each factor level and its interaction with other factors on the response variable. In this paper, we use JMP[5] to analyze the data and characterize the impact of different factors on the IF-SoS performance.

Table 6
IF-SoS DoE experimental matrix.

Exp#	X1	X2	......	X8	Y(m)		
					Rep#1	Rep#2	Rep#3
1	L1	L1		L1	16.7408	15.7926	16.6429
2	L1	L1		L2	26.4720	26.7227	26.2188
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
.	.	.	.	.	.	.	.

Analysis of Variance (ANOVA). In DoE, the analysis of variance (ANOVA) identifies significance of effects of different factors on the response variable. ANOVA utilizes a regression model to fit the experimental data and establishes statistical significance of factors and their interactions. For conducting ANOVA of the IF-SoS experimental data, we utilize main effects plus two-way interactions linear regression model which is given by the following equation [59]:

$$Y_p = \beta_0 + \sum_{j=1}^{k} \beta_{jp} x_{jp} + \sum_{j=1}^{k-1} \sum_{l=j+1}^{k} \beta_{jp,lp} x_{jp} x_{lp} + \epsilon_p \tag{2}$$

In the above equation, all the β's are unknown parameters. β_0 is the overall mean of the response model, β_{jp} represents the main effect of factor X_j for the pth experimental condition and x_{jp} indicates the treatment level setting of factor X_j for pth experimental condition. $\beta_{jp,\,lp}$ represents the interaction effect due to two-way factor interaction of X_j and X_l for pth experimental condition. Y_p is the response value of the pth experimental condition and ϵ_p—residual error—is the error difference between the predicted value of response at pth experimental condition. The ANOVA model assumes that residual errors are normally distributed with zero mean and variance σ_ϵ^2, and ϵ_j is independent of ϵ_l for all j and $l \neq j$ [59].

The ANOVA model for IF-SoS statistical analysis is built using the log transform of response variable values ($\widetilde{Y} = \log Y$) in JMP's DoE utility to fit Eq. 2 and the residual error assumption are validated. The resulting ANOVA model has an R-square value of 0.83, and the model mean response, β_0, of 67.01m [6]. ANOVA utilizes hypothesis testing to establish statistical significance of each factor's main effect and the two-way factor interactions on the response variable. The null hypothesis tested by ANOVA is that varying treatment levels of a factor has no impact on the response variable. This hypothesis is tested using the F-test statistic. These F-tests are performed in JMP and are evaluated based on the p-values, where a p-value of less than 0.05 indicates statistical significance of variation in the mean response due to varying treatment levels (i.e., rejection of the null hypothesis). Table 7 provides the p-values of the main effects and the two-way factor interactions for IF-SoS performance evaluation.

It can be easily concluded from the p-values provided in Table 7 that the main effects of all factors and all most all interactions remain statistically significant for the response variable (i.e., the IF-SoS position error). Statistical significance of a factor main effect implies that different treatment levels of that factor have a significant impact on the response variable. A significant interaction implies that the effect of one factor treatment level on the response variable is dependent on another factor's treatment level. Similarly, an insignificant main effect means that the impact of different treatment levels of a factor on the response variable is insignificant. An insignificant interaction effect means the response variable impact of one factor treatment level is independent of the other factor treatment level.

From the IF-SoS design and evaluation perspective, the ANOVA outcome provides three major contributions: (1) the ANOVA iden-

[5] JMP is a statistical data analysis software (http://www.jmp.com).

[6] In this paper, all of the statistical investigations and tests are carried out on the transformed data. However, any resulting system-level analysis is discussed after de-transformation.

Table 7
IF-SoS ANOVA.

