2023 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
AUTONOMY, ARTIFICIAL INTELLIGENCE & ROBOTICS (AAIR) TECHNICAL SESSION
AUGUST 15-17, 2023 - NOVI, MICHIGAN

EXTRACTING ACTIONABLE INFORMATION FROM HETEROGENEOUS SENSORS IN THE FIELD: A DISTRIBUTED HYBRID AI APPROACH IN CONSTRAINED DOMAINS

Gregor Pavlin[1], Raphaël Boudreault[2], Ate Penders[1], Maurits de Graaf[1], Daniel Lafond[2], Andy Swiebel[1]

[1]Thales Nederland B.V., The Netherlands
[2]Thales Digital Solutions, Québec, Canada

ABSTRACT

We present a modular architecture that enables advanced surveillance functions exploiting data collected from heterogeneous sensors dispersed over multiple, often mobile platforms in the field. Examples of such functions are red forces tracking with surveillance gaps, detection of different types of anomalies, search and rescue operation monitoring, and threat alerting. This novel approach combines a distributed fusion engine, an intelligent process manager, and a system of ruggedized computers, enabling information processing in the tactical domain. The hybrid AI-based heterogeneous fusion engine consists of different algorithms, including various detectors and classifiers, represented as services in a light-weight information management and interoperability layer. This architecture layer enables context-dependent discovery of the right sensing and processing services at runtime that are combined using a robust Bayesian fusion layer exploiting complex correlations in the data. The discovered services are distributed over a network of computing nodes by an intelligent process manager, which optimizes network resource allocation according to communication and processing capacities. The fusion engine and the process manager are delivered to the tactical domain using the ruggedized SOTAS computing and communication infrastructure, achieving efficient, actionable, timely, and consistent situation awareness in constrained domains, such as military vehicles.

Citation: G. Pavlin, R. Boudreault, A. Penders, M. de Graaf, D. Lafond, A. Swiebel, "Extracting Actionable Information From Heterogeneous Sensors in the Field: A Distributed Hybrid AI Approach in Constrained Domains," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 15-17, 2023.

1. INTRODUCTION

The number of sensors on the battlefield is strongly increasing [1–3]. To avoid information overload, an approach is needed to fuse information from heterogeneous sensors in order to get actionable, timely, and consistent situational awareness (SA) in constrained domains, such as military vehicles. The great independence, computing power and storage capabilities of modern military vehicles offer the possibility to process raw information streams and exchange them over limited bandwidth. Thus, in a "mix of means approach" in the mobile domain, there are network nodes with very different capabilities and limitations. They may or may not have specialized computational capacities or strong communication potential. In order to fuse information giving a better SA, and to optimally use the limited resources in the field, three key challenges need to be addressed:

(1) distributed fusion of information from heterogeneous sources;

(2) computational and communication constraints in network resources allocation;

(3) platform supporting C4I (command, control, communications, computers, and intelligence) in constrained environments.

This paper introduces a modular architecture that addresses these challenges. It enables advanced surveillance functions exploiting data collected from heterogeneous sensors dispersed over multiple, mobile platforms in the field. Examples of such functions are red forces tracking with surveillance gaps, detection of different types of anomalies, search and rescue operation monitoring, and threat alerting. This novel approach combines a distributed fusion engine, an intelligent process manager, and a system of ruggedized computers, supported by mechanisms to dynamically allocate fusion functions to processing nodes, without user intervention, to optimally use network resources for information processing in the mobile tactical domain.

The hybrid AI-based heterogeneous fusion engine introduced in Section 2 consists of different algorithms, including various detectors and classifiers, represented as services in a light-weight information management and interoperability layer. This layer enables context-dependent discovery of the right sensing and processing services at runtime, which are combined using a robust Bayesian fusion layer into composite fusion functions to exploit complex correlations in the data.

The discovered services are distributed over a network of computing nodes by an intelligent process manager introduced in Section 3. It aims at optimizing network resource allocation according to communication and processing capacities. The manager addresses a specific combinatorial, multi-objective and constrained optimization problem in order to determine an efficient and reliable fusion network topology. Different evaluation metrics and means of implementation are discussed therein.

The fusion engine and the process manager are delivered to the tactical domain using the ruggedized SOTAS computing and communication infrastructure presented in Section 4. This is a multimedia vehicle communication system developed by Thales Nederland providing a reliable, proven, interoperable and secure data and communication infrastructure for information sharing and communication inside constrained platforms, such as military vehicles and shelters [4].

1.1. Motivating Example

Figure 1 presents a simplified red forces tracking example with multiple sensors: an advanced counter-battery radar (COBRA) on vehicle `v1`; two simple acoustic sensors that can detect and recognize sound of heavy vehicles, installed on vehicles `v2` and `v4`; a dismounted Doppler radar next to vehicle `v2`; and a drone with a camera, a stationary camera, and a seismic sensor in the area of vehicle `v3`.

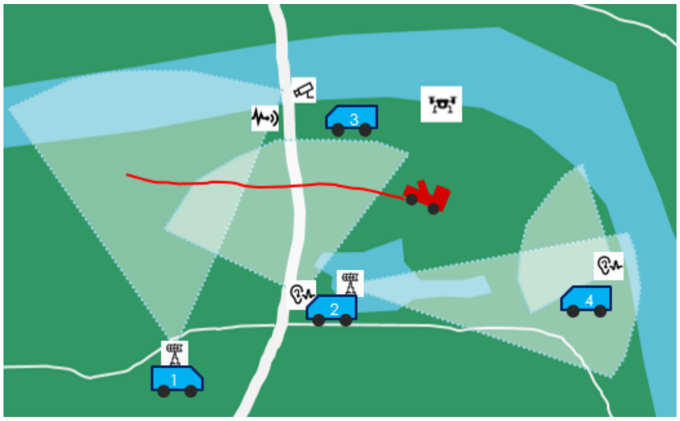

Figure 1: Representation of the red forces tracking example. The red vehicle corresponds to red forces moving through a partially observed area. Different types of sensors cover parts of the area and there exist significant surveillance gaps.

The tactical task in this example is to track the position of red forces, a heavy mobile mortar. Initially, the red forces fire with the mortar, which is detected and localized by the COBRA sensor. However, after a short firing period, the red forces move through an area that is not within the field of view of any available sensor. During this phase, the possible whereabouts of the red forces have to be estimated by combining different types of context information, such as the mobility in the area (e.g. water, solid ground, swamp, etc.) as well as sensor observations and detections, or the lack thereof, in different parts of the area of interest. In our example depicted in Figure 1, it could be inferred that, after losing the initial contact, the red forces should be within an area surrounded by the water and a zone monitored by sensors that do not observe any moving target. Such conclusions, however, could only be made if the correlations between heterogeneous sensor signals and the properties of the environment are understood. The data must also be communicated to the right processing node, on time, and under typical operational constraints (physically distributed assets, limited computation and communication capacity, misaligned sensors, etc.). Section 5 illustrates how this can be achieved for this example using our integrated architecture.

