2011-01-31

Persistent Intelligence Surveillance and Reconnaissance Product Line Architecture ver 1.1

Rapid Prototyping Valued Information at the Right Time (RapidPro-VIRT) Project Team

Persistent Intelligence, Surveillance, and Reconnaissance (PISR) Product Line Architecture (PLA)

Version 1.1

31 January 2011

Prepared by:
Rapid Prototyping Valued Information at the Right Time (RapidPro-VIRT) Project Team

Prepared for:
Marine Corps Systems Command
Program Manager for Intelligence (PM-Intel)

This page intentionally left blank.

Persistent ISR Product Line Architecture (PISR PLA)

The Marine Corps Intelligence, Surveillance, and Reconnaissance Enterprise (MCISR-E) Roadmap[1], issued in April 2010, describes USMC's systematic and multi-faceted approach to seize the "high ground" afforded by intelligence superiority over the enemy. Persistent ISR (PISR) is an emerging tactic for agile collection and exploitation of battlefield intelligence. Product Line Architecture (PLA) is an accepted business tactic for agile development and deployment of the tools that will enable intelligence collection and exploitation. Both PISR and PLA represent profound and deliberate departures from a status quo deemed inadequate by USMC leadership to fight the modern threat.

The mission of the MCISR-E is to integrate all USMC ISR elements into a single networked capability across all echelons and functional areas to achieve superior decision making and enhance lethality.[2] PISR is the MCISR-E strategy to synchronize organic Marine Air-Ground Task Force (MAGTF) ISR assets, and thereby deliver continuous relevant battlespace awareness across all echelons of leadership.[3] A key tenet of MCISR-E is "rapid technology insertion through rapid prototyping and acquisition."[4] Accordingly, in December 2009, the Marine Corps Systems Command (MARCORSYSCOM) Program Manager for Intelligence (PM-Intel) established a rapid prototyping team (RapidPro) in direct support of MCISR-E. The RapidPro mission is to rapidly deliver leading edge, interoperable technologies that assist Marines to find, fix, or kill the enemy in all operating environments.[5]

A PLA defines a structure for an extensible family of reconfigurable systems. Successful PLAs dramatically reduce development cost, complexity, and time to market. They also lower barriers that traditionally impede government's ability to deploy advanced applications, innovative processes, and new generations of computing and communications infrastructure. PISR PLA is based on best commercial practice for open system design, and is optimized for rapid discovery or development, and subsequent fielding, of increments of capability. PISR PLA includes objective measures of value validated by warfighters. PISR systems derived from this PLA will comprise modular components that will mostly come off the shelves of government or commercial product developers. New and better off-the-shelf capabilities become available continuously as technologies and products advance over time. The PISR PLA aims to anticipate such relevant product advances to reduce the time and cost required to incorporate them into specific fielded systems that demonstrably add value over and above status quo.

This document describes the initial PISR PLA launching new processes for rapidly fielding advances in PISR capabilities that directly address USMC warfighter needs. It progresses from a high-level system view, through subsystems, to successively finer components. Because all components in a PLA are generic, system implementers are free to choose their own specific component implementations to best meet the operational needs. Each specific PISR PLA implementation will have a version-specific design document that describes the particular fielded sensors, processes, reasoners, algorithms, and techniques. Each successively released system will contribute to the ideal defined by the PISR PLA, while providing measurable value to the tactical warfighter.

---

[1] Marine Corps ISR Enterprise Roadmap, Headquarters Marine Corps, Intelligence Division, Arlington VA, April 28, 2010.
[2] *Ibid.*, p. 4.
[3] *Ibid.,* p. 18.
[4] *Ibid.,* p. 4, "(g) *Rapid technology insertion through rapid prototyping and acquisition*."
[5] From PM-Intel Rapid Prototyping Team mission brief, slide #9, 2010.

This page intentionally left blank.

Naval Postgraduate School, Monterey, CA.
Version 1.0
December, 2010

This product was developed through a collaborative effort of the Naval Postgraduate School, Office of Naval Research, Los Alamos National Labs, Teledyne Solutions Incorporated, George Mason University, Cougaar Software Incorporated, Charles River Analytics, and Raytheon Missile Systems under sponsorship from the United States Marine Corps.

This Specification is intended to be a living, evolving document. It can contain technical inaccuracies or typographical errors. Changes are made periodically to improve the information herein as the concepts and approaches mature over the life of the project. These changes will be incorporated in new versions of the Specification. The Naval Postgraduate School may make improvements and/or changes in any technology, product, or program described in this Specification at any time in accordance with Marine Corps guidance and the goals of the project.

Contact Information

To request copies, suggest changes, or submit corrections, send a message to the email address below:
hayes-roth@nps.edu

For correspondence, please submit comments, feedback, and requests to:
Rapid ProVIRT Project, IS Department, NPS, Glasgow East 3001
1411 Cunningham Rd., Monterey, CA 93943

File Information
Last Saved: 1/31/2010

Revision History

Date	Version	Description
12/31/2010	1.0	Initial Version
1/31/2011	1.1	Additional sections on PISR Networking and Rapid Prototyping. Editing and rework throughout.

This page intentionally left blank.

Table of Contents

Table of Figures

1 PISR Product Line Architecture

1.1 Introduction

> "Accurate, timely, and relevant intelligence is critical to the planning and conduct of successful operations. Effective intelligence uncovers enemy weaknesses which can be exploited to provide a decisive advantage. Shortfalls in intelligence can lead to confusion, indecision, and unnecessary loss of life, mission failure, or even defeat." [6]

The Marine Corps Intelligence, Surveillance, and Reconnaissance Enterprise (MCISR-E) Roadmap[7], issued in April 2010, describes USMC's systematic and multi-faceted approach to seize the "high ground" afforded by intelligence superiority over the enemy. Persistent ISR (PISR) is an emerging tactic for agile collection and exploitation of battlefield intelligence. Product Line Architecture (PLA) is a tactic for agile development and deployment of the tools that enable intelligence collection and exploitation. Both PISR and PLA represent profound and deliberate departures from a status quo deemed inadequate by USMC leadership to fight the modern threat.

Persistent Intelligence, Surveillance and Reconnaissance (PISR) is the ability to create continuous battlespace awareness through optimized sensor tasking, analyst-directed and operations-focused information processing, and rapid distribution of valued information. PISR collects, processes, and delivers timely, actionable intelligence specifically tailored to enhance operations success by incrementally and probabilistically translating data into pre-defined information value. PISR enables leaders at all tactical echelons to synchronize organic Marine Air Ground Task Force (MAGTF) and Joint apportioned collection assets in the battlespace by providing them with timely, relevant, and continuous awareness within their respective named areas of interest (NAIs). Maintaining continuing relevance requires that PISR continually evolve to effectively employ new sensors and techniques to address new environments, adversaries, and tactics.

The current intelligence systems acquisition systems process is a long, serial, monolithic, and Balkanized relic of the Industrial Age. In contrast, the MCISR-E Roadmap specifies a rapid, parallel, open modular approach to acquisition that is appropriate for the Information Age. PLA represents just such an approach, indeed an approach that is a recognized "best practice" employed by the most successful commercial Information Technology practitioners. Generically, a PLA is a detailed technical specification of a modular, "open", easily customizable approach to assembling technology to provide capabilities associated with a particular problem set. PLA must objectively and thoroughly describe the constraints, functionality, required performance characteristics, and interfaces in sufficient detail to allow PLA-compliant commercial and government off-the-shelf (COTS/GOTS) component providers, and/or developers, to deliver out-of-the-box, plug-and-play functionality. The variety of iPods, iPhones, iPads illustrate the point: the manufacturer can deliver many specific products that deliver content tailored to users' tastes, needs, and operating context. This document describes a PLA optimized per the specific requirements of the MCISR-E roadmap.

Different operating environments, operational objectives, and hostile threats warrant operationally-configured collections of sensors and software components. Unique characteristics of different missions and operational contexts drive the need for multiple diverse system configurations. A PLA assists by defining a family of potential products composed of reusable and interoperable hardware/software frameworks and components. While the PLA describes such frameworks and components generically, such as a map display or an electro-optical (EO) sensor, the system manufacturer assembles a catalog of specific parts that can fulfill the roles of the generic frameworks and components. Thus, the system developer can quickly configure an operationally relevant system by choosing off-the-shelf implementations and combining them as the PLA dictates. The result is a specific product, tailored to a particular user community and operational environment, delivering assured functionality with predictable quality and cost.

PLAs are inherently extensible. PLAs anticipate and facilitate evolution by incorporating a succession of improved component and framework implementations. Successful PLAs dramatically reduce development cost and complexity, time to market, and barriers to the introduction of new technologies, processes, and techniques. Hence, the PISR PLA will enable the USMC to migrate to the latest technologies, rapidly adapt to new operational contexts, and

---

[6] Marine Corps Doctrinal Publication (MCDP)-2, Intelligence, pg. 28.
[7] Marine Corps ISR Enterprise Roadmap, Headquarters Marine Corps, Intelligence Division, Arlington VA, April 28, 2010.

iteratively improve actionable intelligence. PISR Systems conforming to the PISR PLA may vary slightly or significantly. For example, systems with similar functionality may have slight differences across echelons, or may have considerable differences between small forward operating bases (FOBs) in rural environments versus large deployments in major urban peacekeeping operations.

As sensor, collection, storage, computing, and communication technologies advance at exponential rates, the human bandwidth of our front line warfighters remains limited. Further, in many tactical contexts, information value is extremely perishable. Therefore, beyond accelerating the integration and delivery of off-the-shelf frameworks and components, the PISR PLA must provide engineering assurance for technical functionality that will bring the right data to the right person(s) or machine(s) at the right time. PISR PLA explicitly enables operators to delegate the monitoring and selective filtering of data to the machines that constitute the PISR system of interest. Users specify their information requirements and "train" the PISR system to sift through data looking for events matching those requirements.

Detailed functional requirements and quality attributes for the PISR PLA come from extensive interaction with many representative members of the envisioned user community. The requirement collection process includes identifying operational scenarios, vignettes, and included mission threads[8] (see section 0). PISR System architects derive information value chains by working with operators to analyze critical mission threads. The most highly valued information is that which, if delivered in time, would cause the recipient to alter a pre-planned Course of Action (COA), and thereby achieve better mission outcomes. Operators place less value on information that simply provides situational awareness of expected conditions. PISR PLA is optimized to: (1) support intelligence analysis, filtering, and case file development that incrementally and probabilistically translates data into valued information; (2) deliver valued information in time to make a difference.

The strategy for deploying PISR is to use the PLA as a defined technical structure for integrating various tactical ISR concept of operations (CONOPS) and technologies the Marine Corps is developing, evaluating, or has developed. Such technologies include the Ground-Based Operational Surveillance System (GBOSS) variants, Tactical Remote Sensor Systems (TRSS), Intelligence Analysis System (IAS) Family of Systems (FoS), MAGTF Secondary Imagery Dissemination System (MSIDS)[9], the Los Alamos National Lab's (LANL) Wide Area Airborne Surveillance (WAAS) and Tactical Switchboard (TS) systems, to name a few. Concepts developed by the Marine Corps Warfighting Lab (MCWL) include "Enhanced Company Operations (ECO)" which is mature, and Company Level Operations Cell (CLOC), which is still maturing. CLOC includes an intelligence collections and analysis component[10] for quickly gathering and processing intelligence at the company level. The Marine Corps has also developed a CONOPS for Unmanned Aircraft Systems (UAS) FoS.[11] This CONOPS describes a future where multi-mission/payload-capable UAS detachments will provide improved battlespace situational awareness (SA) and communications relays to echelons at battalion-level and below. What is missing is a cohesive integration framework within which the diverse technologies and systems can interoperate synergistically for measured improvements to Marine Corps intelligence capabilities. *The PISR PLA provides this framework by being both an architecture of generic components and a programmatic process for rapidly evaluating, certifying, and fielding new capabilities.*

In summary, (1) composing systems for today's environment from plug-and-play products, and (2) automatically detecting and alerting high-value events so operators can make timely adaptive decisions, constitute the two fundamental pillars of PISR PLA. To build these pillars, USMC Program Manager, Intelligence (PM-Intel) and the Naval Postgraduate School (NPS) have formed a PISR PLA development team. Significant contributors to this work include the Office of Naval Research (ONR Code 30, ISR Thrust), Los Alamos National Labs (LANL), Cougaar Software Incorporated (CSI), Teledyne Solutions Incorporated (TSI), Raytheon Missile System (RMS), Charles River Analytics (CRA), and George Mason University (GMU C4I Center).

---

[8] Mission Threads are carefully described, detailed sequences of activities associated with tasks and/or subtasks that occur within operational scenarios of interest.

[9] See the Marine Corps Systems Command (MCSC) Fact Book, available on line: http://www.marcorsyscom.usmc.mil/sites/cins/Fact_Books.html

[10] CLOC has now been merged with the earlier separate Company Level Intel Cell (CLIC) concept.

[11] Concept of Operations (CONOPS) for United States Marine Corps (USMC) Unmanned Aircraft Systems (UAS) Family of Systems, (FoS), Fires and Maneuver Integration Division, Capabilities Development Directorate, Marine Corps Combat Development Command, Quantico, Virginia. 10 November 2009, Version 2.0, pp. 21-23.

Success of the PISR PLA will be measured by the speed-to-value of PISR generic frameworks and components it enables for particular customers per the following goals:

- Any specific PISR System, by specializing and implementing the PLA, should answer several questions clearly: What threats are reduced? What opportunities can be exploited? What software infrastructure, interface/service infrastructure, and physical capabilities are required and provided? What are the system Quality Attributes (QAs)[12] that can be delivered and to what fidelity?
- A specific system should be easily configured to meet the needs of the users by dealing with such questions as: What are the components and their qualities? How do we assure the system answers user questions and meets user expectations?
- The PISR PLA must prescribe how to test, validate, and certify PISR Systems and to accomplish this quickly and efficiently.
- PISR Systems should continually monitor system performance and detect unanticipated situations and problems. They should alert appropriate actors in the system and automate appropriate adaptive responses.
- PISR Systems should be modified, adapted, and iterated as required in response to lessons learned in the operational environment.

This document presents the concepts and architecture for the PISR PLA in terms of a high-level system view and descriptions of the primary subsystems that constitute a PISR System. The specification of the architecture includes the interfaces and information flows across these subsystems in sufficient detail so developers independently can contribute products for integration into the system. The specification also identifies important external interfaces to drive semantic integration of multiple information sources into the system. Adherence to the PLA enables composition of effective high-quality systems from off-the-shelf frameworks and components and as-needed, focused product developments. Each system released based on the PISR PLA will require a system design document conforming to the PISR PLA that specifies the sensors, processes, reasoners, algorithms, techniques, and other frameworks/components being fielded in that version. The PISR PLA serves as a template for developing PISR System design specifications. Each subsystem specified by the PISR PLA provides a reference design for detailing what the subsystem offers to the PISR System in terms of its capabilities, its interfaces, the information it needs, and the information and/or services it provides to other subsystems.

The following notional operational vignette, modified from the USMC UAS FoS CONOPS, illustrates a "to be" view of MAGTF PISR capabilities in an operational context:

> ***Per Marine Expeditionary Force (MEF) task order, the Regimental Combat Team (RCT)-2 is the MEF's focus of effort for an attack in an insurgent-occupied town in the far western, mountainous part of the MEF's area of operations (AO). The insurgents have mounted several brazen attacks on Marine forces performing security operations near the international border farther to the west. The enemy has been intimidating the civilian population to cover their operations. To support operations, the MEF provided four Unmanned Aerial Vehicle (VMU) detachments of Small Tactical Unmanned Aircraft (UA) Systems (STUAS) (Shadow) to RCT-2 headquarters (HQ) element and the regiment's three battalions. The RCT HQ's Shadow detachment consists of (4) Wide Area Airborne Sensor (WAAS)-configured UASs. Two are airborne at any given time allowing for short-duration 24-hour operations. The Battalion (BN) VMU detachments (3 UA per detachment) each provide two dual-configured Shadows consisting of airborne network relay, electro-optical (EO) sensors, and Signals Intelligence (SIGINT) sensors. With these assets, up to six Group-3 UAs can be airborne at any one time. This provides RCT-2 with 24-hour UAS coverage in its area of responsibility (AOR) and immediate support of its mission to clear the insurgents from this town.***
>
> ***Recent Human Intelligence (HUMINT) indicates most enemy activity is concentrated near the central mosque in 1st Battalion's AOR. Consequently, the RCT S-2 establishes the mosque and surrounding area as a named area of interest (NAI). The RCT S-2 develops a layered collection plan for the NAI in support of 1st Battalion (the main effort). This plan leverages airborne collection***

[12] Quality Attributes are prioritized operator-specified needs to be satisfied by the system.

assets, ground-based assets (GBOSS), ground sensor platoon (GSP) TRSS, HUMINT, and National Technical Means (NTM). The RCT Air Officer coordinates the plan, prioritized with RCT commander's objectives, and adapted to the BN commander's inputs. An automated PISR collection planning tool analyzes blue force CONOPS, threat data, terrain data, and meteorology and oceanography (METOC) data to provide the RCT and BN Intelligence analysts with optimum placement and employment of electro-optical (EO), infra-red (IR), and air-ground sensors. Plans for UA loiter areas and payloads provide both persistent surveillance using EO and SIGINT sensors and seamless command and control (C2) coverage via a tactical, self-forming/self-healing mesh network. These capabilities enable individual battalions to receive valued information, enhance situational awareness (SA), synchronize maneuver, disseminate time critical information, and exploit intelligence.

In addition to the visual EO and IR surveillance of the mosque area, 24-hour SIGINT UA payload sorties provide an airborne signals detection environment that allows the RCT's Radio Battalion (RADBN) to monitor for particular voice communications. Registered SIGINT conditions of interest (COIs) include specific voice communications from High Value Individuals (HVIs).[13] The PISR System continuously monitors for event occurrences and alerts commanders, analysts, and operators when they occur. The Air Officer also values these COIs—if significant activity occurs near the mosque, he may need to task additional full motion video (FMV) and dynamic targeting. The S-2 concurs with this assessment.

The PISR collection management software plans and manages the STUAS loiter areas, mission profiles, and target sets. PISR exports flight information into the air tasking order (ATO) planning process and into automated collections planning software. The system simultaneously networks and multicasts payload data to RCT, BN, and company (CO) Combat Operations Centers (COCs) as required. This process fuses PISR data essential for deliberate Intel analysis at higher headquarters (HHQ) via Tactical Switchboard while simultaneously distributing actionable information and low resolution imagery to lower combat echelons of the MAGTF. PISR System servers maintain connectivity with Tactical Switchboard Viewers (TSVs) at the BN and CO levels and leverage the tactical mesh network to disseminate actionable intelligence through the RCT-2's echelon of command.

After 24 hours of SIGINT and WAAS surveillance, PISR analytics processing detects trends and makes initial correlations. Analysts develop and enhance threat signatures, constructing case files describing HVIs and other information important to ongoing and planned operations. The case files unify information from diverse sources so that analysts can readily find, review, and update critical information.

During the second day of flight operations, SIGINT sensors detect traffic associated with known insurgents and provide general geo-locations of these communication devices. Drawing from the WAAS and SIGINT information from the first day, the S-2 determines that the SIGINT data and the vehicle traffic concentrate at a house (a suspected "safe house") near the mosque. The Air Officer re-directs Shadow #2's Lightweight Expeditionary Airborne Persistent Surveillance (LEAPS) sensor "watch boxes" via the TSV to focus on the mosque's front door, back garden, and all avenues of approach. Shadow #1 continues providing network relay and EO sensing.

GBOSS towers are in place such that 1st Battalion has sensor line-of-sight with the mosque complex via GBOSS-Heavy, complete with a wireless point-to-point link (WPPL-D) back to RCT-2 HQ. Per the collection management plan, company-level GBOSS-Lites are deployed to various sectors and avenues of approach to the mosque to provide EO/IR surveillance over multiple observation

[13] Conditions of interest (COIs) are descriptions of situations in the battlespace that the intelligence analyst considers important enough to warrant an alert if the PISR System sensors and analytics perceive that situation to have occurred. One can think of these as user-defined *situational triggers*—when the PISR System sensors and analytics perceive that situation in the battlespace, the system alerts the user accordingly.

sectors. WPPL-D links on these nodes allow networking with the BN GBOSS Ground Control Station (GCS) and Intelligence Operations Center (IOC), thus enabling dynamic tasking in response to evolving threats. Mast-configured mesh network nodes (WaveRelay and TrellisWare) provide network communications and data relay for the CO, with reach-back to HHQ via the overhead Shadows.

TRSS sensors in the objective area cover dead spots missed by the other assets and provide triggers for air-ground sensor slewing and combat operator notification. These sensors are linked via the GBOSS towers and airborne mesh network relays provided by the Shadows and Ravens.

Meanwhile, SIGINT hits and associated WAAS imagery, correlated with TRSS triggers, alert the RCT and BN S-2s to a potential HVI getting into a white truck at the safe house. The WAAS "watch box" slews to the vehicle per analysts' speculation that the vehicle is heading towards the mosque. The RCT Watch Officer uses his light pen to annotate the HVI event in progress on the Tactical Switchboard screen. The BN and VMU detachment Unmanned Aircraft Commanders (UACs) also receive immediate alerts via their respective TSVs. Instant chat messages inform them that the vehicle should soon arrive at the mosque. In the Regimental COC, the Watch Officer circles the HVI location on his TSV with his light pen. Instantly, updated red position location information (PLI) propagates via PISR System services to appropriate ground BN and CO command elements. Traditional Very High Frequency (VHF) Battalion Tactical Command net confirms receipt. Adapting to the unfolding situation, the Air Officer in the Fire Support Coordination Center (FSCC) adjusts the UA coverage pattern to provide optimized overlapping C2 coverage as well as EO sensor view of the objective.