Fct.	p-val.	Fct.	p-val.
X1	<0.0001	X1*X6	<0.0001
X2	<0.0001	X2*X6	<0.0001
X3	<0.0001	X3*X6	<0.0001
X4	<0.0001	X4*X6	<0.0001
X5	<0.0001	X5*X6	<0.0001
X6	<0.0001	X1*X7	<0.0001
X7	<0.0001	X2*X7	<0.0001
X8	<0.0001	X3*X7	<0.0001
X1*X2	<0.0001	X4*X7	<0.0001
X1*X3	<0.0001	X5*X7	<0.0001
X2*X3	<0.0001	X6*X7	<0.0001
X1*X4	<0.0001	X1*X8	<0.0001
X2*X4	<0.0001	X2*X8	<0.0001
X3*X4	0.7199	X3*X8	<0.0001
X1*X5	0.0014	X4*X8	<0.0001
X2*X5	0.0005	X5*X8	<0.0001
X3*X5	0.1330	X6*X8	<0.0001
X4*X5	<0.0001	X7*X8	<0.0001

Table 8
Main effects Tukey HSD range tests.

	Lvls	IDs			LS means
X1 main effects	L3	A			83.06
	L1		B		61.10
	L2			C	59.29
X2 main effects	L3	A			77.31
	L2		B		67.76
	L1			C	57.43
X7 main effects	L3	A			85.80
	L1	A			85.44
	L2		B		41.04

*Levels not connected by same ID are significantly different.

Table 9
Interaction Tukey HSD range test.

X1*X2 interaction effect								
Lvls	ID							LS mean
L3,L2	A							107.61
L3,L3		B						84.39
L1,L3		B						82.08
L2,L3			C					66.71
L3,L1				D				63.12
L2,L1					E			56.67
L2,L2					E			55.12
L1,L1						F		52.96
L1,L2						F		52.48

*Levels not connected by the same ID are significantly different.

tifies important IF-SoS design considerations by establishing statistical significance of factor main and interactions effects, (2) the ANOVA identifies that IF-SoS design considerations cannot be evaluated in isolation due to the presence of significant interaction effects, and (3) the ANOVA could also reduces the potential IF-SoS design space by identifying insignificant main effects and/or interactions (shaded columns of Table 7 highlight the insignificant interactions for the example IF-SoS considered in this paper).

Once the significant main effects and interactions are identified, the next step is to quantify the impact of different treatment levels of all significant factors main effects as well as the significant interactions. This quantification is provided by conducting Range Tests which also establishes statistical significance between different treatment levels of an individual factor and its interactions with other factors.

Range tests. The range test on the experimental data is performed using the Tukey Honest Significant Difference (HSD) test in JMP. Tukey HSD provides a pair-wise comparison procedure to establish statistical difference between all treatment levels of factor main effects and interactions based on the least square mean estimates from the ANOVA model. For mathematical details of the Tukey HSD test, the interested reader is refereed to references [58,60].

The Tukey HSD test results for select main effects and one significant interaction are provided in Tables 8 and 9 respectively. In both tables any factor treatment levels that are not connected by the same alphabetical ID are determined by the Tukey HSD test to be statistically different at 95% confidence level with a standard error of 1.01m.

The main effects Tukey HSD range test results for factor X1 (LLIF allocation strategy), X2 (HLIF allocation strategy) and X7 (Fusion method) are provided in Table 8. The different alphabetical IDs for all the treatment levels of factor X1 and X2 imply that different LLIF and HLIF allocation strategies significantly impact the IF-SoS performance. Conversely, treatment levels L1 and L3 of factor X7 (Fusion Method) are connected by the same alphabetic ID indicating that there is no significant difference between the two levels (i.e., fusion methods CI and ML) on the IF-SoS position error. Since lower error is desirable for IF-SoS MoP, solely based on the main effects range tests results, a system designer maybe inclined to select treatment levels L2, L1, and L2 for factor X1, X2, and X6 respectively. However, the presence of significant interactions, as identified by ANOVA, advocates that main effects alone are not sufficient to reach system design conclusions.

The Tukey HSD range test results for the interaction between factor X1 and X2 (i.e., LLIF and HLIF allocation strategies respectively) are provided in Table 9. It can be clearly seen from Table 9 that different combinations of factor treatment levels of factor X1 and X2 have a different impact on the IF-SoS performance. The numerical ranges of the least square means values provided in Table 9 indicates that mean IF-SoS performance can vary between 107.61m to 52.48m depending upon the X1*X2 interaction. When the IF-SoS is designed with 'Distributed Fusion (LL1)' and 'Global Assessment & Local Tasking' (i.e., X1,X2 → L1,L2) the resulting IF-SoS mean position error is expected at 52.48m, and if the LLIF strategy is changed to 'Independent Fusion (LL3)' the IF-SoS mean position error increases to 107.61m (i.e., X1,X2 → L3,L2).