2. HETEROGENEOUS FUSION ENGINE

The *Heterogeneous Fusion Engine* enables extraction of actionable information from complex combinations of data produced by heterogeneous, spatially distributed sources. Such fusion is an enabler of advanced decision support functions, such as red forces tracking, detection of different types of anomalies (pattern-of-life), search and rescue operation monitoring, and threat alerting. The Heterogeneous Fusion Engine combines a modular high-level fusion approach (Section 2.1) with a light weight interoperability and information management layer (Section 2.2). These techniques enable loosely coupled modular fusion solutions that can be distributed over multiple processing nodes and exploit different types of algorithms. The approach has proven its merits in different real world settings, such as intrusion alerting and counter drone applications using multiple, heterogeneous sensors.

2.1. High-Level Fusion Based on Hybrid AI

Many types of decision support functions rely on the estimation of possible whereabouts of an object over time. By using the methods discussed in [5], such estimation can be formulated as the computation of a probability distribution $p(x_t|z_{1:t}, \epsilon^m_{1:t}, \epsilon^s_{1:t})$. Here, x_t denotes the target's location at time t and $z_{1:t}$ represents a sequence of sensor observations collected up to time t. Moreover, $\epsilon^m_{1:t}$ and $\epsilon^s_{1:t}$ denote sequences of sets of reports about the target's mobility and sets of reports about the factors influencing the sensor/detector outputs collected up to time t, respectively. An observation z_t can represent different types of measurements as well as detections or classifications at time t. Key to efficient computation of $p(x_t|z_{1:t}, \epsilon^m_{1:t}, \epsilon^s_{1:t})$ is recursive Bayesian inference:

$$\begin{aligned} &p(x_t|z_{1:t}, \epsilon^m_{1:t}, \epsilon^s_{1:t}) = \\ &\eta p(z_t|x_t, \epsilon^s_t) p(m_t = \mathrm{true}|\epsilon^m_t) \\ &\cdot \int p(x_t|x_{t-1}) p(x_{t-1}|z_{1:t-1}, \epsilon^m_{1:t-1}, \epsilon^s_{1:t-1}) dx_{t-1}, \end{aligned} \tag{1}$$

where η is a normalizing constant, $p(x_t|x_{t-1})$ is the dynamic model of the target in case of full mobility, while factors $p(m_t = \mathrm{true}|\epsilon^m_t)$ and $p(z_t|x_t, \epsilon^s_t)$ represent uncertain knowledge about the target's mobility and operational conditions influencing sensors, respectively. $\epsilon^s_t \subseteq \epsilon^s_{1:t}$ denotes a set of reports about the operational conditions that influence a specific sensor/detector and were known at time t, when this source produced output z_t. Similarly, $\epsilon^m_t \subseteq \epsilon^m_{1:t}$ represents a set of reports about the target's mobility m_t known at time t. Equation (1) is a version of the Context-Boosted Particle Filter [6], an approximate inference approach combining sequences of sensor observations with information about the physical limitations of the target. It is a factorized representation, where each factor can be obtained independently, some of them being a result of elaborate processing. In the presented approach, the determination of $p(z_t|x_t, \epsilon^s_t)$ is based on causal Bayesian Networks (BN) that efficiently capture complex correlations between the observations and various influences on the sensor, such as weather, time of the day, season, etc. An example Bayesian model for a Micro-Doppler detector is shown in Figure 2, where w, r and e denote respectively the wind conditions, precipitation (rain, hail, snow), and the presence of moving reflecting surfaces within the sensor's field of view, such as tree leaves. w and r influence the detection rates, denoted by d, and the noise, denoted by n, resulting in false detections. n is also influenced by e. For example, in case of strong winds, the movement of tree leaves within the sensor's field of view could be a source of increased false positive rates. The graph shown in Figure 2 encodes a factorization of a joint probability distribution over the set of variables:

$$\begin{aligned} &p(z_t, x_t, r, w, e, d, n) = p(r)p(w)p(e)p(x_t) \\ &\cdot p(z_t|d, n)p(d|x_t, r, w)p(n|r, w, e) \end{aligned} \tag{2}$$

This factorization enables efficient computation of $p(z_t|x_t, \epsilon^s_t = \{r, w, e\})$, essentially a marginalization of unobserved variables d and n in $p(z_t, x_t, r, w, e, d, n)$. Such models support local fusion of the information ϵ^s_t about the sensor's environment to determine the impact $p(z_t|x_t, \epsilon^s_t)$ the observation z_t has on the overall fusion described by equation (1).

The presented Bayesian sensor models can be used as wrappers of different sensors/detectors, facilitating combination of multiple heterogeneous components into coherent systems that can distill actionable information from disparate data. Namely, outputs of different types of sensors, detectors, and algorithms are efficiently translated to probability distributions $p(z_t|x_t, \epsilon^s_t)$, which in turn facilitate composition of complex Bayesian inference processes according to equation (1). For example, drones equipped with Electro-Optical (EO) sensors in combination with deep neural networks (DNN) detecting people or vehicles can be an excellent source of inputs to a larger fusion task, such as red forces tracking. The neural networks can be placed on the drone or in a vehicle to which the imagery/video is streamed. The detection results z_t are processed by a local BN that outputs $p(z_t|x_t, \epsilon^s_t)$, where ϵ^s_t represents the drone's context, such as visibility, wind, distance and angle of observation. This is an example of a relevant multistage integration pattern illustrated in Figure 3. Raw sensor signals are first interpreted by a specific algorithm, such as a data association or tracking algorithm, various types of detectors and classifiers based on neural networks, etc., represented by green boxes in Figure 3. The outputs of these processes are used by a Bayesian sensor model (light blue boxes) that translates specific outputs of the local signal processing to "fuseable" messages in form of probability distributions $p(z_t|x_t, \epsilon_t^s)$.

Moreover, the probability $p(m_t = \text{true}|\epsilon_t^m)$ representing the chance that a specific type of target can move in a certain area is another factor that can be computed in an independent process [5]. This factor is obtained through the fusion of different types of information about the mobility stemming from geographic information systems (GIS), intelligence collected by people in the field, as well as automated processing of satellite imagery.

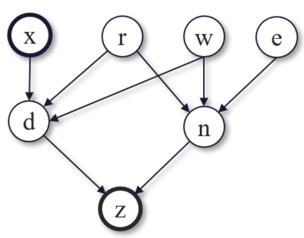

Figure 2: Bayesian sensor model.