PISR components continuously monitor the front parking area and door of the mosque as the white vehicle arrives. A tall, lanky, bearded individual jumps out of the white truck and runs into the mosque. PISR simultaneously pipes the narrow field of view full motion video from the EO Shadow, and wide field of view WAAS Shadow into the RCT and BN COCs. Both the Regimental S-2 and S-3 watch these events in real-time and discuss their options. Meanwhile, the white truck quickly speeds off with several people still inside. The Regimental Watch Officer uses his light pen to circle the vehicle on his screen, thereby directing the UA to adjust the WAAS watch box to track the associated vehicle. His action also triggers TRSS and GBOSS sensors to align and collect sensor data while simultaneously feeding updates to Tactical Switchboard. While continuing to track the white vehicle with WAAS, the SIGINT payload collects information from the area in and around the town. As the truck slows to a stop, all occupants hastily exit the vehicle and start digging up weapons and explosives from a buried cache. The Air Officer instantly receives coordinates of the insurgents via the mesh network. He coordinates on-call rotary wing close air support (RWCAS) via a section of AH-1Z Cobras holding at control point (CP) Viper.

PISR analytics correlate the WAAS feed, SIGINT, and GBOSS/TRSS information. The RCT S-2 determines they have found "their guy" at the mosque. Current video confirms he is in the back courtyard. Commanding Officer, 1st BN receives an execute order to apprehend the HVI. Alpha Company quickly moves to seal off avenues of approach while overhead UAs continue observation to determine if any hostiles in the area may oppose the action. As point elements of Bravo Company approach the mosque compound, the HVI detects Blue forces in the vicinity and begins to evade through the local village. PISR components dynamically push coordinates to the lead infantry elements—all the way down to the squad level—and the Marines on the ground quickly apprehend the suspect without further incident.

1.2 PISR Product Line Architecture Ecosystem

One of the objectives of the PISR PLA is to create the framework for an ecosystem around the value proposition associated with developing and deploying PISR. That ecosystem has several key elements:

- **Research and Development (R&D) Community:** By engineering well-defined pluggable modules for sensors, feature extraction, behavior classification, sensor allocation, and other key components of the PISR

System, the PLA provides a clear transition target and environment for the research and development of new and improved hardware and software components. The expectation is that a diverse and competitive R&D community will emerge focused on incremental improvement and enhancement of successive PLA product versions.

- **Commercial Community:** Long acquisition time lines characterized by bureaucratic overhead are an anathema to both commercial vendors and front line Marines. Enhanced time-to-value is a strong value proposition for both. Well-defined pluggable modules also allow for a variety of commercial vendors to take products that already exist and write interfaces to conform to the PLA. This allows rapid development of a PISR System without the overhead generally associated with large government system production. It enables the USMC to tap efficiently into a huge world of applicable technology and products.
- **Operational Community:** The PISR PLA is driven by operator needs captured through extensive interaction with operators. Architects work with operators to first define information value hierarchies, and then design information value delivery chains. Information value is derived from its ability to take advantage of an emergent opportunity or to avoid an emergent threat. Information value is captured heuristically as Quality Attributes (QAs); i.e., descriptions of how an information system might specifically assist operators in excelling at known tasks. The development process maintains traceability of QAs to each PISR product version and establishes a structured mechanism for system evolution. These changing needs often reflect the evolution of environments, adversaries, and tactics. Engineering support for this evolution ensures PISR Systems can be responsive and timely in providing the right capabilities to the warfighter beyond those envisioned at the time of its original design and development. Ultimately, the QAs relate to the value of information delivered by each PISR product version in context of the particular operational environment. Components that measurably improve delivery of operational value to the user are included in the version; components that do not are excluded. System capabilities evolve in the PLA ecosystem based on the ability of configured components to deliver value to the users.
- **Budgeting Community:** Maintaining well-defined information value metrics such as those described in the Joint Interoperability Test Command Value-Based Acquisition Framework (VAF)[14] and QAs that are prioritized by the user community and traceable to product versions will establish a means of directly relating investment to resulting capability. Further, by categorizing systems in terms of the measured information value they deliver to operators, we can quantify how much any potential component contributes. This quantification provides an estimated return on any potential investment in such a new or improved component capability. Production and delivery of high-value information to the warfighter then becomes the guiding managerial principle for adaptive evolution of MCISR-E capabilities.

1.3 Enterprise Architecture and the PISR PLA

The Marine Corps has studied how best to apply enterprise architecture, within constraints of the Defense acquisition regulations, to its ISR needs.[15] Enterprise architecture (EA) is a methodology for incrementally improving the Information Technology (IT) portfolio to improve business processes. Commercial best practices in EA focus on relatively short cycles of planning and implementation to achieve benefits incrementally and iteratively. Industry has learned that long cycle times produce worse results. In addition, industry has learned the importance of adopting commercially successful architectures, where vibrant markets provide continually improving components that can interoperate and produce combinatorial value. Industrial EA ordinarily adopts and commits to various PLAs[16].

In government applications, EA is less obviously successful. As with most oversight activities, government practice of EA tends to focus on compliance rather than desired outcomes. PISR PLA focuses on desired outcomes per the MCISR-E Roadmap.

---

[14] VAF is a family of metrics that tightly couples desired lag metrics in terms of "Delivered Information Value" (DIV) to information system performance metrics in terms of "Information Processing Efficiency" (IPE). VAF also tightly couples Both IPE and DIV to acquisition process metrics that aim to optimize "Time to Value" (Tv) by emphasizing bundling re-usable components.

[15] See for example "Initial capabilities document update for the Marine Corps Intelligence Surveillance Reconnaissance Enterprise (MCISR-E)," Version 1.4, 30 Sept 2010.

[16] The Carnegie Mellon Software Engineering Institute has conducted many case studies documenting the value of product line engineering in context with defense acquisition processes.

Emerging commercial offerings that show promise for PISR application address mapping, 3D modeling from images and video, image and video analysis, surveillance, biometrics, vehicle and person tracking, monitoring, and alarming. Unmanned vehicles for surveillance and high-risk missions represent additional areas of investment and growth. Clearly, the Marine Corps wants to tap into these areas and exploit progress where possible. The PISR PLA will directly support that objective by applying EA to create a low-barrier gateway for such commercial offerings to: (1) validate their worth, and (2) streamline their deployment to the battlefield.

Abstractly, the goal of EA is to optimize the return on investment in equipment, software, training, and support to implement the measurably best enterprise business processes. "Optimum" means purchasing, deploying, and employing equipment and people in ways that produce good outcomes, quickly, and at low cost. In Marine Intelligence, there are two categories of good outcomes: "specific" value and "general" value.

Specific value corresponds to measures such as saving lives, reducing waste, or shortening campaigns by detecting and responding quickly to specific threats and opportunities. Examples of ways to achieve specific value include: detecting an improvised explosive device (IED) emplacement and avoiding death and injury that would result from its detonation; detecting an incipient ambush and reducing likely harm by avoiding or disrupting it; detecting a high-value individual and seizing him so that future resources won't be expended searching for him or countering his activities.

General value, on the other hand, is obtained by continuously developing background information about the battlespace, including the people, facilities, communications, affiliations, culture, calendar, and so forth. This general background intelligence analysis aims to produce a richer and more accurate model of the battlespace. This work is analogous to making continuous deposits to a long-term investment without guaranteed returns. Benefits arise in various ways, for example: fortuitous discovery of anomalous patterns of behavior serendipitously leads to apprehension of an HVI; a planner's choice of a COA for a particular operation is well-informed by readily available, relevant, processed intelligence; a detected event requires additional information and, fortuitously, applicable pre-processed intelligence is on the shelf. In these cases, general value arises rather unpredictably, and the fraction of background work that proves valuable is, theoretically, a much smaller fraction than in the case of information processing targeted at specific value.

So the overall EA optimization problem has three dimensions: (1) scoping development cycles; (2) balancing an investment portfolio across specific and general value categories; and (3) allocating available PISR resources accordingly. PISR PLA addresses these questions, as we describe briefly here.

PISR PLA envisions implementation cycles of 18 months or less. This time line aligns well with 6-12 month commercial cycles and the notional "Moore's Law" 18-month refresh rate of computer chip architecture. The development cycle must include testing and certification for information assurance and enterprise interoperability. Success requires working with the appropriate authorities to develop new modular approaches to testing and certification that align with the PLA design philosophy.

The PISR PLA provides direct support of specific information value by applying user-defined conditions of interest as a filter to relate detected events to valued outcomes, such as interdicting HVIs and avoiding IEDs and ambushes. The PISR PLA also provides a foundation for improved general information value by providing a case management framework based on a common semantically integrated information base. Case management supports developing ever-richer models of all aspects of the battlespace associated with any particular entity of interest. As for the appropriate balance of effort between specific and general value-related processes, human subjective judgment must apply. The industrial rule of thumb suggests allocating 5 to 20% effort to general value, and the remaining 80-95% to specific value processes.

PISR optimizes resource allocation in terms of both acquisition investment value and operational deployment value. Any condition of interest identified by a user has an expected military utility, such as expected lives saved. Each condition of interest for any particular event can be supported by various "admissible configurations" of system components. The solvable optimization problem then becomes achieving the highest summed utility of supported processes across all time and space where each potential event has an expected frequency and an expected military utility. PISR architects have already solved this problem in a realistic demonstration case[17] (see section 1.6). With maturing and greater adoption of this methodology, PISR PLA will enable program managers to optimize their

[17] T. Levitt, *et al.,* "Valuing PISR Resources Functional Design Prototype Build & Experiments," Version 1.0, 30 September 2010. George Mason University, C4I Center.

acquisition investment portfolios against their key performance parameters, while enabling commanders in the field to employ their ISR resources in ways that assure optimized returned information value against their most critical tasks.

1.4 Top-Level Architecture Concept

The PISR PLA defines a net-centric architecture composed of multiple sensing, analytical, and communications components deployed at networked nodes providing standalone and distributed capabilities. Each node provides a set of functionality that contributes to enterprise goals for gathering and processing data and disseminating timely, high-value information to the right users in appropriate context. As illustrated in Figure 1 below, the PISR PLA comprises four high-level Subsystems: the User Interface (UI) Environment Subsystem, the Situational Awareness (SA) Subsystem, the Management and Control Layer (MCL) Subsystem, and a PISR Information Base (PISR IB) Subsystem. A Dissemination layer interconnects the MCL and PISR IB subsystems for intelligent control over distribution of information to users on a variety of human interface devices. The PISR architecture has three primary external interfaces: **Sensors/Collectors** for the collection of raw intelligence data; **External Data Interfaces** supplying enterprise situational data from external databases and systems; and **Users** who interact with the system to specify information requirements (e.g., conditions of interest), manage Case Files, administer the PISR System, establish system policies, and receive information products. These subsystems and external interfaces operate within a networking environment (PISR Networking) that is compatible with certified, fielded, and operational USMC networks. Information and communications processing within the PISR System complies with two critical frameworks: (1) the Test, Evaluation, and Certification Framework for ensuring the system performs per user-defined objectives and operational environment requirements; and (2) the Information Assurance (IA) Framework for assured enforcement of balanced need-to-share vs. need-to-protect policies. An additional framework, the Life Cycle Management Framework, defines processes for specifying, developing, fielding, and maintaining versions of the PISR System conforming to this PLA. A brief description of these various aspects of the PISR System is provided below.

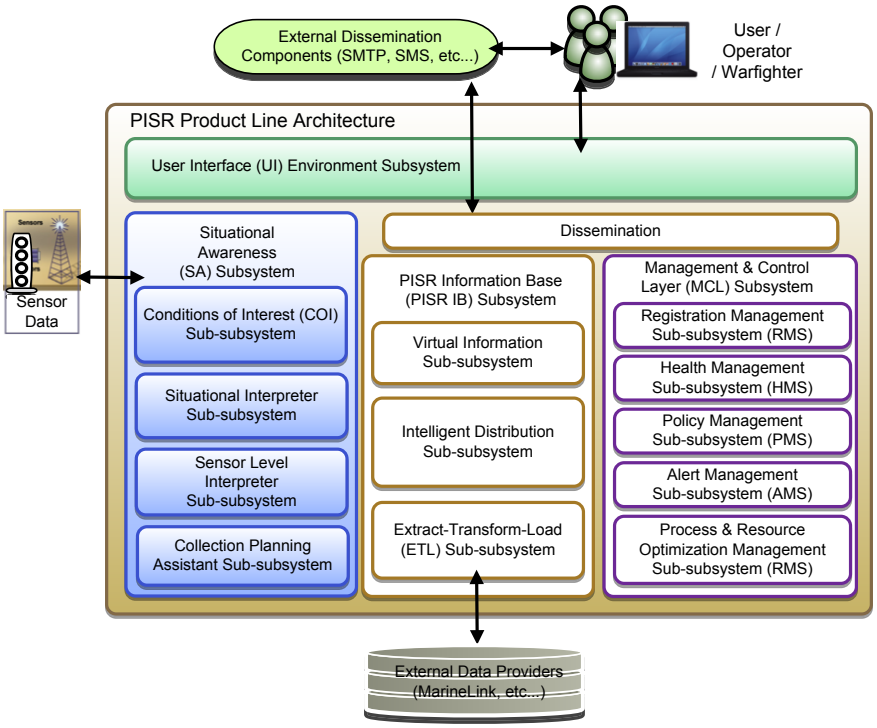

Figure 1. Top-level PISR PLA concept diagram

The ***User Interface (UI) Environment Subsystem*** provides functionality allowing users to interact with the PISR System. The user needs to manage "smart" sensors—the host of devices Marines employ to obtain information from the battlefield, including ground sensors, tower based sensors, and airborne sensors—to obtain timely and actionable intelligence. "Smart" sensors and associated software analytics need to be managed and directed for effective employment in gathering needed information. The user plans the use of these sensors and adjusts their employment to address changing needs. Users must be alerted to events occurring on the battlefield that impact or potentially impact planned and ongoing operations. The user must be able to evaluate and verify information provided from the sensors and analytics. The UI Environment integrates multiple sensor views into common views. The UI Subsystem enables users to organize a variety of data from diverse sources in unified "Case Files." Data are linked in the case files to provide an overall picture of what has occurred and insights into what may next occur. The UI Environment Subsystem meets these requirements by providing the user a web browser based interface for (1) creating flexible and powerful views into employment and operation of the sensors and for (2) intuitive navigation over continuously improved hypotheses and case files encompassing the user's understanding of the battlespace. The UI Environment Subsystem is presented in Section 2 of this document.

The ***Situational Awareness (SA) Subsystem*** provides sensor data processing and reasoning to transform raw sensor information into updated beliefs about the world state. The SA Subsystem is responsible for identifying entity features and behaviors needed to satisfy and inform interested parties. The SA Subsystem comprises the *Conditions of Interest (COI) Sub-subsystem,* the *Situational Interpreter Sub-subsystem, the Sensor Level Interpreter Sub-subsystem(s),* and the *Collection Planning Assistant Sub-subsystem*. The Sensor Level Interpreter is responsible for low-level sensor data interpretation while the Situational Interpreter is responsible for higher-level feature and behavior classification as well as pattern detection. The COI Sub-subsystem processes requests for intelligence deemed valuable by the users. The Collection Planning Assistant aids the user in determining what sensors to employ and where to employ them to best meet collection requirements. The PISR PLA envisions integration of a variety of tools that can be configured to assist users responsible for collection management. Section 3 presents the SA Subsystem.

Dissemination is responsible for defining how the PISR IB and MCL subsystems work together to disseminate valued information expressed in messages and alerts. Dissemination includes the publish/subscribe architecture of the PISR IB, how messages get routed by the MCL, and how alerts are handled. Dissemination includes effective intra-system communication and user notification. The Dissemination section clarifies how the PISR IB will consult MCL dissemination guidance plans to optimize how information flows for effective intra- and inter-system communications. Inter-system communication is facilitated by interfaces to external dissemination components (e.g., mail servers) and interfaces/services provided by external systems. Several examples of dissemination of alerts to users, relying on external dissemination components, are described in that section. Dissemination is presented in Section 4 of this document.

The ***Management and Control Layer (MCL) Subsystem*** is responsible for monitoring the health status of the overall system and for performing the optimization of processes, resources, and information dissemination across the PISR network. The MCL provides plans, guidance, and priorities to the other Subsystems with the objective of maximizing the production and delivery of high-value information in a highly resource-constrained environment. The most constrained resources, in declining order, include human attention, communications bandwidth for mobile warfighters, and time available for adaptive response, among several others. The MCL Subsystem is presented in Section 5 of this document.

The ***PISR Information Base (PISR IB) Subsystem*** provides a variety of data management capabilities to the PISR System. The PISR IB Subsystem supports a semantic fusion model; specifically, providing a shared vocabulary and model of fused entities, which we call the *semantic track model*. The PISR IB Subsystem, through its Virtual Integration Sub-subsystem, contains logic to add value to sensor observations, products of sensor analytics, and other-source data through knowledge of relationships among those data and user information requirements. This sub-subsystem provides data unification across humans and machines, to include operators, automated components (e.g., sensors and analytics), and internal and external information sources. The PISR IB Distribution Sub-subsystem ensures interoperability across the PISR Subsystems by providing efficient distribution of high-value information to different user roles directly or through external systems connected to the PISR System. The MCL Subsystem provides guidance to the PISR IB Distribution Sub-subsystem through a combination of workflow management policies and dissemination planning. The PISR IB Subsystem is presented in Section 6 of this document.

Following the sections describing each PISR PLA subsystem, Section 7 describes ***Key Internal Messaging*** across these components, including detailed walk-throughs of publish/subscribe data flows in the context of specific use cases.

In addition to these principal subsystems, the PISR PLA describes technical and administrative principles that benefit overall system development, accreditation, and deployment processes; specifically, the Rapid Prototyping Process, the Test, Evaluation, and Certification Framework, the Information Assurance Framework, the Life-Cycle Management Framework, and PISR Networking. These are briefly introduced below.

Rapid Prototyping (RapidPro) delivers incremental PISR capability to the U.S. Marines through the use of the PISR PLA. Each delivered PISR System shares a common, managed set of capabilities that comprise the core of the PLA. Additional hardware and software components are added to the core capabilities to meet critical Marine needs. The PISR PLA ***Rapid Prototyping Process*** is described in Section 8.

The ***Test, Evaluation, and Certification (Test/Cert) Framework*** tests and obtains certifications and authorizations for the core components and for any hardware/software added to the PLA to support Marine needs. The Test/Cert Framework tests and validates technical and functional capabilities of PISR components. The framework provides the data and reports necessary to obtain critical certifications to assure that PISR equipment being deployed to the warfighter meets current DOD guidance to be net-centric and interoperable. The framework also provides data necessary to obtain authorizations to connect (ATC) and authorizations to operate (ATO) so the warfighter is assured that the new PISR components are secure, able to operate on classified networks, and cannot be exploited by the enemy. Portions of the test framework are delivered with the PISR Systems to provide a streamlined, intuitive interface for the user to understand and maintain system readiness by identifying, troubleshooting, and resolving system problems. This framework is presented in Section 9.

The Net-Centric Enterprise Services (NCES) Security Architecture describes five basic tenets of Information Assurance (IA): Confidentiality, Integrity, Authentication, Non-repudiation, and Availability. The ***Information Assurance Framework*** describes various IA and security considerations PISR Systems must address in order to complete certification and accreditation at an accelerated rate. This framework is introduced in Section 10.

The ***Life-Cycle Management (LCM) Framework*** describes processes and tools for using the PLA to develop systems from components and for evolving those components and systems. LCM is an integrated, collaborative approach addressing configuration management and software product development from application creation to demise. Without the LCM it will be much more difficult, expensive, and time-consuming to develop and maintain a coherent, compatible product line for PISR. The PISR System LCM enables effective systems management and evolution, and provides information for future integration with related systems. Development of the initial PISR system is an engineering challenge because of the project scope and the rigorous testing/validation requirements, as well as the focus on a product line approach. An effective LCM toolset supports this effort through improved understanding of existing systems and effective documentation of software products. The LCM Framework is described in Section 11.

PISR Networking provides inter-system and intra-system connectivity needed to integrate the PISR System into the Marine operational environment. At the battalion level and below, robust, ubiquitous, ad hoc, mobile mesh networking clusters will constitute the core for PISR intercommunications. Within the clusters, operators, unattended sensors, and aerial and ground manned/unmanned surveillance nodes (towers, UAVs, UGVs, surveillance aircraft, ground vehicles, ground stations, etc.) will maintain the self-forming networking by controlling their location on-the-move. They will also maintain the application load, subject to current terrain and node availability constraints, to address user-specific information delivery requirements. PISR Networking is described in Section 12.

1.5 PISR Users and Execution Environments

PISR Systems will operate in an execution environment supporting a specific set of user roles, as shown in Figure 2. The four different classes of users are:

1. **End-users,** primarily composed of USMC intelligence and operations personnel, are simply referred to in this document as Users. Users are generally the consumers and producers of information in the field.
2. **Commanders, or Policy Makers**, are end-users who set goals and policies for Users. Commanders or Policy Makers are generally at the same or higher echelon as Users. Commanders usually consume information generated by Users and other systems on the PISR network and make sure that the right goals are being set and accomplished.

3. **Administrators** are generally not end-users, but rather the staff that make sure the business rules in operation by the system are performing properly and the system is configured appropriately for the task at hand in the operational environment.
4. **Developers and Maintainers** provide new and improved components for the system to employ as new requirements, quality attributes, and bugs are discovered and addressed through associated development processes.

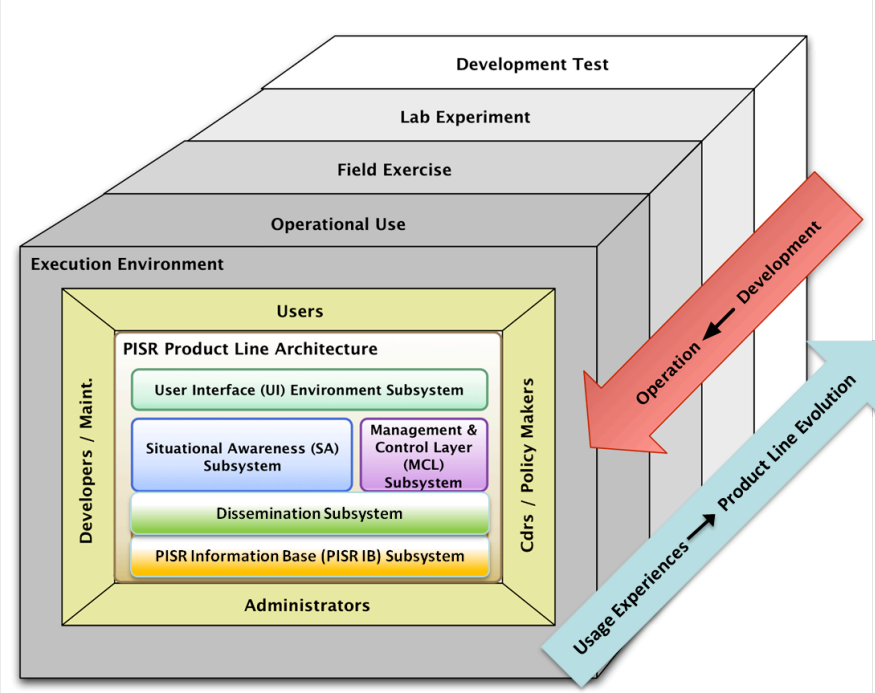

Figure 2. PISR System execution environments

The PISR System Execution Environments shown in Figure 2 represent the contexts in which the PISR capabilities operate. Each PISR product version progresses through four primary execution environments: Development Test, Lab Experiment, Field Exercise, and Operational Use. By utilizing these different execution environments, users of the PISR System progressively gain confidence in its operation. The environments also provide a structured progression from Development to Operation, designed to mature and evaluate each PISR System.