The above discussion provides a compelling statistical evidence for one of the key developments sought in this paper, i.e., the importance of an integrated design and evaluation of LLIF and HLIF. A closer examination of the values presented in Table 9 reveals crucial IF-SoS design and evaluation considerations.

First, the main effects alone are inadequate to determine the IF-SoS performance. If the IF-SoS were to be designed based on individual main effects and lowest IF-SoS position error is desired, the LLIF strategy will be selected as 'Hybrid Fusion' and HLIF strategy will be 'Global Awareness and Global Tasking', i.e., factor X1 and X2 will be treated at levels L2, and L1 respectively (lowest estimated LS mean from main effects Table 8). However, the interaction effects (Table 9) establishes that lowest errors are achieved when factor LLIF strategy is 'Distributed Fusion' and HLIF strategy is 'Global Assessment & Local Tasking', i.e., X1 and X2 are treated at level L1 and L2 respectively. The IF-SoS design suggested by main effects is only ranked the fourth best in the interaction effect Table 9.

Second, it is imperative to note that both the best and the worst IF-SoS performance are achieved for the same treatment level of the HLIF Strategy (i.e., factor X2 treatment at L2 which corresponds to the 'Global Assessment & Local Tasking'). This means that when the LLIF strategy changes, the IF-SoS performance can significantly deteriorate if corresponding changes to HLIF strategy are not con-