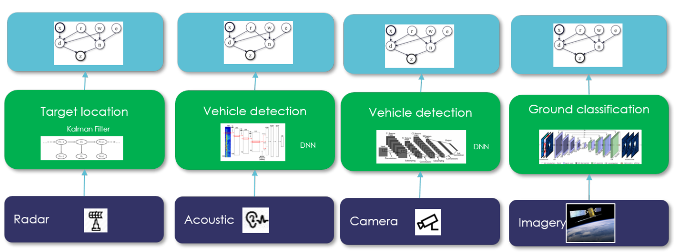

Figure 3: Examples of mappings between sensor signals and interoperable messages in the form of probability distributions.

Overall, the factorization of equation (1) and the presented Bayesian sensor or detector models are key to powerful hybrid AI solutions obtained through combination of different types of sensors and algorithms. The advantage of this approach is that such composite solutions can handle complex data patterns efficiently and support dynamic inclusion of new data sources or processing assets at runtime. For example, if a new drone with an EO sensor enters the area of interest, its outputs are automatically processed by a dedicated DNN and a local BN sensor model, transforming the sensor signals into a factor $p(z_t|x_t, \epsilon_t^s)$ that can be fused at a higher level, by simply including it in equation (1).

2.2. Fusion Engine Architecture

The key to distilling actionable intelligence, that is hidden in complex data patterns, are systems of functions that filter and interpret combinations of correlated data. By composing different factors from equation (1), the raw data is gradually transformed into actionable intelligence. Figure 4 shows an example of such a composite fusion system, a combination of computing services dynamically formed at runtime, implementing inference according to equation (1).

Interoperable functions and sensors. A composite fusion system consists of multiple functions ranging from simple filters to fusion functions computing factors in equation (1). These functions have to interoperate such that the output of a single function can be used as an input for one or more other functions. This requires an integration backbone allowing (i) composition of functions, (ii) activation of these functions on demand, and (iii) dynamic formation of information flows between them.

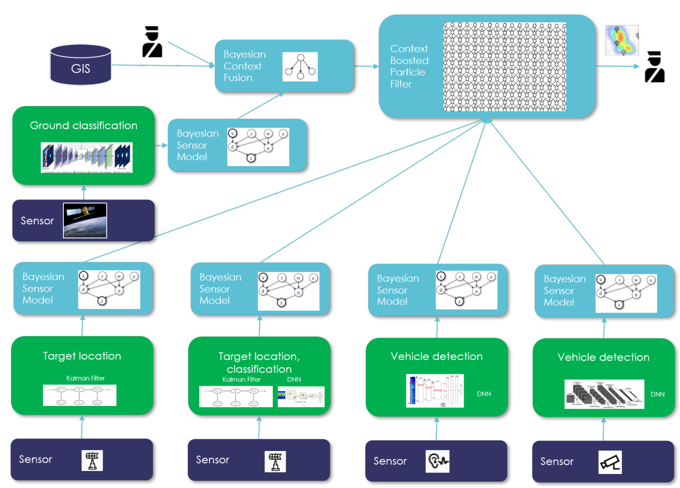

Figure 4: A composite fusion system.

Moreover, all functions and data sources (sensors, databases) should be represented as interoperable services that can be combined into composite solutions. Dynamic Process Integration Framework (DPIF) [7] supports the implementation of integration backbones addressing the above-mentioned requirements. The DPIF architecture introduces a thin "standardization" layer, that enables interoperability between different services. This is achieved in a modular way by using a novel way of representing services, with tools and patterns to enable description of these services (metadata on capabilities and context) [8].

Service composition and information flows. The DPIF architecture supports dynamic activation of suitable services in the context and creation of information flows between them. By specifying the need for a certain type of information and the context (e.g., area of interest), the right services are automatically combined. The resulting system of cascaded services is a composite function that includes all data sources and functions relevant for the task at hand; An example of such a composite function is red forces tracking.

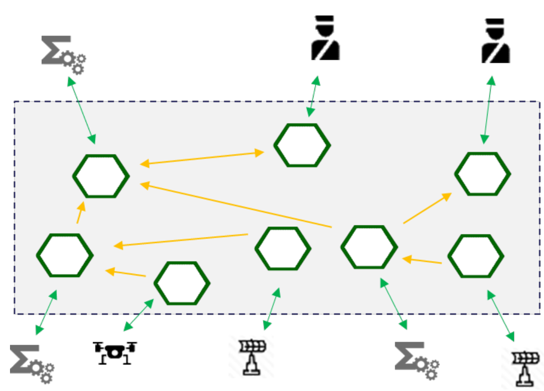

Figure 5: DPIF agents (hexagons) representing various services. The yellow arrows denote remote communication between the services. The agents form a lightweight interoperability and information management layer (represented by the gray rectangle).

Architecture and deployment characteristics. The basic architectural elements are service agents, engines that efficiently capture the incoming and outgoing interface adapters (metadata that describes a service, its inputs, and its context), see Figure 5. The interface adapters introduce interoperability and manage information flows supporting fast extension of the platform with new functions, data sources, and interfaces.

From an operational point of view, DPIF-based solutions are distinguished from other information distribution approaches by their scalability and dynamic character. Sensor and function invocation is determined on-the-fly, based on the context, which allows dynamic joining and leaving of sensors and functions during the lifetime of the system. As DPIF supports dynamic creation of information flows, such services are automatically made available and become part of an already running fusion process, if relevant (hot plugin). This enables a new way of working, where a crew member, by indicating an area on a screen, gets the fused information from all (authorized) sensors in the network that have information about that area.

Moreover, the architecture is designed to support deployment over a network of computing nodes. This is important, as the services often belong to different owners with stringent security measures. In such cases, the service can be hosted locally to ensure data security, while the agent exposes only the agreed types of standardized information with the rest of the system. In this way, the data owner controls when and under what conditions service data is shared. Moreover, the owner has control over the service visibility in the service discovery processes (runtime) as well as in the configuration process (design time).

The functions connect using service types [8], defining (i) services that functions provide to others in the network and (ii) services these functions need to operate. Figure 6 presents a logical view of functions and sensors that connect with services (the colored circles). Note that, the functions and sensors exchange information with the services they need directly, using (logical) peer-to-peer communication lines. For example, in the figure, the top function provides the service of type A and requires information from services of type B, C, and D, respectively. The hexagons in the figure represent agents that (1) enable dynamic configuration of the services, (2) wrap the function or interface to a remote process and (3) maintain the interaction between services and function.