The **Development Test Environment** provides a way to simulate, stimulate, and assess the performance of components, subsystems, and the entire system while in development. The Development Test Environment supports robust regression testing so that the functionality and performance of additional or modified components can be automatically verified.

The **Lab Experiment Environment** provides ways to simulate, stimulate, and assess the performance of the system in hypothetical operational settings. This environment enables Developers to preview the actual performance of a system or component within the field and to collect feedback from other users as to how well the system or component matches the needs or expectations of the target users. This environment facilitates the generation of canned data from simulated live execution of the system. The environment also tracks generated canned data to be used as performance tests against future iterations of the system or component.

The **Field Exercise Environment** provides ways for subsystems or entire systems to be tested against a combination of live and canned data. The Field Exercise Environment stimulates the system and simulates key aspects

of expected operational settings. Information from a field exercise can be used to derive canned data for the Lab Experiment Environment.

The **Operational Use Environment** is the target environment for a PISR System. When a system is deployed for operational use, it is continually under test by the environment using actual real-time data. Test and Validation capabilities are transformed to monitoring Health, Status, and Policy-regulated behavior. Monitoring is in place so that the system can be verified while running, especially during the introduction of new components. Operational data can be captured for use in Field Exercise or Lab Experiment Environments for future component development. The Operational Use Environment provides roll-back capabilities so that the system can revert to a previous state if changes affect the system in an unexpected or problematic way.

Particular PISR systems exhibit many behaviors and qualities when tested or employed. Some of these will motivate change requests or instigate activities that cause improvements in the system components or the architecture itself. The lower arrow from left to right in the figure illustrates this basic idea that the PLA evolves in response to usage experiences. The PLA approach anticipates and supports this need for continuous improvement. When improvements occur in generic frameworks and components, they can yield improvements in all specific systems that incorporate those elements.

1.6 Optimizing Resources, Information Value, and Information Needs

The PISR architecture aims to deliver highly valued information as defined by operators. The value of information is ultimately measured in threats thwarted, opportunities exploited, and missions accomplished. The PISR PLA approach first defines "information value" objectively. For example, avoiding friendly fire might be the highest priority, interdicting IEDs might be the second priority, and maintaining situational awareness might be the third priority. Given this user-defined value hierarchy, we can select user-defined conditions of interest related to those concerns and assign weighted, but otherwise arbitrary, value scores to information on that basis. A set of management and control mechanisms optimize the collection, production, and dissemination of information. The optimization maximizes the delivery of highest-valued information while carefully tasking scarce resources given information needs, estimated information value, resource status, policies, and constraints. For example, the optimization might suggest not deploying available resources to achieve maximum coverage across the area of interest. Rather, the resources should be asymmetrically concentrated to cover locations of Blue and Red force concentrations, Red Force CONOPS, and transportation grids, to provide the best probability that the highest priority conditions of interest will be detected. Under such management and control mechanisms, the system will apply surplus resources to work on a broad set of efficient intelligence gathering and background processing activities that collectively improve readiness for future tasks. The key considerations for these management and control mechanisms include:

- Overall process flow and resource allocation aims at assuring high-value information is produced and delivered in a timely way.
- Scarce resources are allocated to the items of highest estimated information value with greatest possible efficiency and effectiveness to provide the greatest resulting mission value.
- Moving information around consumes communication resources and time, so optimizing dissemination and routing impacts the information value delivered to the final point of consumption.
- Where resources can contribute to satisfying multiple information needs, opportunities arise to economize through shared allocation for collection and processing.

These considerations allow us to break the management and control process into a set of three balanced optimization functions:

- Optimizing the allocation and scheduling of collection resources associated with producing the information.
- Optimizing the process workflow for processing collected information to extract the features and behaviors required to potentially satisfy expressed information needs (e.g., conditions of interest).
- Optimizing the dissemination of information products to appropriate users.

There are strong linkages and interdependencies between these three optimization functions. To provide the greatest possible value to the warfighter, the PISR System must continuously monitor, evaluate, and adjust its operation to maximize the production of information value in an evolving, resource-constrained environment. It does this through the concept of information value, specifically focused on production, dissemination, and consumption of highest-valued information.

The PISR PLA provides a foundation for improved intelligence processing in contexts where our warfighters have limited resources but face potentially overwhelming amounts of data. This motivates our focus on highest-valued information. PISR Systems incorporate a capability to estimate the time-based value of collecting, processing, and disseminating information. Information needs expressed by the users and Commanders, considering the relative prioritization of units, tasks, and missions, combine to determine how much value each type of information will contribute. Estimating information value is a key element of the collection planning, resource allocation, and dissemination planning performed by the MCL and implemented in the various PISR subsystems. By continually assessing expected information value in the dynamic context of the battlespace situation and resource status, the PISR System allocates its scarce resources in a manner likely to achieve the goal of delivering to the warfighter high value in a very constrained environment. Understanding the time-value of information both ensures timely delivery of high-value information and protects the system from expending critical resources on superfluous, stale, or obsolete information.

1.7 Case Files and Case File Support

While persistently updated sources of intelligence from sensors and reconnaissance teams form a critical part of PISR, less frequently updated repositories of intelligence data provide value as well. In particular, intelligence analysts individually and collaboratively maintain information on key entities such as individuals and organizations. This data accumulates gradually over time and is archived indefinitely in *case files*—one per person, organization, or other entity. Much of the data consists of unstructured text as well as video or audio clips.

Despite the generally unstructured nature of case files, some structure does exist. Usually case files on individuals contain the last known position location information. Analysts can also tag case files with keywords to facilitate search or to place an individual on a warning, threat, or watch list. Executive summaries of case files contain other fixed fields such as names and aliases, height, weight, age, and gender, which are commonly known. Case files can also contain links to other case files such as links to relatives or to organizations to which an individual belongs. References to external data sources appear in case files as well. Case files are stored in the PISR IB Subsystem for query, access, update, distribution, and archival storage.

The PISR System helps automate case file management in a number of ways. Since case files are developed collaboratively by a number of analysts, the PISR System can support concurrent editing of the file by multiple authors (e.g., via a virtual whiteboard system including a chat room). Analysts or other PISR users interested in a particular individual or organization can request notification of case file updates and receive automatic alerts when new information is received by the system. Depending on their authorization levels, users can track changes to a case file, supersede changes, and view which contributor made any particular change. Case files can be searched by values within fixed fields or by various tags. In addition, the unstructured text and other content within a case file are searchable as well.

Case files can also integrate with the more automated portions of the PISR System. Tracked entities will have their case files automatically updated to reflect changes in reported location. For example, face recognition or other biometric analytics can update position location information for individuals associated with case files. Analytics that interpret unstructured text potentially trigger other analytics within a PISR System to update various parts of a case file.

Case files are routinely used to manage processes that must be agile and ad hoc. New data in a case file can trigger an analyst or some automated analytic process. These triggering states can lead to automatic generation of emails, data collection requests, or other collaborations. The PISR PLA will incorporate generic case file management capabilities that make it easy for users to define states of case files that warrant attention and that might dictate routine automated or collaborative tasks. Entities in case files might also be associated with conditions of interest relating to specific user information needs, as elaborated in the next section.

1.8 The Language of COIs and Information Needs

Conditions of Interest (COIs) describe high-value events and, thus, can define high-value information. There are many types of COIs, but to a large extent they correspond to "threats" and "opportunities." In this context, a threat means something that portends a surprisingly bad outcome. An opportunity, on the other hand, offers the chance for a surprising good outcome. In modern information systems, people are overwhelmed by a glut of data. The PISR PLA uses automated COI monitoring to find events that correspond to these threats and opportunities. Threat COIs fall into two basic categories: instances of enemy tactics, techniques, and procedures (TTPs); and surprising events that invalidate mission plans by undercutting plan assumptions or prerequisites. Opportunity COIs likewise fall into two basic categories. The first corresponds to enemy positions or situations that expose unforeseen vulnerabilities. The second comprises surprising events that transform potential Courses of Action (COAs) from disfavored to preferred.

For example, assume that a particular squad's first priority is interdicting known HVIs, and its second priority is house-to-house "sanitization" in a particular village. Absent HVI location data, the squad leader's planned COA is to proceed across the sanitized southern bridge and conduct south to north house to house search. COIs might include indicators that the southern bridge is no longer safe, and that an HVI has been located in the squad's area of responsibility. Upon receipt of the first COI, the squad leader might avoid the threat by a different route to avoid using the bridge and altering the COA to begin the search from north to south. Upon receipt of the second COI, the squad leader would abandon the sanitation COA, and commence an HVI interdiction COA. The PISR PLA enables operators to define such COIs and delegate to the PISR System the tasks of finding the matching events and alerting the appropriate people or agents.

If everything went according to plan Marines would always succeed at every mission. Activities of both adversaries and neutral forces, such as civilians, or weather can interfere with plans if not anticipated correctly. Before the battle begins, mission planners predict as many threats to the success of the mission as practical. Despite consulting many sources of information about historical behavior of adversaries and role playing through "what if" scenarios, surprises are always common as the battlespace evolves over time.

Potential threats to a mission identified during mission planning should become COIs. The PISR System to continually monitors the battlespace for events that match COIs and notifies the appropriate personnel when those events occur. If the PISR system delivers the right information to the right people (or machines) at the right time, friendly forces will have time to react to new developments in an informed manner. Freeing personnel from sifting through all data to find these events increases the likelihood that our people will have available attention and time to respond quickly and effectively to events that really matter to the operation.

In addition to monitoring for threats, COIs also monitor for opportunities. Too many times high-valued targets and other wanted individuals have been stopped and questioned over a minor infraction or at a checkpoint only to be released. In such cases, the detainee's biometric data (for example) should trigger a PISR COI for an HVI that would alert the detaining Marines to the person's importance. Marines on patrol also can benefit from recent reports of suspicious activities in their area. Maximizing the value of the information enables friendly forces to maximize the value of their operations.

1.9 Distributed Operation and Control

Management of distributed operations in the tactical battlespace is critical to the success of PISR Systems. PISR Systems will operate in a highly distributed and dynamic network of shared and taskable resources. Just as the exponentially increasing volume of available data drives architectural controls to optimize information value, the geometrically increasing availability of distributed ISR resources drives architectural controls to optimize their operational deployment. Today, much of the required control is achieved by direct human oversight and the localization of assets to specific units. Control will become more challenging as the operational area for PISR becomes more complex, the sensing capabilities of collection assets become more diverse, and the volume of information increases.

In anticipation of this transformation, the PISR PLA considers and supports dynamic management and control of PISR System nodes and Subsystems. The MCL Subsystem provides for distributed health status monitoring, and the reconfiguration and control of sensors and PISR nodes throughout the network. The Dissemination Subsystem provides the efficient movement of information based on prioritization and value, aided by MCL's cognizance of the availability and health status of network resources. Collection assets are deployed and operated in accord with a collection management plan. Similarly, the SA Subsystem employs its analytical resources to perform the highest-valued activities

in accord with pending COIs and other Information Requests. Both collection plans and processing plans are under the control of MCL which continually aims to optimize value produced by constrained resources.

The PISR IB Subsystem provides a shared distributed blackboard to directly support information sharing and satisfaction of *ad hoc* information queries. Because the PISR IB appears local to its clients, this distributed blackboard simplifies many potential complexities that would otherwise arise from the distributed nature of the clients.

1.10 Stakeholder Quality Attributes

Stakeholder Quality Attributes (QAs) describe the PISR Systems' desired and intended behavior with regard to how the target audience is going to use and support the system. Based on prior fusion systems and in consultation with USMC stakeholders, we collectively developed a list of 143 desirable QAs, falling into 18 categories, and illustrated each QA with a use-case scenario. For example, the first category was defined as "Case File creation/modification/use", and this category contained 12 candidate QAs. Below is the first entry in that category:

CAT #	QA #	QA	Scenario
1	1	PISR System supports the creation of Case Files about entities of interest. Case Files contain all information input from the user who opens the Case File including audio files and video clips.	User selects "Case File Creation and Maintenance". User Alpha creates a Case File and identifies Mullah Mohammad Rabbani as a Person of Interest (POI). User enters important information about this person, including who, when, where information was collected. User inputs an audio recording of an interview with Rabbani.

The entire list of candidate QAs was distributed by MARCORSYSCOM throughout relevant parts of the MCISR enterprise. Each voter was allotted 44 votes that could be allocated among one of more QAs in any manner the voter deemed most important. This process has been used in many architecture efforts to align the full spectrum of stakeholders and help provide a shared commitment to consensus priorities. After the voting process, 1188 total votes were cast, and the QAs clustered into three priority tiers, labeled A, B, and C. According to best practices, the C category was required to include at least 1/3 of the rated items and priority A could not be assigned to more than 1/3 of the items. The A-priority items become the principal focus for the first version of the PISR PLA. Any item that received 10 or more votes became an A-priority. The four highest rated items in the A-priority category received more than 40 votes each. The full set of 47 A-priority items is listed in Appendix A. As the PISR PLA matures, techniques to maintain continuous interaction with the operational customer community will expand and improve.

1.11 PISR System Production Process

The definition and refinement of the PLA is just the first stage of a larger business process that must be agile, predictable, and repeatable in order to successfully deliver high value to the operational warfighter through frequent, evolutionary product releases.

This production process, seen in Figure 3, receives direction from stakeholders and policies and is shaped by the technologies managed by existing Programs of Record (the "brownfield environment"), new capabilities from DoD research laboratories (including Science & Technology (e.g., S&T)), and the continually expanding inventory of available off-the-shelf products.

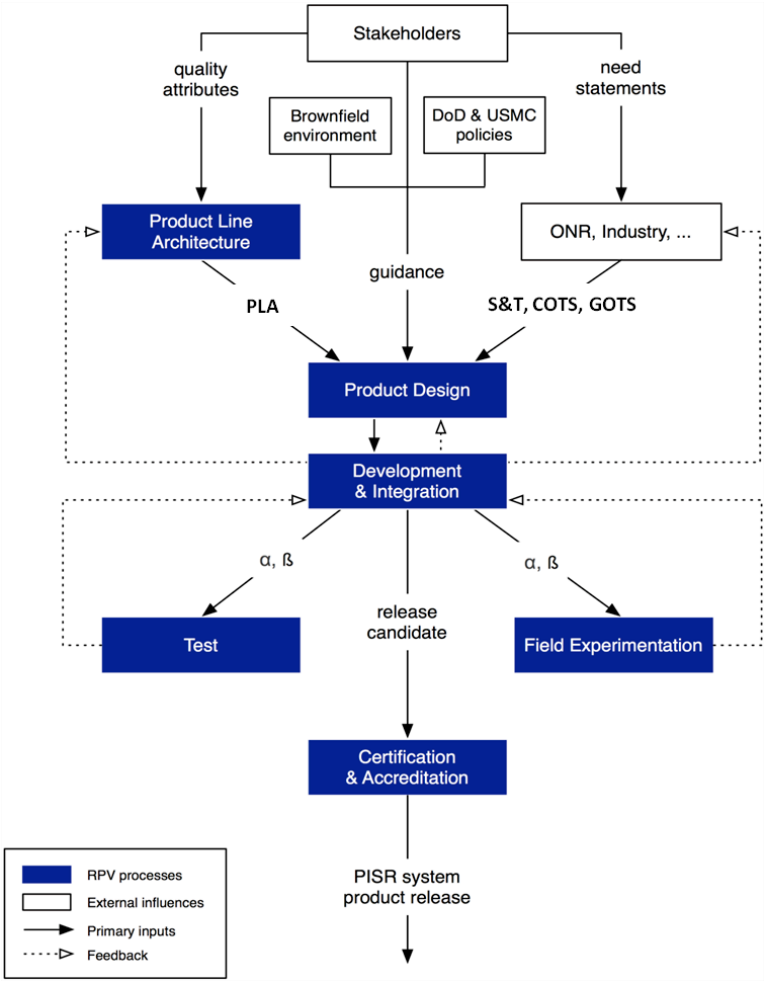

Figure 3. PISR System production process

The production process can be executed in parallel iterations, as shown in Figure 4. Sub-processes of particular note within the PISR system production process include: Product Line Architecture (PLA), Product Design (PD), Development and/or Discovery (of off-the-shelf capability) & Integration (D&I), and Test (T). Other stages include Field Experimentation (FE) and Certification & Accreditation (C&A).

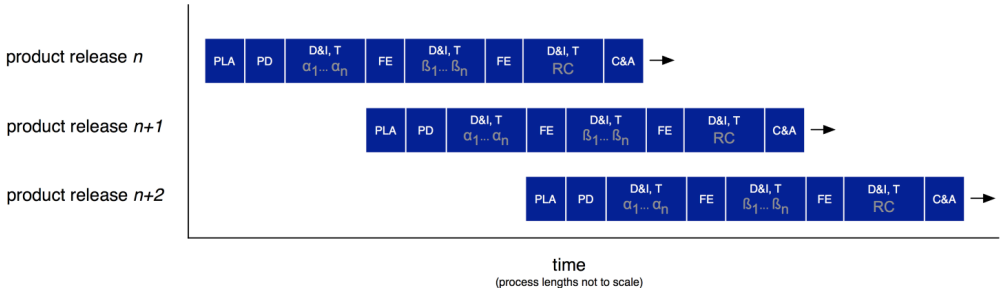

Figure 4. Pipelined, evolutionary product releases

1.11.1 Product Line Architecture Sub-process

The PLA sub-process begins by an initial formulation of desired quality attributes (QAs)[18] of the product line. Associated usage scenarios provide context for each quality attribute. An initial assessment of the perceived user value and technical risk is also assigned to each QA. For example:

Perceived value	Technical risk	Quality attribute	Usage scenario
High	Low	PISR System supports the creation of Case Files about entities of interest. Case Files contain all information input from the user who opens the Case File including audio files and video clips.	User selects "Case File Creation and Maintenance". User Alpha creates a Case File and identifies Mullah Mohammad Rabbani as a Person of Interest (POI). User enters important information about this person, including who, when, where information was collected. User inputs an audio recording of an interview with Rabbani.

The initial set of QAs is presented to the PISR System stakeholders, who discuss, refine, and add new quality attributes. Stakeholders then vote for the QAs that are most important to them. The top ranked QAs (at most 1/3 of the set) are identified as "A" priorities and directly guide the product line architecture and later sub-processes.

Finally, a high-level design for the product line is defined, composed of generic components with well-defined relationships and responsibilities, capable of meeting key stakeholder goals and accomplishing the primary usage scenarios.

The resulting software architecture, while idealized, is suitably detailed and implementable so it can pragmatically guide and bound subsequent product releases. While the PLA should be forward thinking, anticipating technologies and capabilities beyond three years is quite difficult. The PLA produced by this sub-process is credible and void of "magic".

1.11.2 Product Design Sub-process

In Product Design, a PISR System for a particular environment, scale, timeframe, cost, and set of "A"-priority quality attributes is designed in detail, using the PLA as a template.

Such a product can be delivered rapidly through predominantly existing off-the-shelf products or near-fieldable capabilities to instantiate the generic components defined in the PLA. This includes selecting what data to

[18] Jan Bosch, *Design and use of software architectures: adopting and evolving a product-line approach*, ACM Press/Addison-Wesley Publishing Co., New York, NY, 2000.

collect and what analytic processes to configure. Designers in this sub-process play roles analogous to that of a sommelier ("wine steward")[19]. A sommelier:

- collects from the finest sources
- has long-standing partnerships with the vintners
- understands the best "pairings"
- can make well-informed recommendations
- creates combinations that "complete" the meal
- tastes the wine to ensure quality
- doesn't drink the wine
- doesn't own the wine
- ensures the wine is quality and is available on demand

To produce a product design, the PISR System designers must investigate available commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) capabilities, Program of Record (PoR) technologies, and DoD research to assess their compatibility with the PLA vision, stakeholder QAs, and high-priority technical QAs such as ones related to usability, reliability, and robustness. These technical QAs augment the set of stakeholder QAs, and they require an additional cycle of QA and scenario elaboration and voting. The stakeholders for the technical QAs include developers, maintainers, and test personnel.

Additional considerations arise depending on who has created and is supporting an implemented capability, as listed below:

PoR technologies	• Is it used or ignored by the targeted warfighter group? • Is it on the way out and being replaced by something newly favored?
DoD research & Government off-the-shelf	• Has it proven its warfighter utility in a relevant environment (Technical Readiness Level 6)? • Does it significantly replicate functionality that is readily found in industry products at a lower cost of ownership? • Is its lifecycle credibly supported? • Does it bundle via open standards?
Commercial off-the-shelf	• How responsive is support (e.g., correcting issues discovered during integration, lifecycle model)? • Does it promote vendor lock-in thereby reducing portability, composability, or extensibility? • Does it bundle via open standards?
Open-source off-the-shelf	• Does it have a large enough community to sustain itself in absence of a corporation? • How complete is its documentation? • Does it bundle via open standards? • Does its license have undesired implications?

[19] Phillip C. Chudoba, COL USMC (retired), previously Program Manager, Intelligence Systems, Marine Corps System Command, Enabling Persistent ISR for the Warfighter, 2010.

The deltas between what is required by the PLA QAs and the existing solutions become feedback to DoD research laboratories and industry. The Development & Integration sub-process offers one alternative for satisfying deltas requiring modest development.