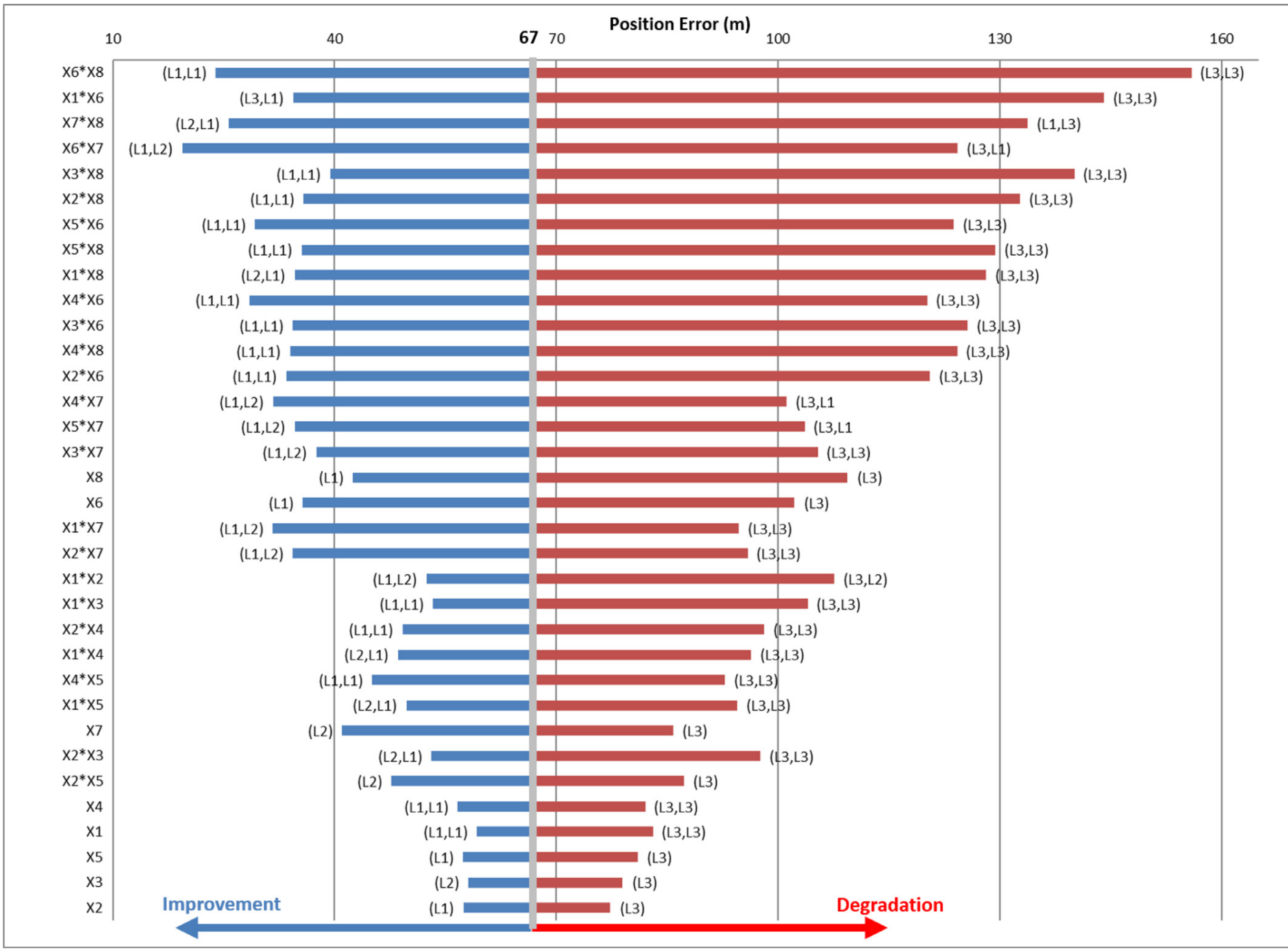


Fig. 11. IF-SoS sensitivity analysis.

currently reevaluated. Table 9 clearly shows that when factor X1 (i.e., LLIF strategy) changes from treatment level L1 to L3, the best IF-SoS performance is only achieved when the HLIF strategy (i.e., factor X2) is also changed from treatment level L2 to L1. From the IF-SoS design perspective, this concludes that design decisions for LLIF or HLIF strategies cannot be evaluated in isolation. Hence, establishing that an integrated LLIF and HLIF design and evaluation is required for the IF-SoS.

The above discussion highlights the importance of interactions for the IF-SoS design and evaluation. In 3^8 full factorial DoE, there are a total of 28 two-way interactions. The ANOVA (Table 7) has identified that 26 of these interactions are statistically significant for the IF-SoS position error. A significant interaction means that the impact on the IF-SoS position error due to one factor treatment level is dependent on the treatment level of another factor. Since these different factors are comprised of SoS allocation, functional dynamics, and physical dynamics parameters of different systems, this proves the claim that SoS considerations are imperative for the IFS design space characterization. Ideally, range tests for all factors and interactions should be thoroughly examined for comprehending the IF-SoS design space, however such an undertaking will not be possible within the scope of this paper. Instead, building upon the Tukey HSD range test results, we provide a sensitivity analysis in the form of a tornado chart for the IF-SoS design and evaluation. Using the ANOVA model's mean response of 67.01m for IF-SoS position error as a baseline value, Fig. 11 establishes the sensitivity of IF-SoS position error to the significant main effects and significant two-way interactions based on maximum and minimum Tukey HSD range test results. The blue bars in Fig. 11 depicts the maximum improvement (minimum position error) from the mean response based on identified treatment(s) of the corresponding factor(s). Conversely, the red bars quantifies the maximum expected degradation in performance when the same factor(s) are treated at the annotated level(s).

Fig. 11 contributes valuable insights for the design and evaluation of IF-SoS. Since the rows in a tornado chart are ranked from most to least sensitive, it can be concluded that the IF-SoS position error remains most sensitive to the interaction between X6 and X8. The red bars in Fig. 11 for X6*X8 indicates that IF-SoS position error can degrade to upto 155m when these factors are both factors treated at treatment level L3, whereas a different treatment level (i.e., L1) for these factors will reduce the IF-SoS position error to 20m (as shown by the blue bar). Difference between various trade offs due to different treatment level combinations of X6 and X8 can be identified by constructing a range test table similar to the one discussed earlier (Table 9). X6 is the measurement availability of active sensors, a LLIF Sen+L0 design consideration, and X8 is the sensor tasking solution availability, a HLIF L4-ST design consideration. Hence, Fig. 11 empirically proves that interactions between LLIF and HLIF cannot be overlooked when designing an IF-SoS.