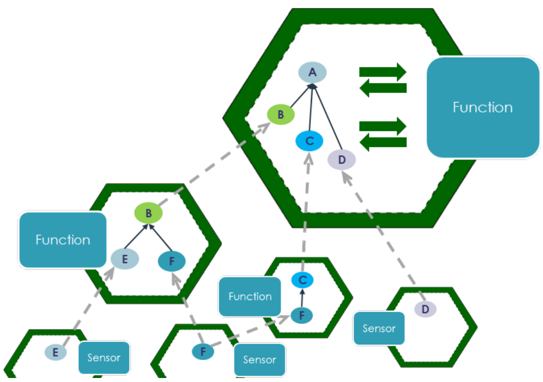

Figure 6: Logical view of DPIF agents connected in a fusion system, where the colored circles represent services provided/required by the functions and sensors.

Wrapping the functions in DPIF service agents allows for isolating the deployment of the functions, such that they can be distributed. Another added value of the structure with direct communication lines is the decentralized communication pattern (i.e., services locally organize the communication links and do not require a central orchestration). A consequence of having a decentralized communication pattern is the need for a discovery mechanism: consumers of service data need to find the potential suppliers. DPIF uses a discovery mechanism in the form of a lookup table, which works as a catalogue of potential service providers, the contact information of the potential service provider is then used to negotiate a direct communication link. For example, in Figure 6, the top node is interested in contact information for the services of types A, B and C in a certain context. Once it obtains their details, a communication link can be established and the functions or sensors are able to exchange information. This mechanism of looking up service providers to establish communication links is a critical point for the system to work. However, a function is actively looking for data providers only during the initiation of the information flow and this requires little bandwidth since only agent addresses are exchanged.

3. INTELLIGENT PROCESS MANAGER

The choice of a network topology in the context of data fusion from multiple sensors can have an important impact on the network's efficiency and reliability. In a centralized architecture, all data from the sensors is sent to a central processing node where the fusion happens and the results are then sent to the users. The advantages of a centralized topology are the fact that this is technically relatively simple to realize, and that the facilities (operating conditions) at the central location can be optimized. These facilities can be related to (perimeter) security, protection against attacks and conditions to operate (ventilation, energy). The main disadvantage of a centralized topology is, however, the fact that it has a single point of failure: the center of the network. On the other hand, a distributed architecture includes more than one processing node in the network, and fusion thus happens in more than one place before reaching the users. According to Liggins *et al.* [9], this type of architecture includes, but is not limited to the following advantages:

> [. . .] lighter processing load at each fusion node due to the distribution over multiple nodes; no need to maintain a large centralized database since each node has its own local database; lower communication load since data does not have to be sent

> to/from a central processing site; faster user access to fusion results since there is less communication delay; and higher survivability since there is no single point of failure associated with a central fusion node. (pp. 95-96)

However, a distributed system also comes with a considerable technical complexity in network architecture, communication links, and fusion algorithms.

3.1. Problem Definition

In our approach, the intelligent process manager solves the problem of selecting the best fusion network topology based on information flow and physical network constraints. This selection includes deciding the communication flow within the network, while also choosing how to allocate the required fusion functions to the processing nodes. The topology must also satisfy the information requests of end users.

Formally, this problem starts with a given (directed) network, $\mathcal{N} = (N, L)$. The set N contains the *nodes* of the network, and $L \subset N \times N$ forms the set of *links*. For two distinct nodes $i, j \in N$, we have $(i, j) \in L$ if there exists a communication link from node i to node j. We distinguish three types of nodes in the network:

a) *Sensor nodes*, nodes that can produce, and transmit data into the network, typically sensors. They form the subset $N_S \subseteq N$;

b) *Processing nodes*, nodes that can receive, process and transmit data into the network, typically computers on which the fusion algorithms can be executed. They form the subset $N_P \subseteq N$;

c) *Human-Machine Interface (HMI) nodes*, nodes that can receive fused data from the network, typically representing users. They form the subset $N_H \subseteq N$.

Figure 7 is an example network $\mathcal{N} = (N, L)$ composed of six nodes, $N = \{n_1, n_2, n_3, n_4, n_5, n_6\}$, two of which are sensors ($N_S = \{n_1, n_2\}$), three are processing nodes ($N_P = \{n_3, n_4, n_5\}$), and one is an HMI ($N_H = \{n_6\}$). In this representation that we call the *network graph*, an arrow from node n_i to node n_j means that there exists a communication link from n_i to n_j, i.e. $(n_i, n_j) \in L$.

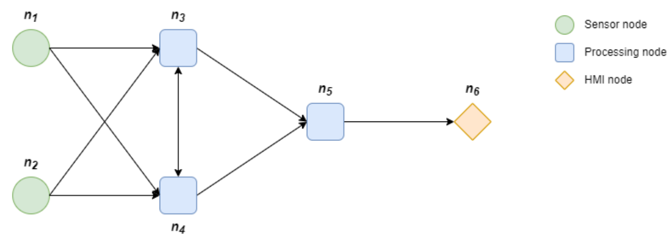

Figure 7: Network graph example.

Data fusion, as seen by DPIF, is composed of independent fusion functions and information flows. In fact, each fusion function combines a set of distinct information types into another set of distinct information types. For example, a filtering function could be transforming a "raw sensor 1" data to a "filtered sensor 1" data, and a particle filter transforming "filtered sensor 1" and "filtered sensor 2" data into "combined sensors" data. Thus, in order to reach a particular information (fused) type, data must go through a number of fusion functions in a particular order, following a predetermined "path". We make here the assumption that there is always only a single way of producing any type of information (unique path), and that a transformed type of information cannot be produced again (no cycling path). This assumption ensures the global fusion process is hierarchical and without feedback (see e.g. [9]). Formally, let T be the set of *information types*, and F the set of *fusion functions*. Each fusion function $f \in F$ is associated with non-empty sets of *input types* $T_f^I \subseteq T$ and *output types* $T_f^O \subseteq T$. These sets must satisfy the above-mentioned assumptions in order to define a *feasible information fusion flow*.

Figure 8 is an example of a feasible information fusion flow composed of three fusion functions, $F = \{f_1, f_2, f_3\}$, and 5 information types, $T = \{t_1, t_2, t_3, t_4, t_5\}$. In this representation that we call the *fusion graph*, an arrow from type t_i to function f_j means that t_i is an input type of function f_j, i.e. $t_i \in T^I_{f_j}$. Similarly, an arrow from function f_j to type t_i means that t_i is an output type of function f_j, i.e. $t_i \in T^O_{f_j}$.