The product design sub-process must be flexible enough to adapt to changing circumstances; for example, a performer experiencing difficulty in a field exercise, or an opportunity to dramatically improve capability via a newly discovered commercial product.

Lastly, the product design produced should be as simple as possible, yet still deliver the desired quality attributes and functionality.

1.11.3 Development & Integration and Test Sub-processes

To accelerate product development, so that products reach users in months rather than years while achieving high levels of quality, the development sub-process implements many of the principles and practices advocated by the agile software development community. As development teams vary widely in experience and preferences, they should tailor the agile principles and practices below to best suit their circumstances.

Principles[20]	• Test based design • Customer as full partner in engineering process • Iterative and incremental development via sprints that deliver small amounts of tested, ship-ready code • Regular adaptation to changing circumstances • Working software is the primary measure of progress • Simplicity — the art of maximizing the amount of work not done — is essential • Sustainable pace
Practices	• Unit testing, fuzz testing • Continuous integration • Extensive code coverage • Code standards • Effective, minimal-overhead metrics (e.g., Scrum burn-down-chart) • Customer interaction

While the product design shapes the technical aspects of the development sub-process, it also determines how much work will be required. Within Development & Integration, the required time is divided into 4-week chunks called *sprints* or *iterations*.

Through sprints, regular checkpoints are defined to assess progress, handle business process "exceptions", re-prioritize tasks, obtain stakeholder feedback, and motivate implementers with goals that are within sight. At the start of each sprint, a subset of features/issues associated with the current product release is assigned to the sprint for completion within the 4-week time frame. As features are implemented, white box tests are created and executed automatically on a regular basis. Partially automated black box acceptance tests (that exercise the entire integrated system) occur as frequently as possible, minimizing the testing required at the end of each sprint. The end goal of each sprint is stable, tested software; perhaps not feature-complete, but incrementally approaching a releasable product.

Further details concerning the Test process and its relationship to the Certification and Accreditation process are provided in Section 8.

[20] From the Agile Manifesto: http://agilemanifesto.org/principles.html

1.12 PISR Functional Use Cases

Discussions with USMC infantry and intelligence subject matter experts (SME) revealed that the high frequency missions in theater today typically involve security patrols, intelligence patrols, and information operations (IO) patrols. Marines say that if PISR is to be valuable to them at echelons of battalion and below, the system must monitor, detect, and alert users of (1) IED emplacement, (2) HVI location/relevant information, and (3) patrol/convoy ambush, as well as sneak attacks on FOBs. As missions and CONOPS rapidly evolve, the scenarios and mission threads of interest will likewise evolve. However, even as the operational view (OV-1) is presented it becomes obsolete; however, the general approach to analyze functional use cases will persist. Two of these scenarios (IED and HVI) illustrate PISR System deployment for a particular tactical environment. There will never be sufficient PISR assets to surveil the entire MAGTF AOI. Therefore, the scenarios assume that the MAGTF Commander has developed his Commander's Critical Intelligence Requirements (CCIRs), designated his Named Areas of Interest (NAIs), and approved the intelligence collection plan. These scenarios (see Figure 5) are focused within a notional NAI where "battalion level and below" tactical operations are supported by a system of PISR assets.

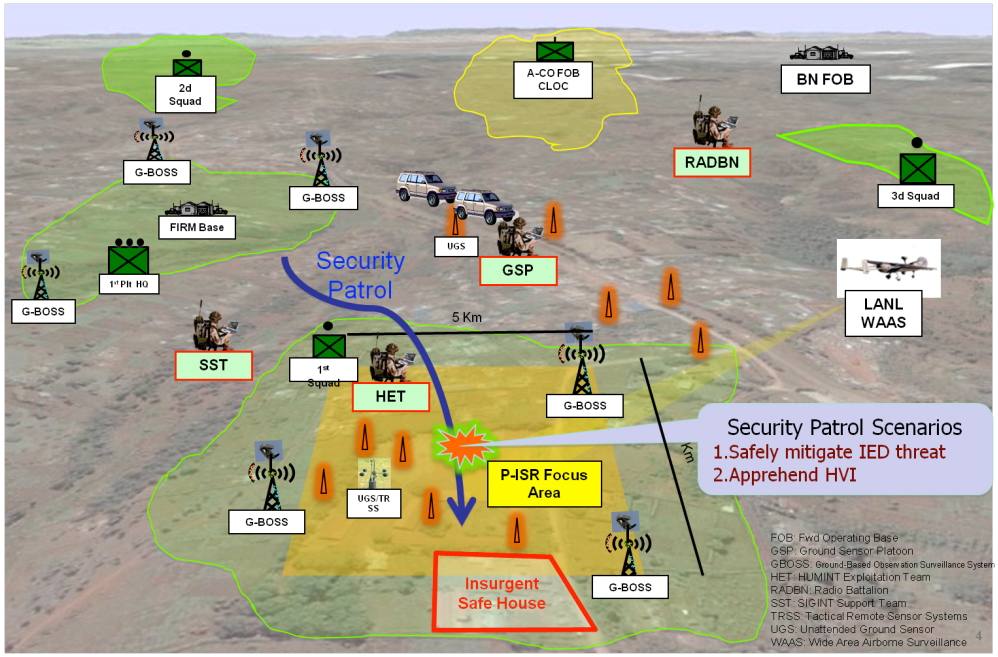

Figure 5. Overview of RapidPro PISR scenario space

The following subsections describe the IED and HVI use cases in short vignettes followed by a description of battlefield activities and associated PISR System activities that will occur in the context of those vignettes. These provide a general idea of PISR System functionality. They also provide an operational backdrop for discussions of the PISR PLA subsystems that occur in later sections of this document. The intent here is to provide the reader with enough general understanding of major components to set the stage for a more detailed description in Section 7 of how data and messages flow within and between components to perform overall tasks. The vignettes are not actual situations. Rather, they only indicate what could happen in a real operation. The reader will find that the second vignette is very similar to the first. This is an important observation: *The capability of the PISR System (built from the generic PLA described in this document) to deal with diverse situations using the same set of user and computational features is a strong validation of the principles embodied in the architecture.*

1.12.1 IED Emplacement Use Case

IED Condition of Interest Scenario (**Figure 6**); i.e., ***Notify me if an IED is emplaced along my route, in my AOR, in destructive range of blue/coalition/host nation forces.***

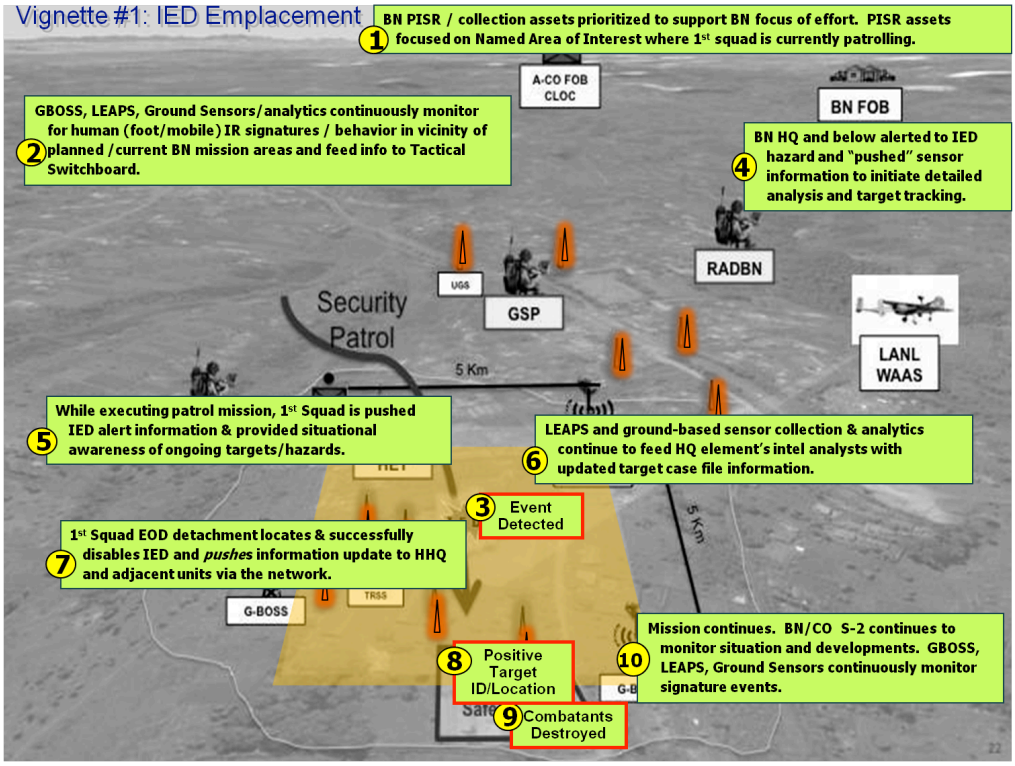

Figure 6. PISR IED scenario

1.12.1.1 Vignette

A-Company, 1st Platoon, 1st Squad's mission is to conduct security patrols in Green province. This province is the battalion's NAI based on reports collected from Human Exploitation Team (HET), Signal Support Team (SST), and RADBN over the past 96 hours. 1st Squad is to intensify its patrols and continue providing updated reports on Green activity. Collection efforts are focused on Green province and sensor assets are aligned to mutually support BN collection efforts as well as disseminate valuable information to the appropriate lowest echelon of execution. In advance of the patrol mission, GBOSS, ground sensors, and WAAS IR sensors and analytics focused on 1st Squad's projected patrol route. Sensors are configured to locate, track, and predict human IR signatures indicative of IED emplacement. These assets are also in direct support of the squad at time of execution. As 1st Squad departs the FIRM base and crosses the line of departure (LOD), PISR System services running on the GBOSS sensors identify anomalous human signature behavior 1 km ahead of the platoon and alert WAAS and UAV platforms for further investigation. WAAS identifies and tracks the potential insurgents and geo-rectifies the hazard location. Immediately, a hazard alert text is pushed to 1st Squad's Platoon command elements, as well as to echelon HQ elements up the chain. By now video is flowing into the BN CP and Intel analysts are engaged in analytic behavior tracking of the suspects. As 1st Platoon nears

the suspected IED site, an Explosive Ordnance Demolition (EOD) detachment positively identifies the bomb and detonates it. HET teams on scene obtain information from the local population that identifies the suspects. These reports are immediately recorded and sent to Company and BN S-2 for correlation with the ongoing sensor collection efforts. The fusion of real-time intelligence with WAAS /UAV target tracking produces a mature target package for immediate Close Air Support. A section of on-call Cobra attack helicopters are then dispatched. The target location is positively identified. Sensor tracking and 1st Squad reconnaissance elements verify no friendly forces are in the vicinity of the target. AH-1Z Cobra attack helicopters roll in and the enemy combatants are destroyed.

1.7.1.1. Activities

1a. Battlefield Activity: BN PISR / collection assets are prioritized to support the BN focus of effort. PISR assets are focused on the NAI where 1st squad is currently patrolling.

1b. PISR System Activity: Prior to this point in time, an Intelligence Analyst would have used the PISR System to build a Collection Plan and would have sent that plan forward for approval. When approved, the Collection Plan would have directed GBOSS, WAAS, and Ground Sensors to deploy to the NAI to provide continuous monitoring. An analyst would have entered into the PISR System a situational trigger (i.e., a condition of interest) to provide an alert if anomalous human activity is detected within the NAI along a route being used by a convoy or patrol. The analyst associates two additional requested actions with the COI: (a) alert BN HQ and below, and (b) produce a summary sensor report to include with the alert.

2a. Battlefield Activity: GBOSS, WAAS, Ground Sensors continuously monitor anomalous human (foot/mobile) IR signatures in the vicinity of planned/current BN mission areas and feed information to Tactical Switchboard.

2b. PISR System Activity: The PISR System is connected to Tactical Switchboard through the Situational Awareness (SA) Subsystem. GBOSS and WAAS are also connected to the PISR System through SA. Other non-real-time information feeds such as MarineLink and Global Command and Control System (GCCS) are connected through the PISR IB Subsystem. The PISR System is looking for information that will satisfy the COI created in Activity 1b.

3a. Battlefield Activity: A vehicle stops at the side of the road for 20 minutes. Two individuals leave the vehicle, dig on the side of the road, and reenter the vehicle. The vehicle leaves the area at high speed.

3b. PISR System Activity: Event Detected; the PISR System identifies the activity in 3a as activity that satisfies the COI created by the analyst.

4a. Battlefield Activity: BN HQ and below are alerted to likely IED hazard and "pushed" sensor information to initiate detailed analysis and target tracking.

4b. PISR System Activity: The analyst who created the COI is immediately alerted with a flashing message on his computer monitor of an event matching his COI. The system executes the action associated with the COI; i.e., alerting that Battalion Headquarters (BN HQ) and below when the COI is satisfied. The PISR System prioritizes the alert so that communication resources are available to send the alert to BN HQ and subordinate elements. The COI-associated actions also requested that a summary report of the sensor information be included with the alert. The PISR System packages a summary of the sensor information with the alert. Several other COIs that are active requested that their authors be sent any alerts dealing with IEDs. These users also receive the alert and begin doing additional research to understand and mitigate the threat. Standing Operating Procedures (SOPs) for this Battalion require a call list of key personnel be alerted when a suspected IED is discovered. The PISR System retrieves this list from PISR IB and uses the Dissemination Subsystem to phone all the people on the list. If a person fails to answer the phone, the PISR System automatically sends a text message and an email to the person. The PISR System continues to monitor communications to know when all persons on the list have been notified and have acknowledged the alert.

5a. Battlefield Activity: While executing the patrol mission, 1st Squad is pushed IED alert information and provided situational awareness of ongoing targets.

5b. PISR System Activity: Actions associated with the satisfied COI require that any Marine units operating within 1.5 km of the suspected IED site receive the alert. 1st Squad is the only unit within the 1.5 km circle so they are pushed the alert through the digital tactical radio system. The PISR System monitors the communication system to determine if 1st Squad responds to the text message. 1st Squad does respond.

6a. Battlefield Activity: WAAS and ground-based sensor collection and analytics continue to feed HQ intelligence analysts with updated target case file information.

6b. PISR System Activity: Additional COIs are created to look for additional enemy activity that may be associated with the IED emplacement. PISR System users monitor WAAS and ground-based sensors looking for additional evidence of enemy activity.

7a. Battlefield Activity: 1st Squad EOD detachment locates and successfully disables IED and pushes information update to BN HQ and adjacent units via the network.

7b. PISR System Activity: 1st Squad EOD sends out a message that the IED has been destroyed. The analyst who initiated the COI is notified. The analyst uses the PISR System to enter the outcome of his COI but decides to leave the COI active so it can detect additional IEDs in this important NAI. An IED Report is filed by 1st Squad into MarineLink. Through its external interface to MarineLink, the PISR IB obtains the information about the IED and adds the information to the PISR System IED incident archive.

8a. Battlefield Activity: Positive target ID/location.

8b. PISR System Activity: Human Intelligence reports flow into the COC indicating that combatants that implanted the IED have been identified and their location is known. These reports are detected by the PISR System and information from the PISR System helps in developing a mission plan to engage them. Analysts and other users who received the alert use the PISR System and other COC resources to plan a mission to attack the combatants. Through its virtual information base integrating information from multiple sources, the PISR System supports the fusion of real-time intelligence and non-real-time intelligence to support mission planning and to improve situation awareness.

9a. Battlefield Activity: Combatants destroyed.

9b. Battlefield Activity: The PISR System is used with other tools to inform users that the combatants have been identified, located, and that little collateral damage will result from an immediate attack on the combatants. The mission is planned and executed and the combatants are destroyed. Analysts and other users connected to the PISR System update their case files with this new information. The PISR System monitors several other data sources including MarineLink and pulls information from those sources into the PISR IB for future use.

10a. Battlefield Activity: Mission continues. BN/CO S-2 continues to monitor situation and developments. GBOSS, WAAS, and ground sensors continuously monitor events.

10b. PISR System Activity: PISR System continuously looks for events that satisfy active COIs and continues to support users through information flow to and from the PISR System User Interface Environment Subsystem.

1.12.2 HVI Use Case

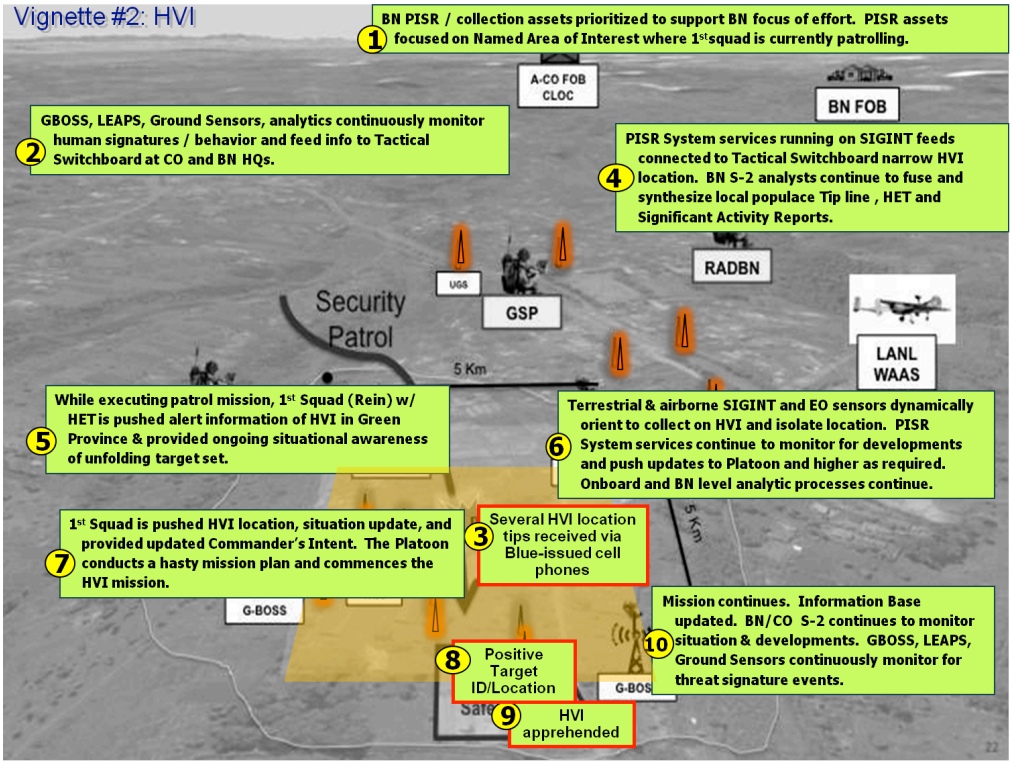

Figure 7. PISR HVI scenario

1.12.2.1 Vignette

A-Company, 1st Platoon, 1st Squad's mission is to conduct security patrols in Green province. This province is the battalion's Named Area of Interest (NAI) based on reports collected from HET, SST, and RADBN over the past 96 hours. 1st Squad is to intensify their patrols and continue providing updated reports on Green activity. Collection efforts are focused on Green province and sensor assets are aligned to mutually support BN collection efforts as well as disseminate valuable information to the appropriate lowest echelon of execution. In advance of the patrol mission, GBOSS, ground sensors, and WAAS IR sensors and analytics are focused on 1st Squad's projected patrol route. Sensors and analytics are configured to locate, track, and predict behavior of the HVI and his associates. HET and patrol significant activity (SIGACT) reports and debriefs are entered into tailored handheld device applications that transmitted the updated info to HHQ for ultimate entry into Marine Link. Tip lines (via cell phone and other means), biometrics, and RADBN data are analyzed and fused to assist in HVI location refinement. The PISR System monitors sensor data, intelligence sources, and concurrent intelligence analytics (human and machine-assisted) to derive high-confidence HVI location coordinates. High-value alerts and significant mission status updates are pushed to 1st Platoon until the execute order is given. Upon apprehension of the HVI, field reports sent via the network continue to be submitted and the PISR Information Base updated.

1.7.1.2. Activities:

1a. Battlefield Activity: BN PISR collection assets are prioritized to support BN focus of effort. PISR assets focus on the Named Area of Interest where 1st Squad is currently patrolling.

1b. PISR System Activity: Prior to this point in time, an analyst has used the PISR System to build a collection plan, sent that plan forward, and the plan was approved. GBOSS, WAAS, and ground sensors have been deployed to the NAI as planned to provide continuous monitoring. An analyst has entered into the PISR System a COI to alert him if a specific HVI is detected within the NAI.

2a. Battlefield Activity: GBOSS, WAAS, and ground sensors continuously monitor human observables and feed information to Tactical Switchboard at CO and BN HQs.

2b. PISR System Activity: The PISR System is connected to Tactical Switchboard through the SA Subsystem. GBOSS and WAAS are also connected to the PISR System thorough SA. Biometrics processing on Tactical Switchboard has the ability to identify specific individuals. Other non-real time information feeds including MarineLink and GCCS are connected through the PISR IB Subsystem and are exchanging information across these connections. HUMINT and SIGINT reports are entered in MarineLink by intelligence personnel. The PISR System is looking for events that match the COI created in Activity 1b.

3a. Battlefield Activity: Several HVI location tips are received that originate from Blue-issued cell phones.

3b. PISR System Activity: Event detected: the PISR System determines that the activity arising in 3a satisfies the analyst's COI.

4a. Battlefield Activity: PISR System services running on SIGINT feeds connected to Tactical Switchboard narrow the HVI's likely location. BN S-2 analysts continue to fuse and synthesize local populace tips, HET reports, and Significant Activity reports.

4b. PISR System Activity: The analyst who created the COI is immediately alerted with a flashing message on his computer monitor when his COI is satisfied. The COI specification includes a request that BN HQ and below be alerted when the COI is satisfied. The PISR System prioritizes the alert so that communication resources are available to send the alert to BN HQ and subordinate elements. Another analyst-specified COI has an associated action requesting that a summary report of the sensor information be included with the alert. The PISR System packages a summary of the sensor information with the alert including pictures of the HVI taken within the hour and previous pictures of the HVI that are stored in the PISR IB. Actions associated with several other active COIs requested that their authors be sent any alerts dealing with this HVI. These users receive the alert and begin doing additional research to help positively identify and track the individual. The PISR System automatically updates case files that include this individual to add current information from the sensors and analytics. SOPs for this BN require alerting a call list of key personnel when an HVI is identified. The PISR System retrieves this list from PISR IB and uses the Dissemination Subsystem to phone all the people on the list. If a person fails to answer the phone, the PISR System automatically sends a text message and an email to that person. The PISR System continuously monitors communications to know when all persons on the list have been notified and they have acknowledged the alert.