Similarly, Fig. 11 establishes that the second most sensitive consideration for IF-SoS position error is the interaction between X1 and X6 (i.e., the LLIF allocation strategy and the active sensors measurement availability). Interestingly, however, is presence of same the treatment level (L3) for factor X1 at the ends of both blue and red bars. Factor X1 treatment level L3 corresponds to the independent fusion strategy illustrated in Fig. 8 and treatment levels L1 and L3 of factor X6 correspond to high and low measurement availability respectively. From the IF-SoS design perspective, this implies that the independent fusion strategy remains highly susceptible to measurement availability. When measurements are frequently available, high measurement availability rate, independent fusion strategy can provide more accurate position error as compared to any other LLIF allocation strategy. However, at the same time, the independent fusion strategy will not withstand a declining measurement availability as well as the other LLIF allocation strategies. Again, a detailed quantification of this susceptibility in relations to other LLIF allocation strategies and measurement availability rates can be thoroughly investigated from the expanded Tukey HSD range test table.

While each factor's sensitivity on the IF-SoS performance cannot be examined here, the ANOVA and Tukey HSD Range tests succinctly characterize the design space and illustrate the integrated treatment of LLIF and HLIF considerations.

6. Conclusion and future work

We apply a SoS engineering architecting process to obtain integrated architectures of IFS and propose guidelines to constrain an otherwise infinite design space of Information Fusion System-of-Systems (IF-SoS). In particular, the approach emphasizes understanding relations between low and high level IF (LLIF and HLIF) design decisions. Design of Experiments (DoE) methodology provides the necessary tools to characterize this design space and identify significant factors in the allocation of functional requirements to physical systems.

The DoE formulation utilized for IF-SoS design space evaluation is 3^8 FFD. This implies that all of the IF-SoS design considerations are encapsulated by 8 factors treated at 3 levels each. However, it is expected that the design space of an IF-SoS expands well beyond eight factors. Furthermore, it is also unlikely that treatment levels of all factors remain homogeneous. Utilizing FFD with increasing numbers of factors and varying number of treatment levels may not be feasible due to the sheer magnitude of data points required for statistical analysis. Application of different DoE techniques will be investigated in future that will allow simultaneous characterization of multiple performance metrics in an increasingly large and heterogeneous IF-SoS design space. Moreover, agent-based models for JDL L5 (user refinement) will be developed in future and multiple modes of user interactions with the IF-SoS, such as active user control, monitoring, and no control, will be included in the IF-SoS experimental design and evaluation.

Acknowledgments

This paper was developed under work supported by the US Missile Defense Agency (MDA) under contract No. HQ-147-10-C-6001. The MDA has reviewed and approved the original manuscript for public release [15-MDA-8471 (9 November 15)] and its subsequent revision [16-MDA-8745 (6 June 16)]. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the MDA. The US Missile Defense Agency does not endorse any products or commercial services mentioned in this publication.

The authors acknowledge contributions of Linas Mockus, Parth Shah, and Walter Schostak of Purdue University Center for Integrated Systems in Aerospace (CISA), and Bruce Craig and Nathan Hankey of Purdue University Statistical Consulting Service for reviews and feedback on this work.