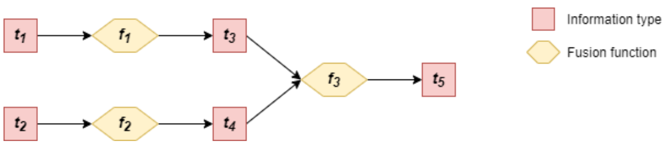

Figure 8: Fusion graph example.

Each sensor node $s \in N_S$ in the network must be associated with the unique information type $t_s \in T$ it is producing. This type cannot be produced by any fusion function in F. Similarly, each HMI node $h \in N_H$ must be associated with an information type $t_h \in T$ which represents an *information request*.

The problem asks, given a network and a feasible information fusion flow, to find a *fusion network topology*. A solution to this problem must define a communication flow within the network, i.e., decide for each link $(i, j) \in L$ if the information type t is sent from node i to node j. Here, we assume that we cannot partially send data from a node to another in the network. Simultaneously, this solution must choose the allocation of the processing algorithms in the network, i.e. decide for each fusion function $f \in F$ if it is executed on the processing node $n \in N_P$. The choices must follow the fusion information flow, while satisfying the sensor data production and HMI information requests.

For example, using the network graph from Figure 7 and the fusion graph from Figure 8, as well as assuming that sensor node n_1 produces type t_1, sensor node n_2 produces type t_2, and HMI node n_6 asks for type t_5, a possible solution could be the one presented in Figure 9. In this representation that we call the *solution graph*, an arrow from node $n_i \in N$ to node $n_j \in N$ with label t_k means that this topology sends information type $t_k \in T$ from n_i to n_j. Furthermore, a label f_w within a processing node $n \in N_P$ means that the fusion function $f_w \in F$ is executed on this node.

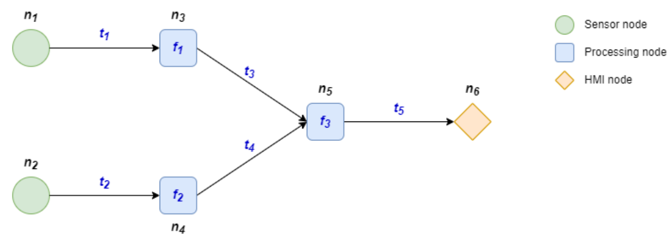

Figure 9: Solution graph example.

3.2. Evaluation Metrics

In general, there will be more than one feasible network topology for an instance of the above-described problem. For example, a fully centralized solution would allocate all fusion functions to a single processing node. Nevertheless, we posit that some topologies are better than others. In order to evaluate the quality of a solution, the intelligent process manager considers the following three metrics:

1. *Network delay*, evaluating the total time it takes for the network to communicate and process the data from the sensors to the HMI nodes;

2. *Links usage*, evaluating the network communication resource usage in terms of data amount being sent, herein as the average communication time on a link over all network links;

3. *Nodes usage*, evaluating the network processing nodes usage in terms of required processing power, herein as the average processing time on a node over all network processing nodes.

Note that using *time* instead of *data* as the basic units for the last two metrics allows them to be weighted by the capacities (throughput, performance), where freeing a high capacity resource has less impact than doing so for a low capacity resource. Also, we use the average in order to ease comparison with different network setups.

According to our requirements analysis, these three elements can be theoretically evaluated by making the following assumptions:

- The data flow through the network is of a single "time step", i.e. of one iteration of information flow from the sensor nodes to the HMI nodes;
- The throughput is stable and not affected by the amount of data transferred;
- No data leak happens during transmission or processing;
- Each node handles parallel computation for fusion functions when possible, without any performance loss.

Table 1: Additional parameters for solution evaluation.

p_s	Data amount (b) produced by $s \in N_S$
b_l	Throughput (b/s) of $l \in L$
o_n	Performance (FLOP/s) of $n \in N_P$
m_n	Memory access speed (b/s) of $n \in N_P$
Q_f	Function giving the data amount (b) of each type in T_f^O after using $f \in F$, given the data amount (b) of each type in T_f^I
O_f	Function giving the required performance (FLOP) of $f \in F$, given the data amount (b) of each type in T_f^I
M_f	Function giving the required memory access (b) of $f \in F$, given the data amount (b) of each type in T_f^I

We must also provide the additional parameters listed in Table 1. Given the quantity of data transferred q_l over the link $l \in L$, the communication time is obtained by $\frac{q_l}{b_l}$. Furthermore, given the performance o_f and memory access m_f requirements of a fusion function $f \in F$ executed on a processing node $n \in N_P$, the computation time is obtained by $\frac{o_f}{o_n} + \frac{m_f}{m_n}$, i.e. the sum of the performance and memory access times.

To illustrate this approach, these metrics have been evaluated for different fusion network topologies on a realistic drone tracking use case. The latter involved four Doppler radars, each connected with high throughput to a performance-limited ("small") node, each of them connected with limited throughput to a single high-performance ("big") node associated to an HMI. The fusion flow required resulting tracks on the HMI distilled by a particle filter from the four processed data by the respective radar models. Results are presented in Table 2. The *Centralized* case assigned every fusion function to the big node. The *Distributed* case allocated the four radar models to their associated small nodes, while the *Hybrid* case distributed only two radar models. Alternative topology possibilities were analyzed in a similar way.

Table 2: Evaluated metrics for the drone tracking use case.

Case	Network delay (s)	Links usage (s)	Nodes usage (s)
Centralized	16.695	0.98463	0.82610
Distributed	11.314	0.22080	0.85267
Hybrid	16.695	0.60271	0.83939

Depending on the chosen metric(s) as its optimization objective, the processing manager determines a different optimal topology. In this case, to minimize network delays or links usage, it would have selected a distributed setting. To minimize instead the nodes usage, and thus maximize the remaining processing capacity, it would have selected a centralized topology.

3.3. Means of Implementation

In the literature, a myriad of problems has been formulated for wireless sensor networks optimization whether to solve design, deployment, or planning questions [10, 11]. Those formulations include many optimization objectives, ranging from minimizing network delays to maximizing the network coverage. The latter is often highly related to the problem of sensors selection for fusion, which has been highly studied [12]. Furthermore, our intelligent process manager problem bears resemblance with many network distribution design problems, notably the ones involving transportation and facility location decisions. A key difference, however, is that fusion changes the amount of data exiting each processing node, so that standard flow conservation rules can't be applied.

The intelligent process manager's problem, as defined in Sections 3.1, can naturally be formulated as a *combinatorial, multi-objective and constrained optimization problem*. Indeed, since the decisions to make involve a finite number of discrete possibilities, with each fusion function either executed on a node or not, and each information type either communicated by a link or not, the solution space clearly involves a combinatorial aspect. Furthermore, considering the network and fusion graph structures, besides having information production and requests, as well as other potential technological requirements and limitations, the problem must contain a considerable number of constraints restricting the solution space. Finally, the multi-objective aspect comes from the consideration of more than one criterion to optimize, potentially simultaneously, as presented in Section 3.2.