5a. Battlefield Activity: While executing its patrol mission, 1st Squad (Rein) with HET is pushed alert information of the presence of a HVI in Green Province and provided ongoing situational awareness of the unfolding target set.

5b. PISR System Activity: New COIs are created looking for Marine assets in the current and projected path of the HVI that may be able to collect additional intelligence or who could assist in apprehending the HVI. 1st Squad is the only unit close enough to be of assistance so they are pushed the alert through the digital tactical radio system. The PISR System monitors the communication system to determine if 1st Squad responds to the text message. 1st Squad does respond.

6a. Battlefield Activity: Terrestrial and airborne SIGINT and EO sensors dynamically orient to collect on the HVI and isolate his location. PISR System services continue to monitor for developments and push updates to the Platoon and higher as required. Onboard and BN-level analytic processes continue.

6b. PISR System Activity: Additional COIs are being created to look for additional enemy activity that may be associated with the HVI. PISR System users monitor HUMINT and other intelligence sources looking for additional evidence on the identity of the individual being tracked and for techniques to maintain and improve the track of the HVI.

7a. Battlefield Activity: 1st Squad is pushed the HVI location, a situation update, and updated Commander's Intent. The Platoon conducts a hasty mission plan and commences the mission to apprehend the HVI.

7b. PISR System Activity: 1st Squad EOD sends out a message that a mission is underway to apprehend the HVI. All PISR System users who have a case file mentioning this individual or who have COIs associated with this individual are notified by the PISR System that a mission is underway to apprehend the individual.

8a. Battlefield Activity: Positive target ID and location.

8b. PISR System Activity: Human Intelligence reports flow into the COC indicating that the individual being tracked is positively identified as an HVI and that his location is known. These reports are detected by the PISR System and information from the PISR System helps support the mission that is underway. Analysts and other users who are receiving PISR System alerts use the PISR System and other COC resources to follow mission progress.

9a. Battlefield Activity: The HVI is apprehended.

9b. PISR System Activity: The PISR System is used with other tools to understand that the HVI has been identified, located, and that little risk will be incurred in an immediate seizure of the HVI. The mission is executed and the HVI is forcibly detained. Analysts and other users connected to the PISR System update their case files with this new information. The PISR System monitors several other data sources including MarineLink and pulls information about this successful mission into PISR IB for future use.

10a. Battlefield Activity: Mission continues. BN/CO S-2 continues to monitor situation and developments. GBOSS, WAAS, and ground sensors continue to monitor events.

10b. PISR System Activity: The PISR System continuously looks for events that satisfy COIs and continues to support users through information flow to and from the PISR System User Interface Environment Subsystem.

2 User Interface Environment Subsystem

2.1 Introduction

The User Interface (UI) Environment Subsystem provides a Web-based interface that allows the warfighter to manage an array of sensor resources to provide timely intelligence information. The ability to achieve a global or synoptic view of the battlefield is now becoming possible with today's electronics and smart sensors. Managing these resources to achieve this is the goal of the User Interface. The user interface must display what smart sensors "see" as well as allowing the user to manage these sensor resources. The User Interface Environment Subsystem supports user interactions to identify situational "triggers" that can alert and cue the user when critical events or conditions are detected in the battlespace. The User Interface Environment Subsystem assists the users in collaboratively creating, updating, and managing case files that store information of interest.

2.1.1 User roles

The primary user for the PISR System is the Battalion Intelligence Staff Officer (S-2) and his subordinates. These users describe the configuration of the system and operate the system by issuing PISR Information Requests (PISR IRs) and building up case files used to track sensor results, review historical data, and document results and actions.

Commanders and high-level intelligence process managers are users of the system as well. They establish policies that the PISR system will adhere to, authorize allocation of sensor resources, establish high level PISR IRs, and alert lower echelons of events and other information.

Users are able to view sensor data, images, and hypotheses on maps as the events occur. Video and still images from sensors are rendered on the maps registered to the viewed location as appropriate. The user is able to playback sensor events and set filters to filter out information as needed.

All forces can register to receive alerts for detected sensor events. Alerts can be configured so that some users can review the alert to verify or confirm it before it gets sent to others.

2.1.2 User's Task Model

The capabilities of modern electronic sensors and networking provide an opportunity for new levels of awareness on the battlefield. The tasks required to make these capabilities successful are:

- Planning – Plan for sensor deployment.
- Verify – Verify that the sensors are deployed at the correct locations and be aware of their health and status.
- Information Requests – Inform the sensor system what to look for.
- Alert – Configure alert messages to be sent to interested parties when events occur.
- Evaluate – Examine the sensor output and determine meaning of the data (interpretation) in a timely manner.
- Adjust – Adjust the sensor's focus as the situation on the battlefield changes. Mobile sensors (e.g., UAVs) can be re-routed and sensors can be cued.

2.1.2.1 Planning

Planning for sensor deployment falls under the category of creating and executing a collection plan. The techniques for planning deployment of sensors must keep up with the dynamic capabilities of current and future sensors. To support rapidly changing situations, the system indicates possible optimal locations for sensor placement and optimal mobile sensor routes. Sensor area coverage arcs and mobile sensor routes need to be displayed with the terrain effects and sensor limitations taken into account. The interface allows the sensor planner to visualize the results of his sensor plan.

2.1.2.2 Verify

Sensors, like all electro-mechanical devices, can fail. The user needs to verify that the sensors are operating correctly and that the system is receiving correct data from them. When a sensor fails or if a given sensor resource that was planned for is not available the system will inform the user.

2.1.2.3 Information Requests

Smart sensors need to be configured for what to look for. This reduces false alarms and avoids overwhelming the user with all the low level sensor data. The user can specify what is of interest to him in a high level way and let the software worry about what the high level representation means in terms of input and output from the sensors.

2.1.2.4 Alert

The user can specify what people to alert, under what conditions they are to be alerted, and the methods used to alert them.

2.1.2.5 Evaluate

Sensors return video, image stills, and various other data feeds. When the sensor and associated analytics detect events of interest, the user is directed to these locations and presented with enough information to understand quickly what has occurred. The user has access to historical information about what has occurred at this location in the past as well as any other relevant data. This and other information can be collected to allow the user to maintain a history of what has occurred and what the occurrence of this event means for future events.

2.1.2.6 Adjust

Sensor detection events can lead to sensor cueing to verify what has been detected or to gather more focused information. More sensors can be tasked to the area of interest. The user is able to communicate with others to adjust the sensor collection plan as the current situation changes and to examine or query the sensor data to know how best to re-task sensors in light of new information.

2.1.3 How the User Accomplishes these Tasks

To accomplish the planning, verify, information request, alert, evaluate, and adjust tasks the user needs a system that is linked to smart sensors, sensor analytic software, and both current and historical data. In the following subsections, each task is examined in more detail to address how the system will help the user with those tasks.

2.1.3.1 The Planning Task

The planning task is accomplished by the Collection Management Assistant that displays the current sensor assets available, their current locations, and any current missions assigned to them. The user can request the system to indicate on a map where it thinks the best sensor locations should be and plan sensor movement to different locations, point them in different directions (if required), and examine how much coverage they would have at those locations. Sensor modes and sensitivities can be taken into account and set to different configurations at different times.

2.1.3.2 Verify Task

The verify task is accomplished by the Collection Management Assistant and Observation Editor. The Collection Management Assistant can display current sensor locations and orientations as well as the field of views and icons that reflect the state of the sensor. Mobile sensors are displayed at their current location with indicators for their current course, speed, state, and track. A sensor icon can be selected and software specific to that sensor can be started to get more detailed information about that sensor. The Observation Editor can be used to obtain direct sensor observation video and to rewind and play it back.

2.1.3.3 Information Request Task

To aid in their effectiveness, smart sensors need direction on what things they should look for. The user employs the PISR IR Editor to specify who (who do we look for), what (what are they doing), when (when are they doing it), and where (where are they doing it). The smart sensors can then prioritize this request with available resources and evaluate what is the best way to accomplish this PISR IR.

2.1.3.4 The Alert Task

Alert messages need to be sent out once the sensors detect an event. The Alert Notification Editor allows users to create alert lists of people to be informed and how the alerts are sent out. Primary communication mechanisms as well as secondary mechanisms can be specified.

2.1.3.5 The Evaluate Task

The evaluate task is accomplished by using different editors. The Observation Editor shows the user that sensors have generated an alert. It indicates where the alert has occurred as well as the relevant information from the sensor related to the alert. It shows on the map the location of the alert and any live feeds available from the sensor tiled to the user's map. This sensor feed can be replayed as required to evaluate what has occurred. The user can bring up the Case File Editor to tie in this sensor alert area and alert type with other historical events that have occurred in this area. The Case File Editor also allows the user to query for other information that might be relevant to this alert. Once the user has a good handle on what is going on, he can use the observation editor to pass any additional information to the system about the alert.

2.1.3.6 The Adjust Task

The adjust task is accomplished by the Collection Management Assistant, the Observation Editor, and the Case File Editor. The Collection Management Assistant is used to change sensor allocations. The Case File Editor can record the adjustment and reasons behind it if required. The Observation Editor can verify that the adjustment was carried out and then view the results.

2.1.4 User Interface Editors

As indicated above, the PISR System user interface is built from a number of editors. These editors allow the user to accomplish the above tasks. The following subsections describe each editor.

2.1.4.1 PISR IR Editor

The PISR IR Editor lets the user specify situational triggers (battlespace conditions of interest) and who to notify once the situation occurs[21]. The editor offers the following features:

- The user defines the basic attributes of who, what, when, and where to describe a component of a PISR IR. Multiple components can be logically tied together to form more complex IRs.
- The system has built-in PISR IR templates and the user can create his own templates for use by himself or other analysts. Templates are used to facilitate the creation of PISR IRs. Templates can incorporate known enemy TTPs.
- Threshold values can be set on the PISR IRs. The sensitivity of triggering and resulting false alarm rates can be set.
- A summary of currently active PISR IRs are available.
- The user can specify additional actions to validate a PISR IR once an alert is triggered.
- If sensor resources are not available or tasked for other purposes, the editor informs the user and the user can try to resolve those issues using the Collection Management Assistant. If mobile sensors need to be deployed to satisfy a request, the system informs the user and the Collection Management Assistant can help facilitate their deployment.
- The user can bring up various situational awareness items on the map display to aid in defining PISR IR geographic location definitions.

2.1.4.2 Observation Editor

The Observation Editor allows the user to become part of the sensor network by reporting human observations to the system. The Observation Editor displays sensor alerts and sensor video. It has the following features:

[21] To be precise, it is really "once the situation is *perceived* to have occurred" as determined by processing of sensor data against various criteria.

- Sensor alert information is displayed on a map at the location of the event. Detailed information about the alert is available on user selection.
- The user can get a list of what sensors are available for live video and he can request that the video be tiled onto a map. The sensor video can be played back, rewound, and fast-forwarded. If supported by the system, the sensor video can be adjusted to provide higher resolution video in areas of importance.
- Sensors without overhead video capability but having video or still images have an icon on the map at their location showing indication of availability of imagery; selecting these icons allows their video or still images to be displayed.
- The user can configure how data from different sensor types are displayed.

2.1.4.3 Alert Notification Editor

The Alert Notification Editor allows the user to set-up notification lists and hierarchies to manage the notification process for sensor events. It allows the user to specify primary as well as backup contact information to insure that the alert messages will get to the required destinations in a timely manner. The Alert Notification Editor has the following features:

- Alert contacts can be changed based upon the error threshold of an alert. Alerts with a greater confidence factor can be specified to be sent to different contact lists than the same alert with a lower confidence factor.
- Different contact methods can be specified for each contact. Each contact method can be given a priority, a level of alert before contact is done, and alert types to be sent to this contact method.
- Backup contacts lists or individuals can also be provided with the alert. These are triggered when all of the primary contacts fail.
- The system contacts the user based upon contact priority, alert level, and alert type. It verifies that the alert was actually received by the contact and if not it alerts using the next contact method.
- Besides user-defined alert lists, the system has Warning, Threat, and Watch lists that are maintained as per the system configuration.
- Case files can have alerts assigned to them so that when they are modified alerts can be sent out.

2.1.4.4 Collection Management Assistant

The Collection Management Assistant is a unified way to manage different types of sensors and their software for collection planning and sensor tasking. It can verify the status and functionality of its sensors. The Collection Management Assistant has the following features:

- Provides a list of available sensors and the status of those sensors.
- Allows for sensors to be added and removed from the system.
- A sensor can be selected and its location and coverage arcs are displayed on the map along with any known missions assigned and the sensor owner. Mobile sensors display their course, speed, track, and time remaining on station.
- Different locations for sensor positions can be explored. The map displays the sensor coverage arcs based upon the new location taking terrain into account. The Collection Management Assistant can be requested to provide “heat maps” that display optimal sensor placement and route information for the given terrain and sensor type given a set of user-definable assumptions.
- Sensor modes and cueing changes can be specified if the user is the owner or, if not, sensor requests can be made.
- The sensor owner can specify priorities for the sensor. The system will use these priorities when deciding to allocate resources to satisfy PISR IRs.

2.1.4.5 Case File Editor

The Case File Editor aids in the creation of PISR IRs and in understanding the context and meaning of alerts. This editor allows the user to query and retrieve database information and use that information to help understand and document events on the ground. With this information the user can narrow down the “who”, “what”, “where”, and “when” data needed for defining PISR IRs. The user can also tie the data gathered from alerts with historical

information. This can lead to new insights as to what is happening on the battlefield. The Case File Editor also allows assumptions to be tracked and to establish a data trail for documentation of events. It has the following features:

- Creation of case files and attaching data items and events to these case files. The editor supports multiple users having simultaneous access to the same case file. Different file types can be linked to a case file.
- Users can browse the different case files and receive alerts when case files contents are updated.
- A chain of evidence display allows for identification of what assumptions have been made, who made the assumptions, and why.
- The reliability of data in the case file can be shown as well as the reliability of data source(s).
- Queries into the database can be entered and their results saved in the case file along with user annotations and notes. Filters can be applied to these queries.
- Reports can be generated from the case file. Report templates are provided to ease this task. The user can export reports to Word documents and e-mail.

2.1.4.6 PISR System Configuration Editor

The PISR System Configuration Editor lets the system administrator configure the PISR system. He can specify what sensor types are supported as well as defining what software the sensors will use. He can specify and customize all the PISR PLA subsystem processes and start and shutdown these processes. Policies of the different subsystems can be modified from their defaults. Any customization of the PISR system is performed using this interface.

2.1.5 Portability across devices and platforms

The PISR system must support different hardware platforms the user will employ to access the system. It must support many different sensors that have different capabilities, from TRSS ground sensors to sensors mounted on UAVs. It is expected that integration of advanced hardware and software will evolve capabilities of the PISR System. The "thin client" web browser architecture of the User Interface Environment allows any platform to access the PISR user interface. The user interface services approach allows services to be replaced and third party software to "plug-into" the user interface.

The PISR System must support many different sensor configurations and the many different visualization capabilities that these sensors will provide. The Observation Editor integrates the various visualizations into a common view. Sensor vendors specify the display capabilities of their sensor types and the Collection Management Assistant allows the user to select from these different display capabilities to view the sensor output.

2.1.5.1 Adding a New Sensor Type to the User Interface

The System Configuration Editor allows the user to add a new sensor package to the PISR system. The editor uses the following information to configure its displays:

- Metadata configuration. The Metadata that goes along with the sensor data needs to be specified so the user interface can present the data.
- Display Script. The different display options and how each option is to be displayed are written in a display script. The user interface builds display options and display methods based upon the data in the script. The Observation Editor then presents the sensor data to the user using those display methods. The Display Script language is extensible, providing the capability to add new functions as sensor and display technology evolve. A Display Script is created for each new sensor type. Refer to Figure 8.

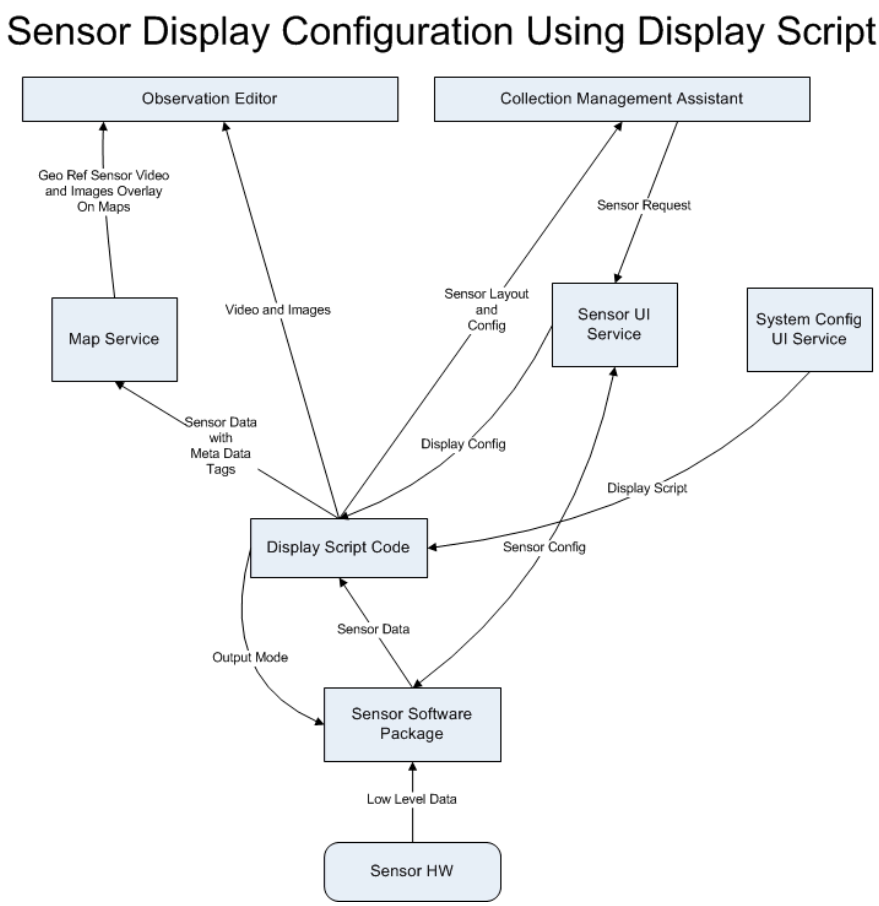

Figure 8. Sensor display configuration using a display script

The Display Script describes to the UI subsystem what the options are in displaying the sensor information and how to present each option. In the above figure, the Display Script Code processes the sensor image data based upon the user's selected display options. The display script code routes the image data to the Map Server if it has the capability to be presented as a map overlay, or it can direct the sensor images directly to the Observation Editor. If any transformation of the image data needs to be done, the Display Script code can pass it to the graphical processing unit (GPU) to perform the transformation. The Collection Management Assistant configures the sensor display based upon information returned from the Display Script code. Given the display mode passed in by the Collection Management Assistant, the Display Script code configures the sensor software package to provide that information.

2.1.6 Key Quality Attributes

11 The key QA for the User Interface Environment Subsystem is QA 51 (see Networking for PISR

11.1 Introduction

This chapter describes a set of requirements for integrating and managing the PISR network. It is the latest chapter in PLA document, based on results of most recent joint studies with Tactical Network Topology (TNT) team during the Fall quarter of 2010. Correspondingly, the current version is limited to most well understood battalion and below networking architecture requirements as well as fundamentals of 8th Layer based network management technique to be designed in accordance with Management Control Layer architecture.

11.2 Battalion and below

11.2.1 Probable ways to deliver bits at this level

At the battalion level and below, robust, ubiquitous, ad hoc mobile mesh networking clusters constitute the core for PISR bits delivery. Within the clusters, operators, unattended sensors, aerial and ground manned/unmanned surveillance nodes (towers, UAVs, UGVs, surveillance aircraft, ground vehicles, ground stations, etc) maintain self-forming networks by controlling their location on-the-move as well as the application load, subject to current terrain and node availability constraints , and COI based information delivery requirements.

Within the cluster (1-3 mile radius footprint) most of the layer 1/2 wireless links are the Line-of-Sight (LOS) types. We define cluster as small scale **squad level** network of operators, vehicles, unmanned nodes, and unattended sensors. However, the mesh character of the node-to-node connectivity allows to overcome most of the LOS obstacles by extending the peer-to-peer mesh around terrain obstacles, or alternating the links through the high elevation (towers in the area) or aerial relay nodes The result is is highly dynamic short-haul architecture, which employs light portable radios, hand-held PISR devices, and wearable relays.

Additionally, within the cluster, several single short-haul obstacle penetration or/and n-LOS links could be employed to augment the self-forming end-to-end mesh by through through-the-wall or n-LOS of capability.

The mesh enabled, sensor-unmanned systems-USMC operator PISR clusters could be interconnected by:

- Broadband wireless point-to-point links via the ground (towers), aerial (UAVs, tactical blimps, or air balloons), and sometimes limited orbital (Ku-band GEOS) nodes. This is a small scale solution with 3-4 PISR clusters, more suitable for the force protection type scenarios, in which the area of surveillance is fixed and doesn't change for several days or even weeks;
- Broadband wireless self-forming mesh links among the cluster gateway nodes via the ground (towers, reconnaissance vehicles, UGVs), aerial (UAVs, tactical blimps, or air balloons), and emerging orbital nodes. This is a more scaled solution for 6-12 PISR clusters, more suitable for highly dynamic ISR scenarios, in which the area of surveillance is changing hourly and might include surveillance areas distributed geographically beyond 200 mi area. Directional steerable antennas are highly desirable for maintaining inter-cluster broadband wireless mess architecture.

Figure 67 illustrates a small-scale PISR cluster example as assembled for the November 15-18, 2010 RPV-TNT Trial.