References

[1] F.E. White, Data Fusion Lexicon, Techincal Report, Data Fusion Panel, Joint Directotrs of Laboratories, Technical Panel for C3, 1991.
[2] F. White, O. Kessler, Data Fusion Perspectives and Its Role in Information Processing, in: Handbook of Multisensor Data Fusion, CRC Press, 2008, pp. 15–43.
[3] D.L. Hall, J. Llinas, An introduction to multisensor data fusion, Proc. IEEE 85 (1) (1997) 6–23, doi:10.1109/5.554205.
[4] D.L. Hall, S.A.H. McMullen, Mathematical Techniques in Multisensor Data Fusion, Artech House, 2004.
[5] J. Llinas, M.E. Liggins, D.L. Hall, Handbook of Multisensor Data Fusion: Theory and Practice, CRC Press, 2008.
[6] D.L. Hall, C.-Y. Chong, J. Llinas, M. Li, Distributed Data Fusion for Network-Centric Operations, CRC Press, 2012.
[7] E.L. Waltz, J. Llinas, Multisensor Data Fusion, Artech House, Inc., Norwood, MA, USA, 1990.
[8] W. Koch, L. Snidaro, K. Rein, J. Garcia, S. Andler, G. Wah Ng, Pannel Discussion - Multi-Level Fusion: Issues in bridiging the gap between high and low level fusion, in: 15th Conference on Information Fusion (FUSION), Singapore, 2012.
[9] E. Blasch, E. Bossé, D. Lambert, High-Level Information Fusion Management and System Design, 1, Artech House, 2012.
[10] P.H. Foo, W. Gee, High-level information fusion: an overview, J. Adv. Inf. Fusion 8 (1) (2013) 33–71.
[11] E. Blasch, J. Llinas, D. Lambert, P. Valin, S. Das, C. Chong, M. Kokar, E. Shahbazian, High Level Information Fusion developments, issues, and grand challenges: Fusion 2010 panel discussion, in: 13th Conference on Information Fusion (FUSION), 2010, pp. 1–8.
[12] M. Bedworth, J. O'Brien, The omnibus model: a new model of data fusion? IEEE Aerosp. Electron. Syst. Mag. 15 (4) (2000) 30–36, doi:10.1109/62.839632.
[13] A.N. Steinberg, C.L. Bowman, Systems Engineering Approach for Implementing Data Fusion Systems, in: Handbook of Multisensor Data Fusion, CRC Press, 2008, pp. 561–596.
[14] D.A. Lambert, A blueprint for higher-level fusion systems, Inf. Fusion 10 (1) (2009) 6–24, doi:10.1016/j.inffus.2008.05.007.
[15] M.A. Solano, S. Ekwaro-Osire, M.M. Tanik, High-level fusion for intelligence applications using recombinant cognition synthesis, Inf. Fusion 13 (1) (2012) 79–98, doi:10.1016/j.inffus.2010.08.002.
[16] B. Solaiman, Boss, L. Pigeon, D. Guriot, M.C. Florea, A conceptual definition of a holonic processing framework to support the design of information fusion systems, Inf. Fusion 21 (2015) 85–99, doi:10.1016/j.inffus.2013.08.004.
[17] E.P. Blasch, D.A. Lambert, P. Valin, M.M. Kokar, J. Llinas, S. Das, C. Chong, E. Shahbazian, High level information fusion (HLIF): survey of models, issues, and grand challenges, IEEE Aerosp. Electron. Syst. Mag. 27 (9) (2012) 4–20, doi:10.1109/MAES.2012.6366088.