In order to model, implement and solve the problem, a state-of-the-art combinatorial optimization paradigm could be used, such as *Constraint Programming* or *Mixed-Integer Programming*. These methods have the advantage of supporting rather straightforward mathematical formulations, as well as an indefinite number of various constraints, which makes them easily extensible. Plus, they are *exact*, i.e. they are guaranteed to find an optimal solution while proving its optimality. To our knowledge, no specific algorithm currently exists for this optimization problem. Note that we are aware of ways to develop non-exact approaches providing near-optimal solutions, such as genetic, particle swarm optimization based, evolution based, custom heuristic and metaheuristic algorithms (see [10]).

The development and testing of the optimization component is currently ongoing work. To validate the feasibility of our approach, we have first implemented a *brute-force* (exhaustive) search method. However, the execution time of the the brute-force method increases exponentially with the number of nodes in the network, thus showing the need to develop more robust and scalable optimization methods.

4. SOTAS COMMUNICATIONS SYSTEM

The heterogeneous fusion stack and intelligent process manager are supported in the mobile tactical domain by the Application Hosting functionality of the versatile SOTAS multimedia communications system [4]. SOTAS is a family of IP-based robust mobile communication, data transfer and computing platforms specifically designed for platforms operating in constrained conditions, such as alongside railroads, on oil platforms, and in military vehicles. The computing platforms (server units) support hosting of different types of applications (own or 3rd party). SOTAS provides high-quality voice and data services, such as intercom, telephony, radio, IP/Ethernet networking services (routing, switching), and server computing.

When looking at the architecture of a military vehicle, one can distinguish the local domain, where the information of sensors associated to the vehicle is collected. Typical sensors are infrared cameras (mounted on the vehicle or on a UAV that is connected to the vehicle), laser-range finders, acoustic sensors, and radar systems (Figure 10).

Figure 10: Sensors mounted on a vehicle.

There is also the C4I-domain where the local information is combined with information from other platforms in the field. To that order, many types of external long-range data connections with different technologies are supported by SOTAS. Examples are narrowband VHF-radios, characterized by omnidirectional behavior, long distances, and a low data rate; UHF radio's, characterized by higher bandwidths and shorter distances (compared to VHF), directional radios; and satellite communication systems.

SOTAS can operate in both the local as well as the C4I-domain, and can also be employed to achieve a separation between those domains.

The SOTAS system consists of the following components (Figure 11). The Tactical Network Node (which consists of a number of layers determined by functional demands) is a central device that provides communication and data services. The audio services are of high quality and include audio processing (e.g., addition of audio streams, and encoding of different streams). Provided data services are Ethernet switching and IP routing. The Server Units of the SOTAS product family support application hosting, with an application server that can host different 3rd-party applications (such as the fusion engine and the intelligent process manager introduced herein, or battlefield management system applications). In addition, the SOTAS family contains a time server to provide accurate timing (for example, to support radios) in case the timing from external sensors (GPS) is lost in a GNSS denied environment. To further improve voice quality and reduce the cognitive load of the vehicle crew, Dynamic Noise Reduction is supported, assuring crystal clear speech over the intercom or radio, optimizing battlefield endurance. The SOTAS system is modular, and composed of different layers that provide different functionality, targeted to the specific role of the vehicle.

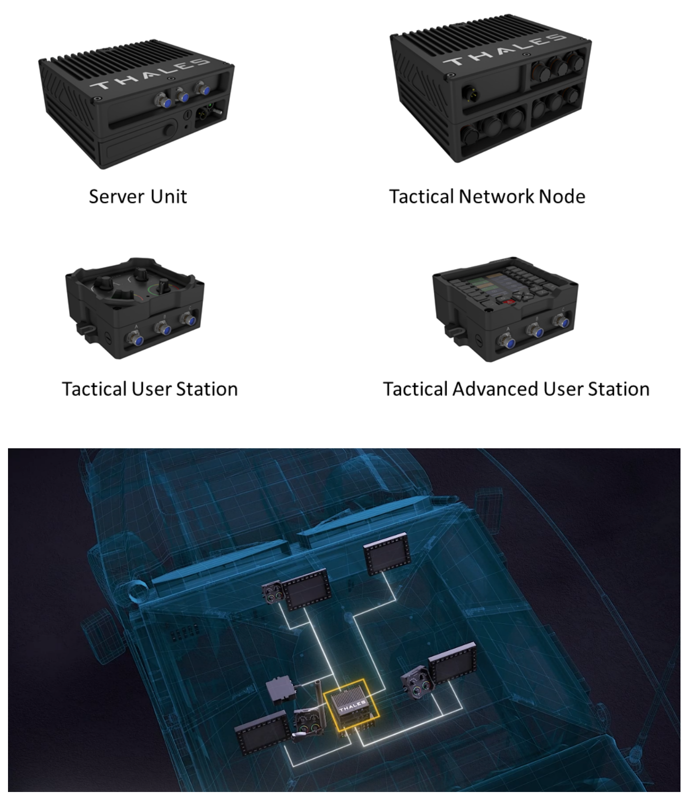

Figure 11: (top) Four elements of the SOTAS product family; (bottom) Example application in a military vehicle.

The Soldier Machine Interfaces (SMI) come in different variants. The SMI (Panel) provides a large display, with a standard way to select different applications inside that display. The Tactical Advanced User Station is a unit that provides access to communication services, audio services, and advanced capabilities like management and maintenance (crew box type of unit). Finally, the Tactical User Station is a unit that provides access to the communication services, audio services, and basic channel selection capabilities.

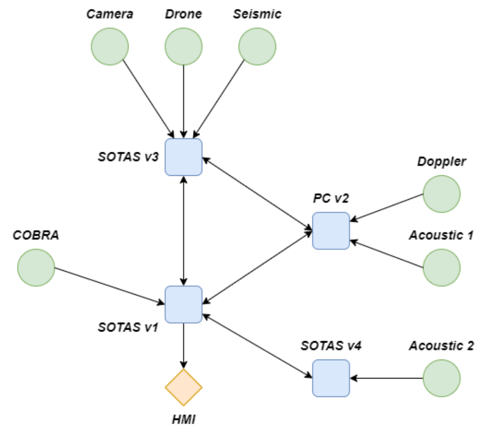

Figure 12: Network graph of the red forces tracking example.