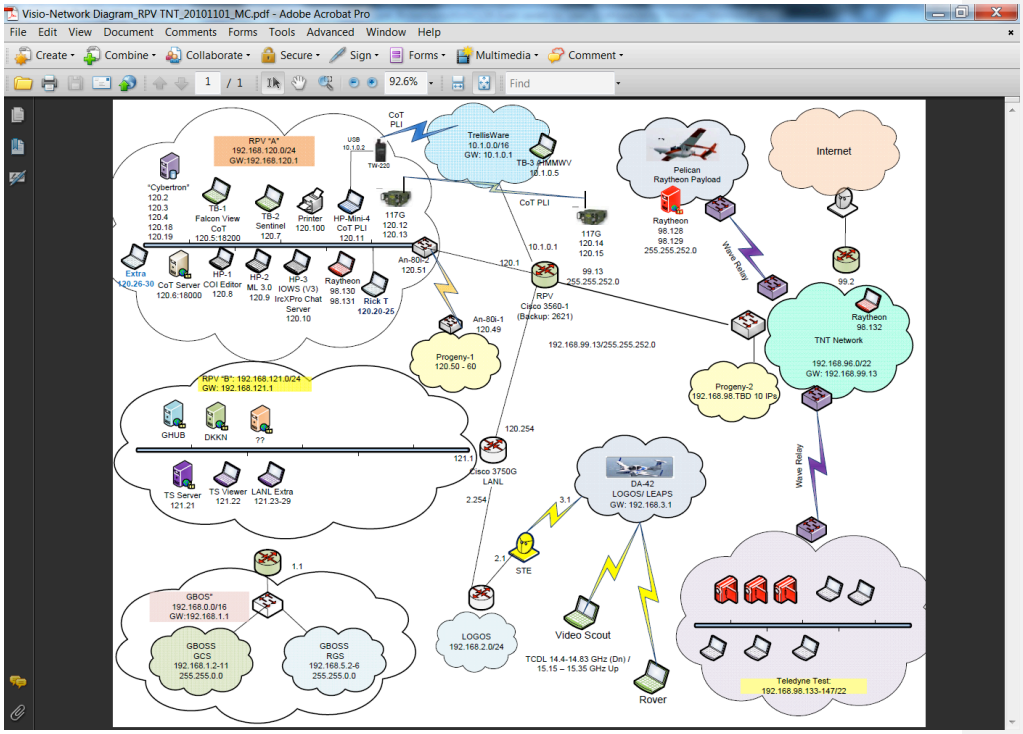

Figure 67. RPV-TNT tactical network diagram

- Support Mechanisms IP Space Routing Architecture
- Wireless Mesh Platforms
- Wireless platform Layer 2 bits-CoT message adapters/parsers: interoperability enablers

11.2.1.1 IP Space Routing Architecture

The routed network design for the PISR architecture was driven by two primary objectives: segmenting portions of the network to reduce the traffic load across bandwidth-constrained network links, and enabling multiple parallel data paths through the network.

During previous tests, it has been observed that with a moderate number of computing devices connected into a common Local Area Network (LAN), the level of "ambient" (background broadcast and multicast) traffic can exceed 1 Mbps. This ambient traffic is largely comprised of ARP requests, NetBIOS announcements, switch management protocols such as Spanning-Tree, and other similar discovery and management protocols. Although these protocols are necessary to support certain application functionality, an excess of traffic on bandwidth-constrained links drastically reduces the "useful" throughput of that link. Most wireless portions of the network, such as the Trellisware data-enabled radios, have a much lower maximum throughput than the wired portions of the network. Overhead traffic that is not noticed on a 100 Mbps or Gigabit wired network can significantly impact application traffic on a wireless link.

To prevent overloading constrained links, routing boundaries were implemented between the primary wired segments and any major and bandwidth-constrained wireless segments. As can be seen in Figure 1, the Track-A segment was separated from the Trellisware segment by a routed boundary. Likewise, Track-A and the TNT segments

were separated, since each network, though wired, contained many computing devices generating ambient load on the network.

Routing also allowed the use of multiple parallel pathways without introducing configuration pathologies. If multiple paths exist from one point on a LAN to another, a pathology called a "bridging loop" can occur, where packets will continue to traverse in a loop between the two points. Most modern switches prevent this behavior by selecting one path and disabling the others. However, it may be useful in some cases to allow certain traffic over one path versus another, or to share the load across multiple paths. Routers are able to implement these rules. For instance, there were multiple connections between the Trellisware segment and the Track-A segment; one supported all end-to-end application traffic, the other was used exclusively for management traffic (node position and performance monitoring).

11.2.1.2 Layer 2 bits-CoT Adapter : An example of Trellis Ware PLI-to-CoT parser

TrellisWare (TW) radio provides Position Location Information (PLI) in two data formats: KML (formerly Keyhole Markup Language) and JSON (JavaScript Object Notation, which is a lightweight data-interchange format) . Due to limited TW radio bandwidth, JavaScript Object Notation (JSON) data wrapping format was selected to be used, since to compare to KML format, the JSON generates more compact data messages. As shown in Figure 68, TW radios are forming mesh network of mobile units TW-1 – TW-n. Each unit provides PLI via mesh network (CheetahNet) by reporting its location to TW-master unit, specifically configure for that purposes. In TrellisWare terms, this unit is also known as command node or CMD.

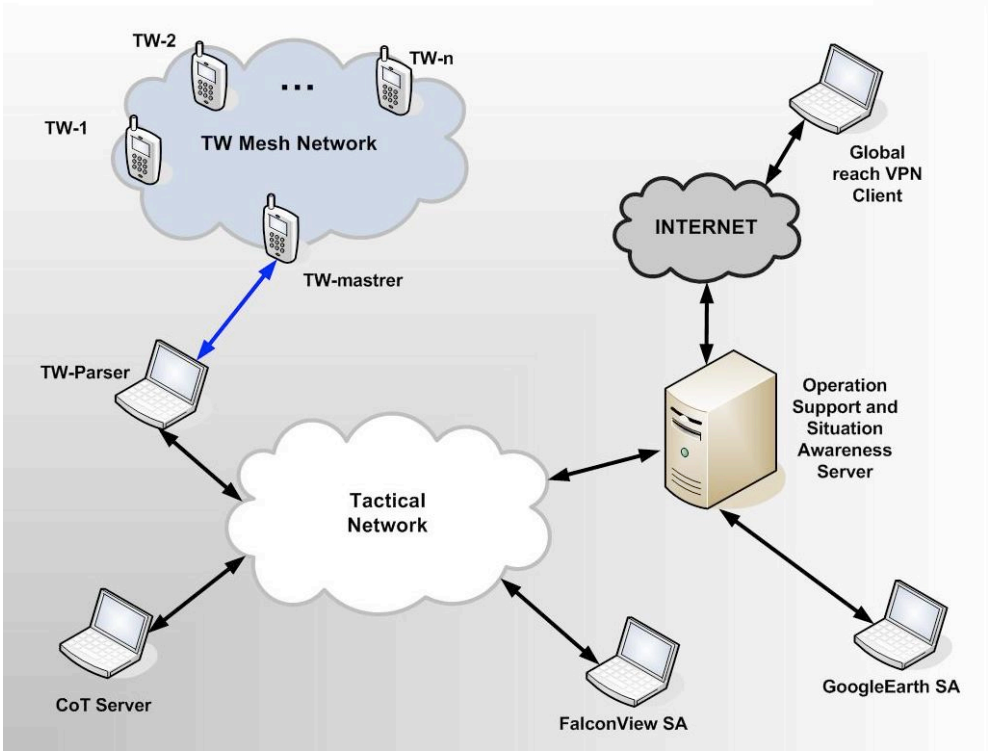

Figure 68. Network topology and TW mesh integration

TW-master radio via USB or Ethernet cable connected to computer running TW-Parser software. TW-Parser software was designed for current RPV experiment to provide the following:

- Polling TW-master data in JSON format
- Parse JSON data
- Generate CoT messages based on parsed data
- Send CoT messages to CoT Server to update FalconView and GoogleEarth SA

Operation Support and Situation Awareness Server generates GoogleEarth SA view based on CoT messages flow. The CENETIX SA Server located in NPS was playing this role in RPV experiment. Another important role of CENETIX SA Server is to provide global reach functionality to the remote VPN clients. Each PLI postings was time-stamped and stored in SA Server database for later analysis and replay. An example of database query of single TW unit tracking on GoogleEarth SA presented in Figure 69. Live tracking as it appears on GoogleEarth SA is presented in Figure 70.

Figure 69. Example of TW unit tracking on GoogleEarth situational awareness

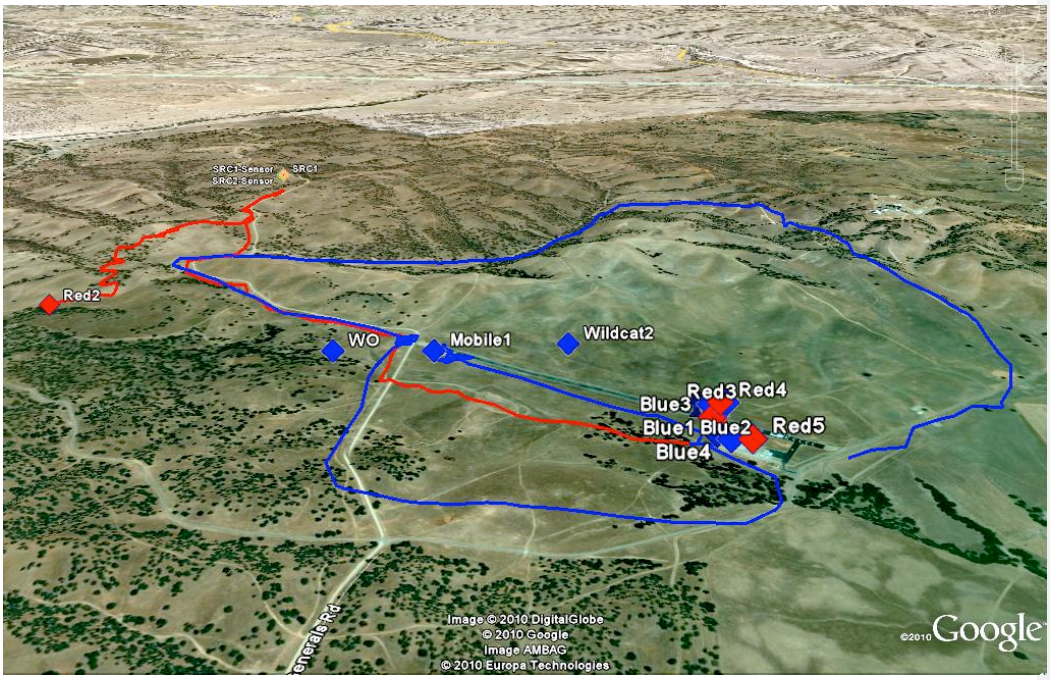

Figure 70. Example of live tracking on GoogleEarth situational awareness

The TW Parser GUI presented on Fig. 3. GUI allows user to assign IP address to the master node (CMD unit) in which PLI from all available via CheetahNet TW units will be collected until polled out by TW Parser within assigned polling interval. The CoT Server IP and its core configurable parameters are also available via TW Parser GUI.

TW Parser GUI provides JSON parsed data from each TW node currently registered with CheetahNet. Only nodes covered by CheetahNet mesh network and providing adequate security key might be successfully registered with CheetahNet. The PLI set of data consists of Latitude, Longitude, Heading, Speed, and Altitude. TW unit registered with network but failed on its GPS fix, will be represented by record with yellow background in GUI table grid as shown on Figure 71. Poor GPS reception or malfunctioning (disconnected) GPS antenna should be considered as the most likely reason for that. The CoT format allows to map the TW radio location and movement into the common operational picture GUI tracks (as shown in the GoogleEarth figures), while the CheetahNet data elements allow to track the health status of each radio node. The association of such two types of GUI, is an important requirement for integrating tactical radio nodes in the battalion level situational awareness environment.

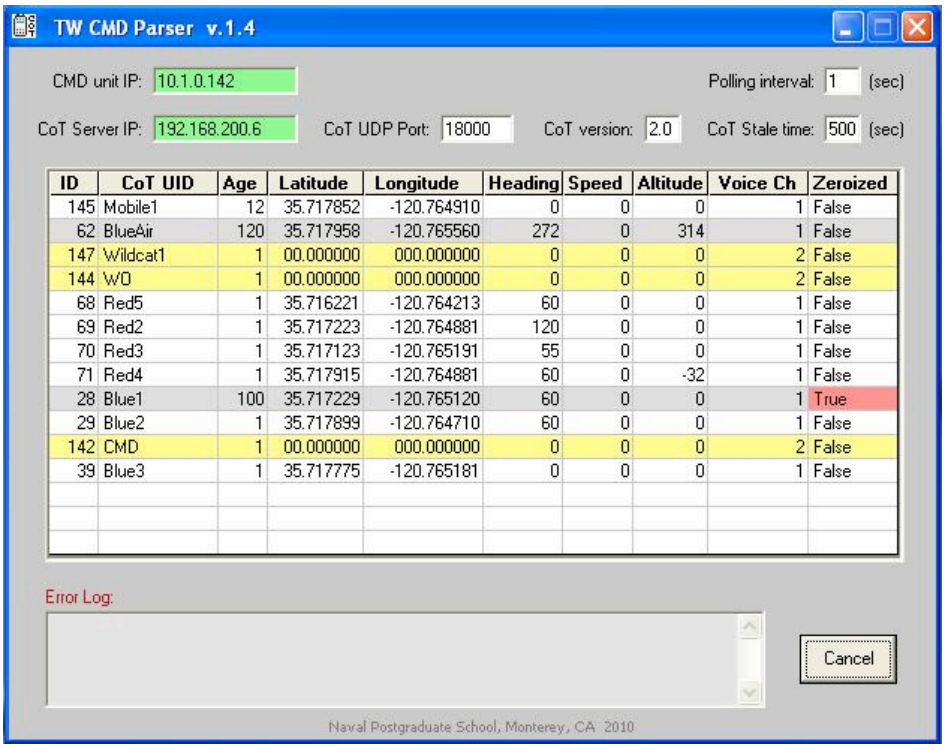

Figure 71. TrellisWare JSON-CoT parser GUI

If no data are received after 20 polling intervals, the TW node is considering as disconnected and will be marked with a gray color background. Some possible disconnection reasons are: out of mesh network coverage, battery failure, zeroized unit. Zeroized unit also marked with a red color background field. The Age field is a counter of polling intervals since the last successful update. If Age is more than 20, then Latitude and Longitude are representing the last known PLI before GPS lost. TW Parser generates CoT message in accordance with unit's status. As a result, the shape and color of TW unit icon is visually representing current unit's status on FalconView SA. The example set of unit icons representing current unit's status are shown in Figure 72.

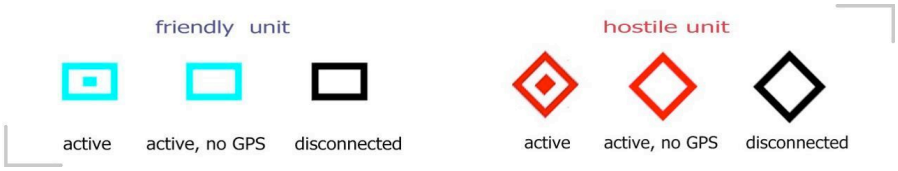

Figure 72. Example of icons representing unit status

The current version of TW Parser works with up to 15 TW units, but can be easily modified to manage more units if needed.

11.2.1.3 Standards for PISR Cluster Mesh

Based on the last 5 years of NPS-USSOCOM-DHS field experimentation with different mesh networking solutions for ISR, HVT (High Value Target tracking) and MIO (Maritime Interdiction Operation) missions, we recommend the following standards for PISR cluster mesh networking:

- PISR Self-forming mesh broadband wireless mesh: OFDM 802.11
- Mesh enabled software programmable radios
- Short-Haul obstacle penetration: UWB (Ultra-Wide Band), MIMO (Multiple Input-Multiple Output),
- Mesh Routing Standard: MANET (Mobile Ad Hoc Networking-DARPA)
- Mesh Routing with Feedback Control: CBMANET (Control-Based MANET-DARPA)

11.2.1.4 Standards for Inter-Cluster Links

Similarly, the extensive field experimentation studies of different inter-cluster links, conducted at NPS for the last 5 years show most promising performance of the following platforms:

- Point-to-Point fixed: OFDM 802.16
- Tactical Cellular (GSM, GPRS)
- Mesh mobile, with directional steerable antennas: OFDM 802.11
- Orbital fixed: Ku-Band GEOS
- Orbital routing: IRIS LEOS

11.2.1.5 What's off the shelf to support developers/integrators in rapidly reapplying this in the next system

- PISR Self-forming mesh broadband wireless mesh: OFDM 802.11: Persistent Systems Wave Relay, fixed and wearable systems, MANET standard
- Mesh enabled software programmable radios: Trellis Ware radios, Harris 117G
- Point-to-Point fixed: OFDM 802.16: Redline Corporation A 80i system

11.3 8th Layer

11.3.1 How we make this system controllable so that we can optimize the value of bits delivered

In accordance with 8th Layer concept, the PISR network could be made controllable through the coordinated work of PISR node monitors, which associate network status at Layer 1-3 with the health and services constraints at the higher levels of node functionality:

- SNMP events Monitor (OSI layers 1-3),
- SA constraints Monitor (MCL Registration Service),
- Service constraints Monitor (MCL Health Service Monitor,,).

In such an architecture the SNMP event-constraints monitor is simply a commonly used SNMP agent manager, relocated from the Network Management System suite at the NOC to the PISR node 8th layer suite. Unlike it, the monitors for SA constraints and SLA requirements negotiation do not have a common standard, and these need to be developed.

Given the fact that in the current PISR architecture, Management and Control Layer (MCL) subsystem by Coogar is responsible for monitoring configuration (SA), health, and policy constraints associated with PISR nodes, the 8th control of most valuable bits delivery could be accomplished through the integration of SNMP MIB Agents

with MCL monitors. This would allow to put under control such variables as application switching, node physical mobility initiation, receiver context and requirements modeling, sender dynamic information context and transmission requirements modeling, recipient context determination, SLA generation, SLA negotiation, QoS monitoring and SLA assurance, etc.

We envision that coordination of different monitoring processes within the 8th Layer would be driven by the network productivity SLA requirements. Each hyper-node would evaluate its own 8th Layer controllable variables. Each hyper-node would attempt to optimize its own sub-network by making changes in the application load, or by moving the node physically to a better postion (Node mobility control) as depicted in Figure 73. The "dualilty" of 8th layer adaptive management technique is that the SNMP-type performance monitor observes an instantaneous network behavior at Layer 2 and Layer 3 levels, however the SLA controls could only be applied via the MCL agent

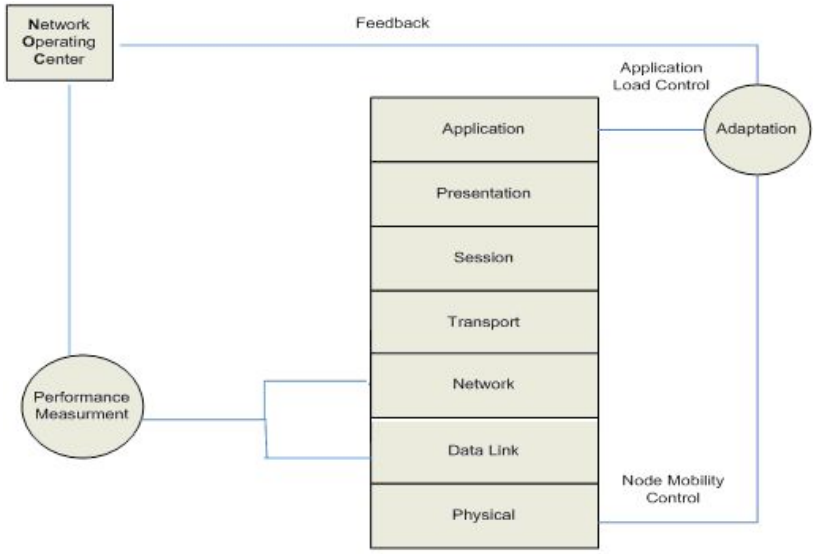

Figure 73. Intelligent adaptation required to maximize network productivity

platform at Layer 7 and Layer 1 respectively (Figure 73). Translation of of SNMP alerts into the load change controls (Layer 7 control), or/and node mobility control (Layer 1) should be done via the MCL Health Status Knowledge Reasoner (performance measures translation into the MCL knowledge base) and MCL Policy Knowledge Reasoner (Layer 7 and Layer 1 controls). Correspondingly, the following 8th Layer Adapters might be needed:

-SNMP (read)-----→MCL Health Status Knowledge Reasoner Adapter: Translation of SNMP based performance measures (RFC 1213 and related SNMP MIB standards);
-MCL Policy Knowledge Reasoner------→SNMP (write) Adapter: Translations of MCL Policy Knowledge Reasoner rules into the SNMP (write) applications load changes and mobility related node controls;
-SNMP MIB extensions might be needed to maintain translation to Health Knowledge Reasoner Alerts and provide for the lower level SNMP (write) controls.

11.3.2 The 8th Layer Memory

In addition to key monitoring processes, the 8th Layer protocol, which enables adaptive network management by the hyper-node itself, should also include a memory mechanism. Such memory mechanism would record and apply a small-scale knowledge base reflecting configuration, performance, security, and application management experiences of NOC crews. The MCL Health Status Knowledge Reasoner could be the main building block for memory component

11.3.3 The 8th Layer Solvers

If we were to define the 8th layer ontology, the most straightforward way would be to represent it through a concatenation of quantitative and context-based constraints reflecting the NEML, NML, SML, and SLA requirements, with SLA constraints defining the goal-seeking intelligence of the 8th Layer. Adapting different resources of physical, link, network, transport, and application layers of hyper-nodes functionality would require a multiple criteria solver, which would enable the hyper-node to perform feasibility analysis and then compromise on a large number of heterogeneous constraints.

Appendix A. A-Level Stakeholder Quality Attributes): "Users should be able to access the PISR System using only a computer browser (Microsoft Explorer preferred) without the need for large application software modules on the user's computer." To meet this objective the User Interface Environment Subsystem takes with a "thin client" approach. This means that most of UI logic is on the web server side and only what is needed to get the data from the user and pass data along to the user is contained in the "thin client". The bulk of the User Interface Environment Subsystem therefore resides in services on the web server side.

Another key goal of the UI is that no user's manual be required. The user interface is a web page that appears like most other web pages that a user can navigate intuitively without needing a manual. Sensor image data is displayed or tiled on a map and the context of the information is incorporated into the display of the data. Tasks can be performed with a minimum number of key strokes and graphical representations make the information presented easy to understand.