[18] M. Solano, G. Jernigan, Enterprise data architecture principles for High-Level Multi-Int fusion: A pragmatic guide for implementing a heterogeneous data exploitation framework, in: 2012 15th International Conference on Information Fusion (FUSION), 2012, pp. 867–874.
[19] E. Blasch, A. Steinberg, S. Das, J. Llinas, C. Chong, O. Kessler, E. Waltz, F. White, Revisiting the JDL model for information exploitation, in: 2013 16th International Conference on Information Fusion (FUSION), 2013, pp. 129–136.
[20] J. Roy, S. Wark, E. Bossé, Computational Aspects of Information Fusion, Concepts, Models, and Tools for Information Fusion, Artech House, Norwood, MA, USA, 2007.
[21] E.L. Waltz, D.L. Hall, Requirements Derivation for Data Fusion Systems, in: Handbook of Multisensor Data Fusion, CRC Press, 2008, pp. 549–560.
[22] M.A. Solano, J. Carbone, Systems Engineering for Information Fusion: Towards Enterprise Multi-Level Fusion Integration, in: 2013 16th International Conference on Information Fusion (FUSION), 2013, pp. 121–128.
[23] C.G. Pernin, E. Axelband, J.A. Drezner, B.B. Dille, J. Gordon, B.J. Held, K.S. McMahon, W.L. Perry, C. Rizzi, A.R. Shah, P.A. Wilson, J.M. Sollinger, Lessons from the Army's Future Combat Systems Program, 2012,
[24] M. Jamshidi, Systems of Systems Engineering: Principles and Applications, CRC Press, 2010.
[25] M. Jamshidi, Introduction to System of Systems, System of Systems Engineering, John Wiley & Sons, Inc., 2008.
[26] A.H. Levis, L.W. Wagenhals, C4isr Architectures: i. developing a process for c4isr architecture design, Syst. Eng. 3 (4) (2000) 225–247 doi:10.1002/1520-6858(2000)3:4⟨225::AID-SYS4⟩3.0.CO;2-#.
[27] D.M. Buede, The Engineering Design of Systems Models and Methods, 2nd ed.., John Wiley & Sons, Hoboken, NJ, 2009.
[28] OlegV. Sindiy, Model-based System-of-systems Engineering for Space-based Command, Control, Communication, and Information Architecture Design., Thesis PhD–Purdue University, 2010.
[29] R.C. Kenley, T.M. Dannenhoffer, P.C. Wood, D.A. DeLaurentis, Synthesizing and Specifying Architectures for System of Systems, 24th Annual INCOSE International Symposium, Las Vegas, NV, 2014.
[30] M.A. Solano, SoSE architecture principles for Net-Centric Multi-Int Fusion Systems, in: 2011 6th International Conference on System of Systems Engineering (SoSE), 2011, pp. 61–66, doi:10.1109/SYSOSE.2011.5966574.
[31] F.I.P.S.F.P.N. 183., Integration Definition for Function Modeling (IDEF0),, 1993,
[32] A.N. Steinberg, L. Snidaro, Levels? in: 2015 18th International Conference on Information Fusion (Fusion), 2015, pp. 1985–1992.
[33] E. Blasch, J. Garcia Herrero, L. Snidaro, J. Llinas, G. Seetharaman, K. Palaniappan, Overview of contextual tracking approaches in information fusion, Geospatial InfoFusion III, 8747. pp. 87470B–87470B–11, SPIE, 2013, doi:10.1117/12.2016312.