5. INTEGRATED APPROACH

This section illustrates the integrated approach using the red forces tracking example from Section 1.1. Following the scenario in Figure 1, four military vehicles, `v1`, `v2`, `v3`, and `v4` moved to the indicated positions, each providing specific surveillance assets. Vehicles `v1`, `v3` and `v4` are equipped with a powerful SOTAS Server Unit, while `v2` uses a low-performance PC enhanced with an accelerator for processing of neural networks. `v1` is directly connected to the COBRA sensor, as well as the SMI SOTAS component for this mission (*HMI*). `v2` carries an acoustic sensor (*Acoustic 1*) and is directly connected to the dismounted Doppler radar. `v3` is connected via a local military Wi-Fi to the stationary camera and the seismic sensor. Moreover, `v3` has an interface to the imagery of a drone system controlled from that vehicle. Vehicle `v4` carries another acoustic sensor (*Acoustic 2*). The processing nodes from `v1`, `v2` and `v3` can communicate via a radio tactical link with medium bandwidth, while `v4` can communicate with `v1` via a satellite link with limited bandwidth. This current network structure is captured in the network graph of Figure 12. Note that in this example the target's motion and the outputs of the surveillance assets are simulated.

Based on the initial intelligence about the presence of red forces, the operator instructs the system to start the red forces tracking function in a certain surveillance area. This triggers the DPIF mechanism to determine the relevant sensors and fusion functions that should be composed into this tracking function. This information is compiled in the fusion graph of Figure 13.

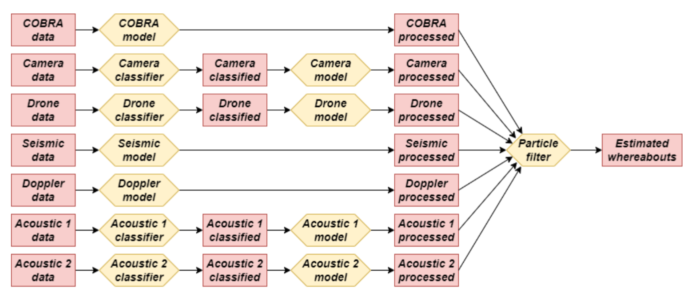

Figure 13: Fusion graph of the red forces tracking example.

The intelligent process manager is supplied with the network graph, the fusion graph, and the information about the available services on the nodes[1]. Satisfying the different physical constraints, and minimizing the overall network delay, the process manager determines the optimal fusion

[1]For the purpose of the example, we assume that each compute node has the local copy of the various functions such that the intelligent process manager can decide where to start the function.

network distribution illustrated on the solution graph of Figure 14. According to this solution, v2 PC runs the local sensor models as well as the neural network for the sound classification using its accelerator. Vehicle v3 handles the Bayesian models for the camera, the drone and the seismic radar, as well as two neural network classifiers. v4, similarly, treats acoustic data from another area with a neural network pre-processing.

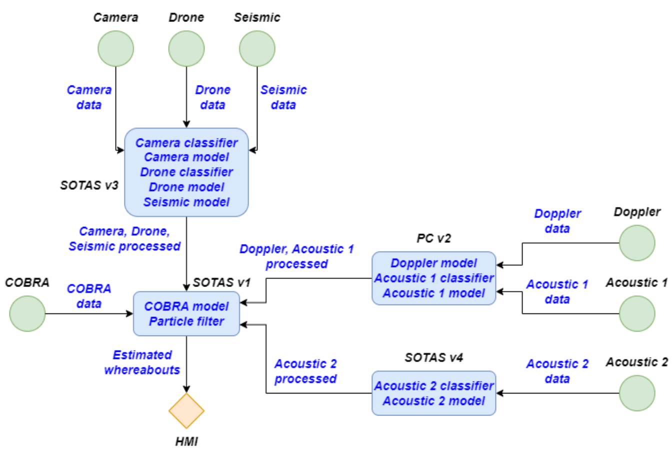

Figure 14: Initial solution graph of the red forces tracking example.

The fusion network topology provided by the process manager is used by the heterogeneous fusion engine as a set of instructions for the DPIF mechanism to activate different types of services on the chosen computing nodes, and establish the information flows among them. In this way, the fusion engine forms a composite fusion system similar to Figure 4, essentially implementing a distributed version of equation (1) that is adapted to this specific constellation of sensors and operational constraints. In this context, the data from each acoustic sensor goes through a neural network classifier recognizing the sound of armored vehicles. There is also a specific neural network for each video stream from the camera and the drone used to classify and localize heavy vehicles. The local Bayesian sensor models translate outputs of various signal processing solutions to a format that can easily be used by the particle filter to generate the resulting estimates of the whereabouts. The estimation results are sent to the vehicle v1 SOTAS node and displayed in the form of heat maps on the HMI as shown in Figure 15. The presented heat maps are output of a real distributed fusion process with simulated target movements (red circle) and sequences of simulated sensor/detector outputs. The color of the heat map corresponds to the probability that a target is at a certain location; the warmer the color, the higher the probability. Heat maps "encode" the entire information absorbed through the fusion of complex time series consisting of heterogeneous data types. The computation of the heat maps is carried out by the Context-Boosted Particle Filter that supports inference over space and time, given the sequences of observations as well as the lack thereof (see Equation (1)).

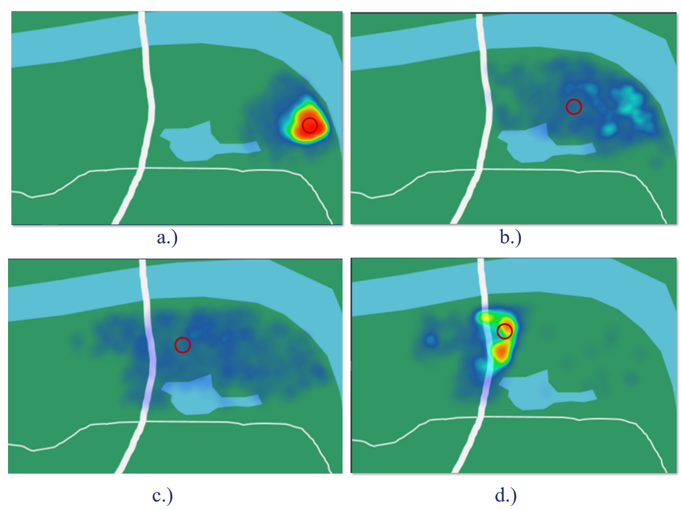

Figure 15: A sequence of estimated whereabouts of red forces using simulated target and sequences of sensor measurements.