Current technologies such as Ajax for dynamic web page design can provide a fast and fluid interface. Just the part of the web page that needs updating is regenerated, not the complete page.

2.2 Top Level Architecture

The User Interface Environment provides a web browser-based interface that communicates with Web-based UI server processes to allow the user to accomplish the various tasks described earlier. As discussed above, the UI takes a "thin client" approach to this problem. Functionality is easily changed by just replacing the UI services as required. External tools have easy access to the parts of the UI that they are interested in, and it is easy to plug-in new tools.

The UI services enable separation of views (what the web browser shows) from the rest of the UI logic. The server-side logic interfaces with the rest of the PISR System. The UI Environment Subsystem is designed to support integration of third party tools by connecting to those services. For example, Figure 9 below shows Tactical Switchboard and the Semantic Web-based Interface for Marines (SWIM) are shown as 3rd party tools integrated into the PISR System. It is assumed that other 3rd party tools will tie into the PISR System UI as they are developed so that the benefits and features of these tools can be accessed seamlessly by the user. Here the Tactical Switchboard tile server is being used by the PISR Map Server to display real time video imagery on its maps. The Case File Management UI Service obtains data from the SWIM server and includes it into the case files that are displayed.

The following subsections describe the various services offered by the UI Environment Subsystem.

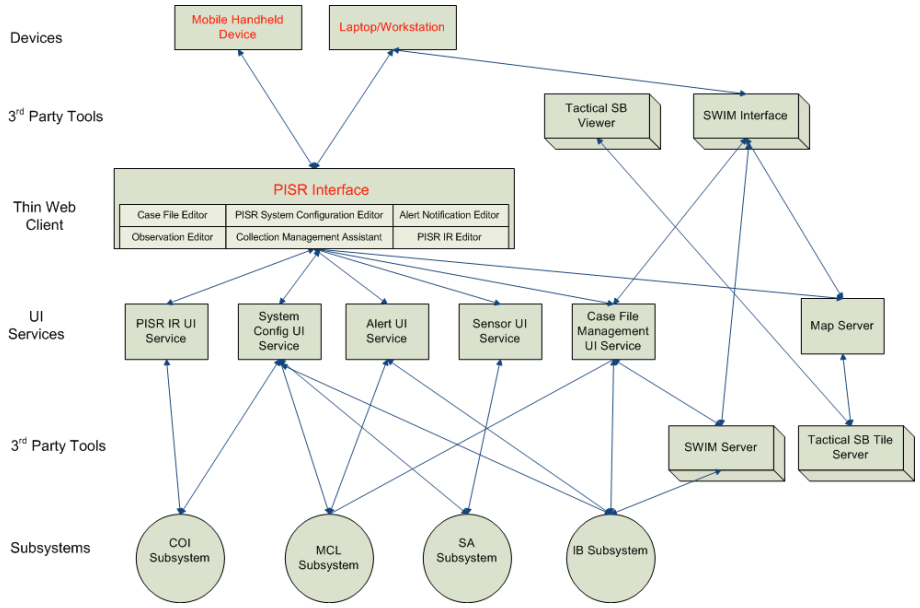

Figure 9. UI Environment Subsystem architecture

2.2.1 PISR IR UI Service

The PISR IR UI Service supports creation of PISR IRs defining the "who", "what", "where", and "when" that should be looked for by the sensors as well as whom to notify when the event is detected.

2.2.2 System Configuration UI Service

The System Configuration UI Service allows the user to bring on line and configure the different subsystems within the PISR System. It also allows integration of third party tools into the system and the swapping of both hardware and software components. Sensor software packages can be added or removed. This service is accessed by system administrators of the system. The service is also used to specify policy direction for each of the subsystems.

2.2.3 Alert UI Service

The Alert UI Service handles the alert configuration for the alerts output when the sensors detect something satisfying the conditions of interest (situational triggers). This service deals with who is to be notified for what alerts and what the notification and backup notification methods are.

2.2.4 Sensor Management UI Service

The Sensor Management UI Service provides access to the sensors. Individual sensors can be added or removed from the PISR System. Sensor-specific software will be called to check if sensors are operating properly. Software can review the data coming in from the sensors and can configure the sensors as needed. The software can generate sensor coverage and availability displays. The Collection Management Assistant interfaces to this service to manage collection operations.

2.2.5 Map Service

The Map Service provides map data to the user interface and manages the polygons and other user input that the user has provided on the map displays. The Map Service supports image map tiles and other map overlay data

coming in from external sources. This service also provides the standard Open Geospatial Consortium (OGC) map services.

2.2.6 Case File Management UI Service

The Case File Management UI Service provides a mechanism for the user to collect and view data related to PISR IR requests. Software can query for historical data for display and for linking with sensor events and other related data. Case files help the user define the initial PISR IR conditions and better understand the sensor alerts by placing them in the context of the events occurring on the battlefield.

Figure 10 depicts data flows across the UI Environment Subsystem and Case File Management Service.

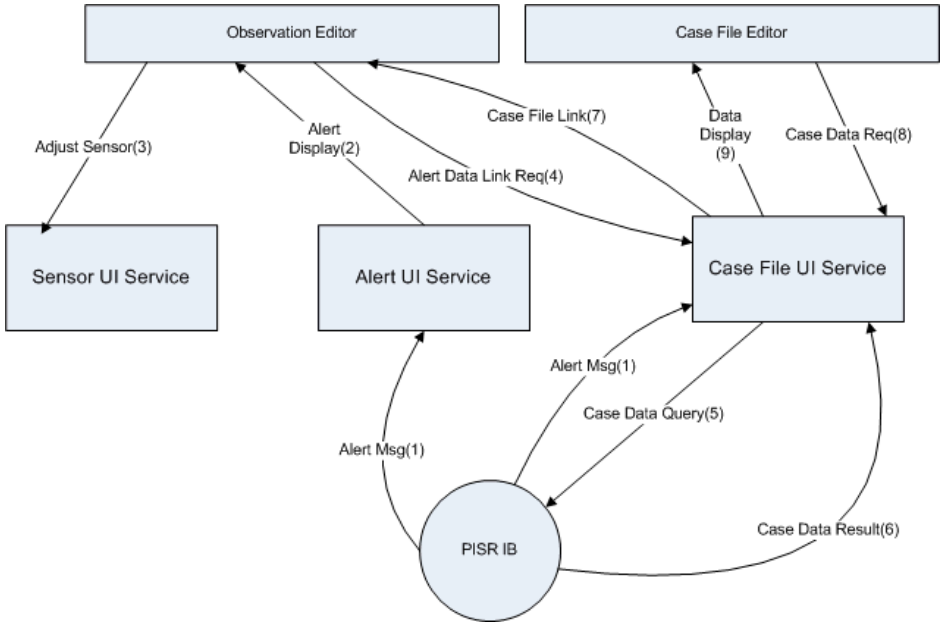

Figure 10. Alert and case file data flow through the UI Environment Subsystem

An alert message comes into both the Alert UI Service and the Case File Management UI Service. The Alert UI Service displays the alert message. The user can task the sensors to verify the alert. The user can manually request information about the alert or can use the Observation Editor to configure the system to provide that information automatically. The Case File Management UI Service requests information about the alert and its associated data from the PISR IB. The related case file data is retrieved and when the user opens the Case File Editor the data associated with the alert is displayed. The system also displays relevant related information on other activities in the area, including how often alerts have occurred in the past in this area.

2.2.7 Sensor Display

The combined view of multiple sensor sources on the battlefield is one of the most important tasks of the PISR System UI. The user needs to be able to focus in and drill down on important areas. Different views of what the sensors display are vital for understanding the complete picture. The User Interface Environment Subsystem utilizes sensor image data overlaid on top of maps and display scripts to provide different views and capabilities to the warfighter.

2.2.7.1 Sensor Image Overlay onto Maps

Display of sensor video onto a map overlay is a compute-intensive task that can use significant network bandwidth. The PISR system tests the bandwidth of the connection and adjusts the video playback quality and frame

rate accordingly. Transformation of the image can also be expensive. The PISR system utilizes the display hardware's GPU to perform the graphical transformations required. The PISR system passes along only the changed video bits to reduce bandwidth and processing requirements.

2.2.7.2 Display Script Configuration

Display processing of sensor images must be done as efficiently as possible. Different sensor types and configurations can provide many different options for what is displayed. In addition, as new sensors are added they need to be easily integrated. Display pipelines need to be configured for each of the sensor image data paths. These pipelines have common element blocks but must be customized for each sensor type. The display scripts accomplish this by allowing each sensor type to have a custom display pipeline. Filters and special processing steps can be defined and "compiled" into custom processing methods for each sensor type's display pipeline.

2.3 UI interfaces

2.3.1 Interfaces to Subsystems Internal to the PISR System Provided by the UI Environment Subsystem

Each of the UI services described above provides features to web browser-based thin clients or third party tools. The Map Service is based on standard OGC map server interface. Other services are implementable (for example) as Java services built on top of the Spring Framework. The "thin" web browser clients access these interfaces. Third party tools can also access these services. The following subsections identify the UI Environment Subsystem services and their associated operations.

2.3.1.1 PISR IB UI Service

- createActiveCOI – This creates a condition of interest (COI) request that gets passed to the COI Subsystem. The data passed contains "who, what, when, where" information that the smart sensors need to look for. Also passed is the error tolerance that the user wants to associate with this tasking.
- deleteActiveCOI – This deletes an existing COI request.
- listActiveCOIs – This lists the current active COIs.
- createTemplateCOI – This creates a COI request that is saved but not set active. This is a template that can be used later. COI templates owners can be the current user or the system (if created by the system administrator). Templates can be incomplete COIs. They can be missing instance data that would be required for an active COI.
- deleteTemplateCOI – This deletes an existing template.
- listTemplateCOI – This lists all the COI templates. This can be filtered by just the user templates or system ones.

2.3.1.2 PISR System Configuration UI Service

- addSensorPackage – This allows a new sensor type to be configured within the PISR System. It defines what software needs to be called for managing the sensor and tying it into the SA Subsystem.
- removeSensorPackage – This called to remove a sensor type from the PISR system.
- getSubSystemPolicy – This gets the identified subsystem's existing policy.
- setSubSystemPolicy – This sets the identified subsystem's policy.
- addDisplayScript – This adds a new display script to the UI Environment Subsystem. The display script allows for new information to be selected and displayed.

2.3.1.3 Alert UI Service

- createAlertList, deleteAlertList, listAlertLists– These handle alert lists which are a list of contacts that get notified when a sensor event is detected.
- createContact, deleteContact, addContact, listContacts– These manage contacts and add them to alert lists. The different methods of how this person should be contacted are specified. Also specified is what level of threshold of confidence must be reached before this contact is alerted.
- requestAlerts – The Alert Service uses Comet technology to implement long polling from the web browser to support alert pushes from the Alert Service. As web browser technology improves this can be converted to a "true push".
- getAlert – This fetches the alert data to display for an alert returned from the requestAlerts.

2.3.1.4 Sensor UI Service

- getSensorStatus – This returns the status of the specified sensor. Status includes sensor class, the state of the sensor, any mission assigned, who assigned the mission, and the owner.
- getSensorList – This returns the complete of sensors that the PISR system knows about.
- getSensorLoc – Returns the location of a given sensor.
- getMobileSensorLoc – Returns the location, heading, sensor direction, and speed of a mobile sensor.
- getSensorCoveragePolygon – Returns the coordinates of what this sensor can see given ideal conditions and flat terrain. This data can be used with a terrain analysis to get the true sensor polygon.
- getSensorAttributes – Returns the settable attributes of a given sensor.
- setSensorAttributes – Sets the attributes of a given sensor.
- getSensorDisplayCap - Gets different ways the sensor data can be displayed.
- setSensorDisplay – Sets how the sensor will output to the display.

2.3.1.5 Map Service

- getCapabilities – Returns the layers that the map server supports.
- getMap – Returns the given map layer.
- describeLayer – Returns the description of a layer.
- getFeatureInfo – Returns the description of a given feature.
- getTileSensors – Returns the sensor feeds that can tile onto a map overlay
- startTileSensor – This sets the sensor video to be tiled. It specifies the start time and the playback speed, resolution, and if loopback is to be done.
- stopTileSensor – This stops the sensor video tiling for this sensor.
- createRoute, deleteRoute, listRoutes –These routines handle the display of routes on maps.
- createPolygon, deletePolygon, listPolygons – These routines handle the display of polygons on maps.
- createPoint, deletePoint, listPoints– These routines handle the display of point locations on maps.
- createOverlay , deleteOverlay, listOverlays– These routines handle overlay manipulation.

2.3.1.6 Case File Management UI Service

- createCaseFile – Returns a case file identifier what allows grouping of data references.

- deleteCaseFile – Deletes a given case file.
- addItemCaseFile – Adds a data item to a case file.
- getCaseFile – Returns the information contained in a case file
- listCaseFiles – Lists the case files known by the system. Filters are passed in to filter the results.
- removeItemCaseFile – Removes a data item from a case file.
- addInterestedParty – Adds a user to be notified if case file data changes.
- removeInterestedParty – Removes an interested user.
- listInterestParties – Lists the users interested in this case file.
- getLocHistory – Returns a list of events that has happened within a given polygon. Filters are passed in to filter the results returned.
- searchName – Returns list of events associated with the given name. Filters are passed in to filter the results returned.
- searchEventTime – Returns list of events associated with the given time. Filters are passed in to filter the results returned.
- searchString – Returns list of events associated with the given string. Filters are passed in to filter the results returned.
- linkCaseFile – Links a case file to another case file, a PISR IR, an alert, or another external event.

2.3.2 Interfaces to Systems External to the PISR System

2.3.2.1 Interfaces Provided by the UI Environment Subsystem to Systems External to the PISR System

A goal of the PISR system is to allow tools external to PISR to interface into the PISR system. The PISR System UI needs to be able to integrate those tools' displays and allow users to make use of the features of those systems. External tools can have their own map displays, imagery, and data displays that need to be presented. External tool displays fall into two categories:

1. The external tool has a web based display. Bringing up the external tools display then is a matter of providing a link to it that can be created by the system configuration editor when that tools definition is defined. For example, the Case File Editor can have a link to the SWIM case file management tool and the user could just click on that link to open up the SWIM tool.
2. The external tool does not have a web based display. The external tool can utilize the display scripts and access the back end of the UI services to get displays rendered in the PISR system. The display scripts will provide the presentation layer for the third party tool. Data for the displays can come from a new sensor package that was added or directly from the PISR IB Subsystem through its external adaptors.

2.3.2.2 External Interfaces Used by the UI Environment Subsystem

The UI Environment Subsystem can be classified by the thin client interface used within the web browsers and the Java servers that provide the bulk of the UI Environment Subsystem. The thin client interface utilizes the Javascript interfaces that are provided within a web browser. In the PISR PLA reference implementation, frameworks like Hibernate and Spring can be used. The reference interface into the Java services can utilize the Spring framework and other Java enterprise services to accomplish their tasks.

This page intentionally left blank.

3 Situational Awareness Subsystem

3.1 Introduction

This section describes the Situational Awareness (SA) Subsystem portion of the PISR Product Line Architecture. A goal of the SA Subsystem is to accelerate the rate at which product vendors contribute components that solve USMC SA problems. The type of products of interest for this PISR PLA subsystem are the analytics, sensors, and sensor integrators–including integrators for human generated intelligence–that perform the myriad of SA functions required to enable Persistent ISR. Ultimately, the USMC instantiates the SA Subsystem of the PLA with selected pre-qualified components appropriate to current missions and integrates these into effective PISR SA Subsystems. Part of that work is empirical and part of that work is the formulation of an effective integration platform. The empirical work requires a survey, analysis, and classification of sensor capabilities, including a description of their operating envelopes, controls, and outputs. The goal of the empirical work is to develop categories of sensor types represented as suitably abstract generic sensor components. Specific sensors described as instances of these generic sensor categories can then be efficiently integrated into an effective PISR System. This is the basic approach used throughout all PLA developments.

In addition to categorizing sensors, the SA Subsystem PLA also needs to abstractly represent the situation interpretation capabilities of analytics. The first analytic layer converts low-level sensor outputs into higher-level interpretations or hypotheses. Typically these base hypotheses describe entities and features observed in the sensor data. For example, one analytic might identify a human form in a video while others might determine the posture, movements, and purposeful behaviors. Still others might identify the size, gender, and hair color, and these might feed into others that hypothesize the identity of the person. In addition to analytics that convert sensor data into hypotheses about people, others generate hypotheses about vehicles, people-vehicle combinations, buildings, facilities, organizations, social networks, and so on. Rather than consume raw sensor input, some higher-level analytics build upon the hypotheses of other lower-level analytics. This is a very large information processing space. The goal of the SA Subsystem of the PISR PLA is to enable the USMC to employ the best components in each such category for the mission at hand. The goal throughout is to bring better SA capabilities to the warfighter, at the lowest possible cost, with the least delay. The overall architecture of the SA Subsystem accomplishes this goal by specifying the interoperation of generic component types and easily incorporates specific sources of hypotheses (sensors and analytics) associated with those generic component types.

The extent of the SA problem is broad, including fielded and developmental sensors and analytics, and everything from collection planning to fusion and focus of attention. To assure reasonable progress, the PLA incorporates best of breed proven capabilities while providing an architecture designed for incremental extensibility and evolution. The initial instantiation of the PLA must therefore focus on these available capabilities: (1) current sensors and intelligence processing software used by Marines; (2) government off-the-shelf capabilities for SA being produced by ONR and other relevant DoD programs such as the Navy's Comprehensive Maritime Awareness; and (3) established and proven paradigms for SA and data fusion, especially distributed blackboards.

All SA systems are concerned with fusing multiple sources of information to build a credible model, description or interpretation of entities, events, and other aspects of the environment. The terms *model*, *description*, and *interpretation* are roughly synonymous. Any operator in the battlespace needs to understand the environment, the players, their capabilities, and their intentions. Because such understanding always rests on perceptions and inferences, the knowledge and interpretation are uncertain. SA systems fuse information to develop the most credible interpretations of their observations, using the basic method of science to generate hypotheses consistent with the observations and testing those hypotheses against alternatives. A valid situational model explains the observations received and predicts future observables.

SA combines bottom-up and top-down activities. When humans react to stimuli, such as an unexpected sound or movement, they focus attention and appropriate resources in the vicinity of the stimulus to collect more information, feed hypothesis-generating analytics, and develop a mental model of what's happening. When these models generate specific expectations, people choose to orient their sensors toward places and events of a particular predicted sort. This top-down expectation-driven approach dominates the collection efforts of intelligent systems such as human beings. One special case of this top-down approach arises when intelligent systems attend to high-value potential events, even if they are not explicitly predicted. Because intelligent systems understand that they must react quickly to particular

threats or high-value opportunities, they orient their collection, sensing, and analysis so they will not miss these events, even at the risk of ignoring other potential stimuli. The PISR PLA presumes that resources are inadequate to collect and fully process all relevant data, so the PISR System focuses on effective processing of high-value information. Information can be pre-identified as high-value from an analysis of potential enemy actions, as through an Intelligence Preparation of the Battlefield (IPB) process, from the analysis of plan dependencies such as PIRs, or from other sources of conditions of interest (COIs). All of these methods produce descriptions of high-value events that the PISR SA System should detect with acceptably high confidence. To do that, it must combine bottom-up and top-down methods, orienting collection and analytics towards relevant and significant features, and detecting and responding to relevant stimuli by cross-cueing appropriate follow-up methods.

In short, the SA Subsystem portion of the PISR PLA must specify these things:

1. Categories of sensors abstracted into generic sensor component types
2. Ways to adapt specific sensor products to an appropriate component type
3. Categories of analytics abstracted into generic component types
4. Ways to adapt specific analytic products to an appropriate component type
5. A language of hypothesis types, suitable for representing models of dynamic situation elements including entities, attributes, behaviors, and states
6. Interfaces to an information repository, a *blackboard,* for recording, updating, and publicizing evolving hypotheses
7. Frameworks for implementing processes that employ components to generate or improve hypotheses
8. Control mechanisms for other subsystems to prioritize the generation of hypotheses in order to produce the highest value, while respecting the constrained resources of human attention, sensors, communications bandwidth, processing power, and storage

Ultimately, a mature SA System PLA will incorporate empirically validated, sound engineering answers for all of these eight elements. In the early stages of this project, the SA PLA can provide only initial answers for most of these elements.

3.2 Situational Awareness Subsystem Architecture

Several of the stakeholder-specified A-priority Quality Attributes listed in Appendix A relate significantly to the SA Subsystem. The QA with the fifth most votes (#13) requires that PISR System users can define Conditions of Interest (situationally relevant information requirements) about various entities in the battlespace and specifying whom the PISR System should notify when it detects the defined conditions. The eighth most prominent QA (#34) requires that COIs be flexible enough to express enemy patterns of activity that correspond to adversary TTPs. QA #35 requires the PISR System to monitor COIs in near real time. Other QAs such as #24 specify that COIs must detect specific entities such as High Valued Individuals.

To achieve these COI-specific QAs as well as achieving the overall goal of producing timely intelligence data for USMC, the SA Subsystem consists of the following sub-subsystems:

- Conditions of Interests
- Situational Interpretation
- Sensor Integration and Interpretation
- Collection Planning Assistant

The COI Sub-subsystem of the SA PLA tracks the declared persistent intelligence requirements of the users. It directs the analytics and sensors to collect intelligence data of interest. The Situational Interpretation Sub-subsystem manages the combination of analytics that generate hypotheses about the battlespace. These hypotheses both feed back into other higher analytics within the Situational Interpretation Sub-subsystem and feed into the COI Sub-subsystem. The Situational Interpretation Sub-subsystem depends in turn on the Sensor Integration and Interpretation Sub-subsystem. In addition to translating sensor input into a generic vendor neutral format, the Sensor Integration and Interpretation Sub-subsystem extracts basic hypotheses such as the position location information of a detected person or

vehicle at a point in time. The final component of the SA PLA is the Collection Planning Assistant, which is responsible for providing feedback and advice to help a collection manager using the system decide upon an optimal set of COIs to register. The rest of this section describes these major components of the SA Subsystem as well as their relations to other subsystems within the PISR PLA. The following subsections further elucidate the individual components.