[34] L. Snidaro, J. Garcia, J. Llinas, E. Blasch, Context-Enhanced Information Fusion: Boosting Real-World Performance with Domain Knowledge, Advances in Computer Vision and Pattern Recognition, Springer International Publishing, 2016.
[35] E. Blasch, S. Plano, DFIG Level 5 (User Refinement) issues supporting Situational Assessment Reasoning, in: 2005 8th International Conference on Information Fusion, 1, 2005, pp. xxxv–xliii, doi:10.1109/ICIF.2005.1591828.
[36] M.W. Maier, Architecting principles for systems-of-systems, Syst. Eng. 1 (4) (1998) 267–284, doi:10.1002/(SICI)1520-6858(1998)1:4⟨267::AID-SYS3⟩3.0.CO;2-D.
[37] D.A. DeLaurentis, R.K. Callaway, A system-of-systems perspective for public policy decisions, Rev. Policy Res. 21 (6) (2004) 829–837.
[38] P. Bianco, R. Kotermanski, P. Merson, Evaluating a Service-Oriented Architecture, Techincal Report, Software Engineering Institute - Carnegie Mellon University, 2007.
[39] J.S. Dahmann, J. Rebovich, J.A. Lane, Systems engineering for capabilities, CrossTalk The Journal of Defense Software Engineering 21 (11) (2008) 4–9.
[40] D.A. DeLaurentis, W.A. Crossley, M. Mane, Taxonomy to guide systems-of-systems decision-making in air transportation problems, J. Aircr 48 (3) (2011) 760–770.
[41] A.K. Raz, D.A. DeLaurentis, A System-of-Systems Perspective on Information Fusion Systems: Architecture Representation and Evaluation, AIAA Infotech @ Aerospace, AIAA SciTech, American Institute of Aeronautics and Astronautics, 2015, doi:10.2514/6.2015-0644.
[42] E. Bonabeau, Agent-based modeling: methods and techniques for simulating human systems, Proceedings of the National Academy of Sciences 99 (suppl 3) (2002) 7280–7287, doi:10.1073/pnas.082080899.
[43] C. Joslyn, L. Rocha, Towards semiotic agent-based models of socio-technical organizations, in: Proc. AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, Tucson, Arizona, 2000, pp. 70–79.
[44] M. Mane, D.A. DeLaurentis, Sensor Platform Management Strategies in a Multi-Threat Environment, Infotech@Aerospace, Garden Grove, CA, 2012.
[45] A. Mour, R.C. Kenley, N. Davendralingam, D.A. DeLaurentis, Agent-Based Modeling for Systems of Systems, in: INCOSE International Symposium, 23, Philadelphia, PA, 2013, pp. 973–987, doi:10.1002/j.2334-5837.2013.tb03067.x.
[46] D.N. Fry, R. Campbell, D.A. DeLaurentis, Modeling Systems-of-Systems from Multiple Design Perspectives: Agents, Interfaces, and Architectures, in: AIAA Modeling and Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, Kissimmee, FL, 2015.
[47] Y. Bar-Shalom, P.K. Willett, X. Tian, Tracking and Data Fusion, YBS Publishing, Storrs, CT USA, 2011.
[48] C.-Y. Chong, S. Mori, W. Barker, K.-C. Chang, Architectures and algorithms for track association and fusion, IEEE Aerosp. Electron. Syst. Mag. 15 (1) (2000) 5–13, doi:10.1109/62.821657.
[49] S.S. Blackman, R.F. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, 1999.
[50] S.J. Julier, J.K. Uhlmann, General Decentralized Data Fusion with Covariance Intersection (CI), Handbook of Multisensor Data Fusion, CRC Press, 2001.
[51] A.K. Raz, D.A. DeLaurentis, Performance evaluation of distributed Track-to-Track fusion systems, in: 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), San Diego, CA, 2014, pp. 1585–1590, doi:10.1109/SMC.2014.6974142.
[52] J. Llinas, Assessing the Performance of Multisensor Fusion Processes, in: Handbook of Multisensor Data Fusion, CRC Press, 2008, pp. 655–675.
[53] E. Blasch, E. Bossé, D. Lambert, Measures of Effectiveness for High-Level Information Fusion, Information Fusion Management and Systems Design, Artech House, Norwood, MA, USA, 2012.
[54] E. Blasch, P. Valin, E. Bossé, Measures of effectiveness for high-level fusion, in: 2010 13th Conference on Information Fusion (FUSION), 2010, pp. 1–8, doi:10.1109/ICIF.2010.5711858.
[55] K. Sambhoos, C. Bowman, J. Llinas, A case study with design of experiments: performance evaluation methodology for level 1 distributed data fusion processes, Information Fusion 12 (2) (2011) 93–104, doi:10.1016/j.inffus.2010.03.003.
[56] J. Llinas, C.L. Bowman, K. Sambhoos, Test and Evaluation of Distributed Data and Information Fusion Systems and Processes, in: Distributed Data Fusion for Network-Centric Operations, CRC Press, 2012, pp. 379–408.
[57] K. Rekab, M. Shaikh, Statistical Design of Experiments with Engineering Applications, Statistics: A Series of Textbooks and Monographs, 20055246, CRC Press, 2005.
[58] M.H. Kutner, C.J. Nachtsheim, J. Neter, W. Li, Applied Linear Statistical Models, 5th, McGraw-Hill Irwin, 2005.
[59] P. Goos, B. Jones, An Optimal Screening Experiment, in: Optimal Design of Experiments, John Wiley & Sons, Ltd, 2011, pp. 9–45.
[60] D.C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, Hoboken, NJ, 2005.