Figure 15.a shows the moment the target was detected by the acoustic sensor on vehicle v4. After a while, the target left the field of view of this sensor and entered the surveillance gap. From this point on, no contacts could be made with the

target for a prolonged period of time. Figures 15.b and 15.c show the estimated whereabouts in this period of time. Despite prolonged periods in the surveillance gaps, the uncertainty of the target's position remained limited. This was possible because the fusion considered the fact that the Doppler radar, the stationary camera and the seismic sensor have not produced any detections in their fields of view during that period. This meant that the red forces have not attempted to cross the bridge or the area south of vehicle `v4`. Moreover, by being able to use the information about the areas in which the heavy mobile mortar cannot move, such as the river, a lake, and a swamp, the uncertainty of the whereabouts was additionally reduced, as shown by Figures 15.b and 15.c. Finally, the red forces entered the field of view of *Acoustic 1*. While this sensor is very imprecise, Figure 15.d shows that the uncertainty about the possible whereabouts was significantly reduced.

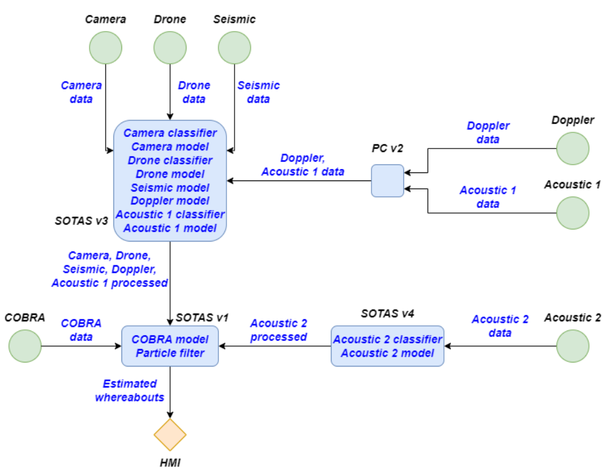

Figure 16: Alternative solution graph of the red forces tracking example following a processing node loss.

In order to show the resilience of this approach, we now assume that node *PC v2* loses its processing capabilities, but can still freely transfer data. A simple solution would be to send all its allocated fusion functions to the vehicle `v1` SOTAS node. However, to minimize the processing requirements on the latter, and due to no connection available with `v4`, the process manager chooses instead to send the raw data to `v3` and let it handle the required functions. This solution, illustrated in Figure 16, allows the fusion engine to preserve the SA and its continuous estimates of the red forces whereabouts.

6. CONCLUSION

Vehicles and vehicle-associated sensors are crucial elements for information-based operations. To get efficient, actionable, timely, and consistent situational awareness in this constrained domain, an approach is required to fuse the information from heterogeneous sensors. Furthermore, to fully exploit the trade-offs between information processing and bandwidth in the field, one needs to distribute fusion processes and dynamically allocate the fusion functions to processing nodes, which can have specialized computational capacities or communication potential. In this way rich actionable information can be distilled from disparate sensors, while optimally exploiting the computing and networking resources in the mobile tactical domain.

This paper introduces a novel modular architecture addressing these challenges. It combines a distributed fusion engine based on hybrid AI correlating the heterogeneous data, an intelligent process manager optimizing resources allocation according to communication and processing capacities, and a system of ruggedized computing and communication hardware. The integrated approach has been illustrated with a red forces tracking example involving multiple vehicles and heterogeneous sensors. Ruggedized mobile computers on vehicles interface sensors and host a system of fusion algorithms that are dynamically combined into an efficient surveillance solution, optimally adapted to mission constraints. The used example illustrates principles and properties that apply to many different types of solutions, not just red forces tracking. It enables a robust and efficient implementation of complex models and inference processes spanning an ad-hoc system of networked computing nodes in the field. The ruggedized computing nodes provide a robust and stable processing platform, the modular fusion engine supports distribution of algorithms over the computing nodes and their composition into advanced fusion solutions, while the intelligent process manager relies on the formalization of a new optimization problem to determine on which computing nodes different fusion components should be used.

As further work, we notably plan to implement the optimization algorithm discussed in Section 3.3, and compare it with our current brute-force method. Finally, while the Heterogeneous Fusion Engine and the SOTAS components have been tested in various physical settings, real world experiments will be repeated with the integrated solution.

7. ACKNOWLEDGEMENTS

We would like to acknowledge the Department of National Defence (DND) of Canada for partly funding this research work through the Innovation for Defence Excellence and Security (IDEaS) program (contract CP5-5125).

8. REFERENCES

[1] T. Karako and W. Rumbaugh, "Distributed defense: New operational concepts for integrated air and missile defense," *Center for Strategic and International Studies (CSIS)*, Jan. 25, 2018.

[2] R. Dean and N. Teeple, "First Report: Awareness & Sensors," *CDA Institute, NORAD Modernization Forum*, 2020.

[3] N. Glesby, "Information dominance," *CDA Institute, NORAD Modernization Forum*, 2021.

[4] "Sotas C4i infrastructure," Thales Group. (2023), [Online]. Available: `https://www.thalesgroup.com/en/sotas-c4i-infrastructure` (visited on 03/06/2023).

[5] P. de Oude, G. Pavlin, and J. P. de Villiers, "High-level tracking using bayesian context fusion," in *2018 21st International Conference on Information Fusion (FUSION)*, Jul. 2018, pp. 1415–1422.

[6] R. Claessens, G. Pavlin, and P. de Oude, "Context driven tracking using particle filters," in *2015 18th International Conference on Information Fusion (Fusion)*, Jul. 2015, pp. 1128–1135.

[7] G. Pavlin, M. Kamermans, and M. Scafes, "Dynamic process integration framework: Toward efficient information processing in complex distributed systems," *Informatica*, vol. 34, no. 4, 4 2010.

[8] A. Penders, G. Pavlin, and M. Kamermans, "A collaborative approach to construction of large scale distributed reasoning systems," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 06, pp. 1083–1106, Dec. 2011.

[9] M. Liggins, C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 95–107, Jan. 1997.

[10] M. Iqbal, M. Naeem, A. Anpalagan, A. Ahmed, and M. Azam, "Wireless sensor network optimization: Multi-objective paradigm," *Sensors*, vol. 15, no. 7, pp. 17 572–17 620, 7 Jul. 2015.

[11] D. Kirov, P. Nuzzo, R. Passerone, and A. Sangiovanni-Vincentelli, "Optimized selection of wireless network topologies and components via efficient pruning of feasible

paths," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, Jun. 2018, pp. 1–6.

[12] X. Deng, Y. Jiang, L. T. Yang, M. Lin, L. Yi, and M. Wang, "Data fusion based coverage optimization in heterogeneous sensor networks: A survey," *Information Fusion*, vol. 52, pp. 90–105, Dec. 1, 2019.