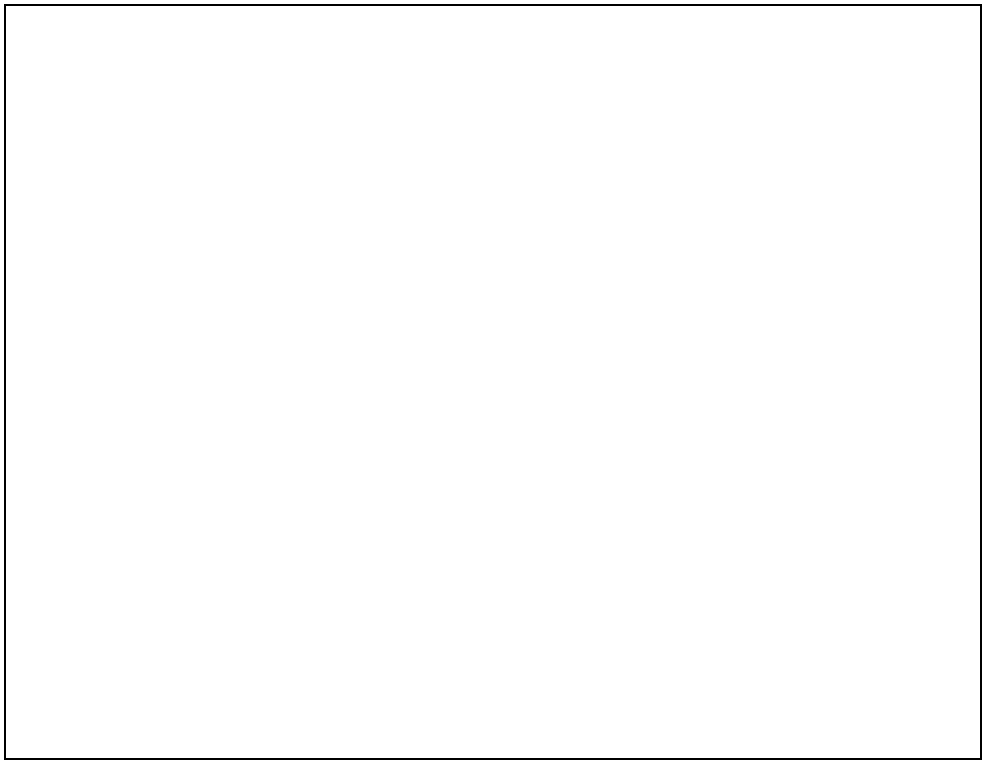

Figure 11. Situational Awareness Subsystem architecture data flow

Figure 11 depicts the components of the SA PLA and the connections involved in the primary flow of data. The user interacts with the PISR IR Editor to create, read, update, or delete (CRUD) the COIs. Upon receipt of a COI from the editor, the COI Subscription Manager stores a copy in the COI Subscription Store portion of the PISR Information Base (see Section 6) and forwards the information to the COI Validating Translator. The COI Validating Translator validates that the COI conforms to context-sensitive restrictions on well-formed COIs. The well-formed COIs continue on to the COI Interpreter, which continually monitors the PISR Information Base blackboard for matching patterns of hypotheses. When the COI Interpreter finds a match, it notifies the Dissemination Subsystem (Section 4). In order for the COI Interpreter to find hypotheses of interest, a number of Situational Interpreters analyze the set of hypotheses in the PISR IB blackboard and generate new hypotheses. Other Situational Interpreters can build upon the hypotheses of lower layers of interpretation. The lowest level hypotheses originate from the Sensor Level Interpreters that process raw data from Sensors. The remaining sub-subsystem, the Collection Management Assistant, remains unconcerned with the specific hypotheses. Instead it only interacts with the user and with the COI Subscription Store.

3.2.1 Conditions of Interest

The Conditions of Interest Sub-subsystem of the SA PLA collects the conditions that are of interest to warfighters, intelligence analysts, or other users and monitors the estimated state of the world in the PISR IB blackboard. For example, an analyst may be interested in observing the battlespace for indications of threats to a planned convoy movement. COIs represent situational triggers that, when they are observed by the array of sensor and analytic resources, indicate something of great interest to the analyst has occurred or is occurring. Such occurrences cue or alert the analyst for subsequent action to further improve understanding of the situation or to alert forces for operational response. When it estimates the conditions are met, the COI Subsystem disseminates information about met conditions through the Dissemination Subsystem. The following sections detail the design of the COI Subscription Manager, the COI Validating Translator, and the COI Interpreter.

3.2.1.1 COI Subscription Manager

The COI Subscription Manager collects information about the conditions that are of interest to users of the PISR system. Each COI is associated with the originator who created the subscription and with the users who may edit the COI definition. A subscription to a COI associates the COI with subscribers who may include people, case files, missions, or plans. Subscriptions also keep track of who imposed the COI on the subscribers (which may be the subscriber or a superior officer) and the relative priority of the notices when that COI matches relevant hypotheses in the PISR Information Base. Since the COI Subscription Manager's operations are simply the usual CRUD operations it serves as an adapter between the PISR IR Editor and the PISR IB's COI Subscription Store, in addition to initiating processing or cancelling processing of COIs. COIs may have an expiration time as well indicating the latest time of value for that kind of information.

3.2.1.2 COI Validating Translator

Users of the PISR System have roles and tasks in support of their missions. These roles, tasks, and missions naturally lead to specific interests in potential events within a geographic area during a window of time. Human sources of information, sensors, and automated analytics create hypotheses about the state of the world with some degree of certainty, some of which are relevant to the COIs subscribed to by users. The COI Validating Translator connects the high-level interests of the user to lower-level hypotheses about the battlespace produced by the analytics by generating a lower-level COI expression. This translation occurs by traversing the abstract syntax tree representing the high-level COI and producing a lower-level one. During the traversal each leaf expression representing a high-level interest is replaced by a new sub-expression containing specific conjectures combined with various logical and temporal connectives. Analytics in the system are specialists—they are able to interpret particular sub-expressions and focus their computations on associating sensor observations and other reported data that can provide evidence to confirm the truth (probabilistically) of the conditions in the sub-expression.

Before the translation can occur, the input COIs must be well-formed. While user interfaces may perform preliminary input validation in order to provide low-latency user feedback, systems tend to be more robust and more readily support multiple sources of input if the final input checking occurs after the user interface components. The checking performed by the COI Validating Translator mostly involves checking the unit labels on quantities. Distances must be in length units, expiration times must be expressed in time units, and so forth. This can be implemented via straightforward structural recursion over the inductive structure of the abstract syntax trees. The unification or constraint solving phases often found in type checkers for more sophisticated languages are unnecessary.

3.2.1.3 COI Interpreter

The COI Interpreter continually monitors the PISR IB blackboard for hypotheses that could form an instance of a COI. Since the analytics deposit their hypotheses into the PISR IB, the COI interpreter queries for either individual hypotheses that appear within the COI sub-expression or for larger sub-expressions using compound query expressions. While querying for large sub-expressions within the COI may improve performance by offloading more work onto the PISR IB Subsystem's database query engine, the COI Interpreter may still require the confidence levels of the individual leaf hypotheses in order to compute a confidence level for the sub-expression.

When the COI Interpreter receives notification of a relevant hypothesis, it retrieves the relevant COIs that reference that type of hypothesis. It then substitutes the hypothesis data for the references within the COI expression. To describe this process, we say the particular hypotheses *instantiate* the generic COI. Multiple combinatorial instantiations may be possible for COIs with multiple references. The COI Interpreter then simplifies the COI

expression through partial evaluation and stores the simplified COI. Simplified, partially instantiated COIs are also candidates for further instantiation and simplification. For example, if a user was initially interested in at least two people approaching a location *b*, but separated from one another by at least a given distance *d*, once the PISR System detects a single such person at location *x*, the simplified, partially instantiated COI is merely interested in one more person approaching location *b* but at least *d* distance from *x*.

In addition to constraints on geospatial position location information, COI interpretation must be cognizant of several notions of time for hypotheses. First, a hypothesis has a time when it occurred. The Sensor Web Enablement initiative's Observations and Measurements standard defines *sampling time*–the time at which the measurement applies. Hypotheses also carry a *result time*–the time at which the procedure producing the measurement completed. While some sensors provide near-real-time detection, informants may mention information they noticed days ago. The result time may also differ from the insertion time when the hypothesis enters the automated PISR system. For example, a Marine may need to return to the Forward Operating Base (FOB) to submit an after action report. Finally, as with COIs, a hypothesis may have a latest time of value (LTOV), which is the time after which the event is no longer of interest and should be discarded.

Simplified COIs reach several end states. COIs that simplify to *True* generate alerts. COIs that simplify to *False* (i.e., due to unsatisfied constraints) are discarded. COIs that expire due to the expiration time associated with the COI or due to the latest time of value expiring are also discarded.

When the COI Interpreter detects an instance of a COI, it forwards the COI along with the pedigree describing how the pattern variables of the COI were bound to specific hypotheses to form a COI instance. This, combined with the subscription information, informs the Dissemination Subsystem which destinations desire the data.

3.2.2 Situational Interpreter

The Situational Interpreter consists of a collection of analytics that consume and produce hypotheses within the PISR IB blackboard. The Situational Interpreter also provides control mechanisms to prioritize the scheduling of analytics in order to maximize the expected value of the resulting hypotheses. The first layer of analytics consumes hypotheses about positions, locations, and rudimentary observable features of entities in the battlespace such as people, vehicles, equipment, infrastructure, or events. These often produce fused results that combine hypotheses from different sources correlated in time to produce an estimated state of the battlespace. The next higher layer of analytics consumes hypotheses about states to produce hypotheses about behaviors over time. The detection of behaviors may require robust reinterpretation of the individual states based on estimations of the likely misclassification of one state as another state or to fill in missing (undetected) states. Some analytics further consider the behavior hypotheses to discover anomalous behaviors that indicate suspicious or hostile activities. Figure 12 depicts the typical levels of interpretation.

In addition to producing new hypotheses, the COI Subsystem and some analytics also maintain the pedigree of

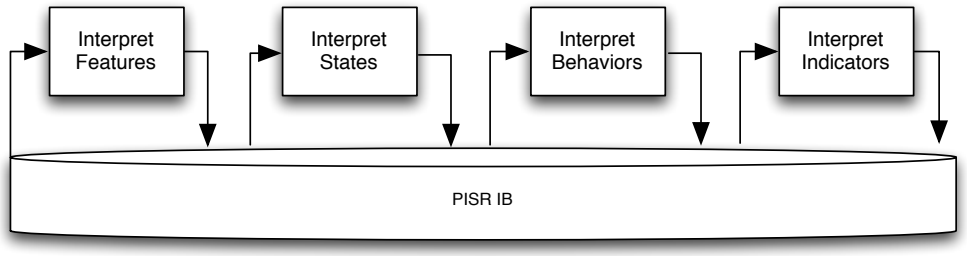

Figure 12. Levels of situational interpretation

the chain of inferences behind each hypothesis. This information can effectively provide more information from which a human can further ground and refine their personal assessments of believability or estimation of value. Even with analytics that do not maintain their chains of inferences due to the additional complexity, the SA Subsystem still tracks the sensor interface or analytic that produced the hypothesis. The SA Subsystem's desire for tracking pedigree must be balanced against the open architecture and need for an ecosystem of pluggable analytics which may not be designed with the PLA in mind.

3.2.3 Sensor Level Interpreter

The Sensor Level Interpreter performs two main functions. The first is sensor integration. Ideally, sensors should already conform to the Sensor Web Enablement's Observations and Measurements standard.[22] Until that occurs, however, the Sensor Level Interpreter subsystem must provide adapters from vendor's proprietary formats. The second function of the Sensor Level Interpreter is to extract basic hypotheses from the raw sensor input. For example, an analytic might detect images of people within video frames from an EO or IR camera and estimate soft biometrics about the person detected. This layer discards large volumes of information that could otherwise inundate the system and reduce its performance.

3.2.4 Collection Planning Assistant

The Collection Planning Assistant supports end users in choosing the COIs to which they should subscribe and planning PISR asset allocation to best collect on those COIs. As with many optimization problems, the first step in finding an optimal configuration of COIs and asset allocations is to measure the value of a given such configuration. Given the anticipated potential enemy Courses of Action determined during IPB, the Collection Planning Assistant suggests COIs that would detect indicators of these Courses of Action. In addition, the Collection Planning Assistant also suggests a PISR asset allocation that will minimize false positives (inaccurate detections) and false negatives (missing events that it should have detected). Providing a reasonable estimate of the expected false positive and false negative rates helps Marines determine how much to trust the system.

3.3 Situational Awareness Subsystem Interfaces

The Situational Awareness PLA interfaces with other subsystems within the PISR PLA to form a coherent whole. The next two sections describe the interfaces the COI Subsystem provides and requires respectively. All user interfaces assume a remote procedure call protocol when interacting with components in other processes.

3.3.1 Interfaces to Subsystems Internal to the PISR System Provided by the SA Subsystem

While the SA PLA utilizes several interfaces from other subsystems, the only interface it provides to other subsystems of the PISR PLA is for the management of COI subscriptions. The Sensor Web Enablement Working Group of the Open Geospatial Consortium specifies a Sensor Alert Service for publishing and subscribing to alerts. The SA PLA extends the Sensor Alert Service specification to subscribe to COIs rather than direct sensor output.

3.3.2 Interfaces to Systems External to the PISR System

While the SA PLA interacts with users indirectly via the UI Environment Subsystem and Dissemination Subsystem, the only direct external interface to the external world is via the Sensor Integration interface. While many vendors provide proprietary and often under-documented interfaces, the SA PLA specifies standardized sensor interfaces that sensor components must adopt or be translated into. The Sensor Web Enablement Working Group specifies several relevant standards. For example, the Sensor Model Language (SensorML) describes sensors including their locations and tasking interfaces, while the Observations and Measurements standard models the hypotheses produced by sensors.

[22] Refer to http://www.opengeospatial.org/standards/om.

4 Dissemination

4.1 Introduction

Dissemination processing integrates activities of the MCL Subsystem and the PISR IB Subsystem to pass information to warfighters through appropriate interface mechanisms. This section describes how the MCL Subsystem and PISR IB Subsystem interact to effectively and efficiently disseminate information in the PISR System. *Dissemination*, according to JP 1-02, is the "*conveyance of intelligence to users in a suitable form*". For the PISR System, we look at dissemination in a broader sense to include both the movement of information among the components of the PISR System as well as to users and external systems. For clarity, we break this broader definition into three concepts:

- Messaging – Creation and management of internal PISR messages that do not have an end-user in mind, rather they are from subsystem to subsystem. These are information packets such as feeds from a TRSS sensor about person detected events to the Sensor Level Interpreter Sub-subsystem.
- Alerting – Movement and management of PISR messages that are directed to users as high-value information. Alerts are artifacts of analyzing various messages within the system. When the PISR System determines that a user needs to be notified of some information, that information is disseminated to appropriate devices and transformed into a human-digestible format for one or more users.
- External Dissemination Component (EDC) – Components that support alert dissemination. This is a general contract for a component to conform to in order to receive alerts from the PISR System. The EDC is then responsible for translating the alert from the internal PISR alert format to the format required by the external dissemination component.

  Dissemination Management, as defined in MCRP 5-12C,

  "Involves establishing dissemination priorities, selection of dissemination means, and monitoring the flow of intelligence throughout the command. The objective of dissemination management is to deliver the required intelligence to the appropriate user in proper form at the right time while ensuring that individual consumers and the dissemination system are not overloaded attempting to move unneeded or irrelevant information. Dissemination management also provides for use of security controls which do not impede the timely delivery or subsequent use of intelligence while protecting intelligence sources and methods."

The MCL Dissemination Management Module (DMM) (Section Key Component Functionality), as part of the Process and Resource Optimization Management Sub-subsystem (PROMS) Sub-subsystem (Section 5.2.5), handles dissemination management by producing dissemination guidance plans that are distributed to PISR components for processing and implementation. The MCL, under the Alert Management Sub-subsystem (AMS) (Section 5.2.4) plans alert routes to a variety of EDCs. This section describes the general contract for creating a new EDC, as well describing some of the potential EDCs that are anticipated. Examples of these EDCs are Cursor on Target (COT) systems, text messages, email, and Internet Relay Chat (IRC) messages. The PISR IB Intelligent Distribution Sub-subsystem (Section 6.2.2) is responsible for PISR messaging and provides support for connecting additional EDCs to the PISR System.

For brevity, the details covered in the MCL and PISR IB will not be repeated here, but interested readers should refer to the respective sections listed above.

Dissemination addresses a core set of requirements derived from *MCWP 2-2 MAGTF Intelligence Collection*. These requirements are critical to a successfully operating PISR System and include:

- Dissemination Planning:
  - Processing the statement of intelligence interest (SII), IRs, CCIRs, COIs, and other inputs to derive effective and optimized dissemination schemes and dissemination plans
- Dissemination Execution:
  - Efficient dissemination of collection results and other intelligence products

- Efficient movement of raw and processed data and information among the components of the PISR System in support of PISR operations

- Dissemination Management:
  - Monitor the operation of the dissemination process, the state of relative system resources and environmental conditions to provide active dissemination management and to adjust the dissemination schemes and dissemination plans as appropriate

All other subsystems of the PISR System should be capable of receiving, interpreting, and (as appropriate) implementing the dissemination schemes and dissemination plans provided by the MCL DMM.

4.2 Dissemination Architecture

Dissemination is divided into three different concepts. It is not intended for Dissemination to be a subsystem in its own right; rather, it is the glue that ties together the MCL and PISR IB subsystems. Dissemination is described in this separate section to add clarity to its basic goals. As mentioned in the introduction, these concepts are messaging, alerting, and external dissemination components.

4.2.1 Messaging

Messaging is a key concept in the PISR System architecture. Communications among various subsystems requires an agreement on how information is passed from one system to another. Messaging fulfills that need without the requirement of having each subsystem know and directly communicate with each subsystem interested in the artifacts that they generate. For example, when a TRSS sensor detects an event, it can package that information into a message for the PISR IB to handle and give to the appropriate subsystems that are looking for that information (i.e., those that have registered subscriptions for that kind of information). The TRSS sensor does not need to know what additional analytics are needed to use its events; rather, it knows the messaging contract established by the PISR IB to notify any interested party through naively publishing a message. Any system that wants to publish an alert can package the information into a proto-alert message for the MCL to pick up and disseminate accordingly. The PISR IB handles messaging between different PISR subsystems by using the publish/subscribe paradigm to decouple message-generating subsystems from subsystems that are utilizing those messages. Messaging is further augmented by another system, the MCL, whose job it is to analyze what is actually useful to the PISR System as a whole in accordance with the current information optimization goals of the system. MCL issues guidance on what information needs to be distributed, what information should be collected, and what activities should be run to generate valuable information.

4.2.2 Alerting

Alerting is a special case of messaging where the message's target is a user or set of users instead of one or more component(s) of the PISR System. Alerting demands its own handling because, rather than just utilizing system priorities and policies, a user's properties and information requests must be considered as well. If an alert request is to be sent via text message and one of the target users does not have a phone number, then an alternative dissemination method needs to be selected. Picking the appropriate way to contact a user depends on what network alerting capabilities are present, what types of messages a user can handle, and their personal preferences for being alerted.

Alerting takes the form of a special class of message that needs to be further processed to be delivered to the correct external dissemination components for delivery to the user. The AMS does this additional routing of an alert as well as figuring out who is actually interested in an alert. While the interested user might be listed in the proto-alert, sometimes alerts need to get sent to others depending on their personal preferences or the preferences of a group they belong to. Any system can create a proto-alert message and post it to the PISR IB for the MCL AMS to further route to the appropriate user(s).

4.2.3 External Dissemination Components

External dissemination components (EDC) are components that handle a processed alert to actually send to a user or set of users. The primary contract of an EDC is that they need to be able to take a processed alert message and transform it into the appropriate messaging format for final delivery to a user or set of users. An EDC registers with the MCL Registration Management Sub-subsystem (RMS) (Section 5.2.1) the information required for it to actually process an alert to perform delivery to a user. Information takes the form of what user properties are necessary for an

alert to be processed as well as what additional data is necessary for the EDC to perform its function (e.g., the Internet Protocol address of a Simple Mail Transfer Protocol [SMTP] server or the authentication credentials for it). It is the responsibility of AMS to verify that a processed alert has all the information an EDC says it needs before forwarding the alert messages via the PISR IB to the EDC. For example, the AMS needs to verify that the users to which it is invoking the SMTP EDC to send an email all have email addresses.

Once an alert has been delivered to the EDC for processing, it is the responsibility of the EDC to verify as well as it can the delivery, receipt, read status, and action taken as a result of the alert. It is the responsibility of the EDC designer to incorporate as many of these alert states as possible in an implementation. The Health Management Sub-subsystem (HMS) (Section 5.2.2) provides a way for the EDCs to send these state changes as status messages to be discovered by the AMS. It is the responsibility of the AMS to monitor the HMS for these status changes. Not all EDCs can support all the different states; however, an EDC should attempt to support as many different alert states it can know about. By utilizing the messaging infrastructure, EDCs can break up state change messages into different logical components. For example, read receipts from an email would probably be handled by a different component than the one that sent the original email. The two different components would be able to publish their respective state changes for an alert and allow the AMS to subscribe to and correlate those state changes.

The following subsections present a small set of possible EDCs. Additional EDCs are encouraged and can be added easily as long as they register their required data needs along with their capabilities.

4.2.3.1 Internet Relay Chat (IRC)

Internet Relay Chat (IRC) is a well-established protocol for sending messages to users via a chat server. There are a variety of end-user clients that enable a user to connect to a chat server and specify a username or nickname to which they can send or receive messages (mIRC is a commonly-used windows-based version of such a client). IRC revolves around the idea of channels where users can chat with one another in a semi-public forum. Users can join channels that they are interested in, given that the room is not full and that they have been given authority.

IRC Messages come in two different forms, one is a private message in which only the intended receiver can see the message from another user. Another is a semi-public message in which a user posts a message in a channel for all subscribers of that channel to see. An IRC EDC comes in two forms, matching the public and private messaging formats available. Both IRC EDCs have common limitations. There are flooding controls on an IRC server that prevent a single user from sending more than a few messages a second as well as limits on the maximum payload size of the message (e.g., about 400 characters). Despite these limitations, IRC is a good way to post notifications for broad consumption.

For the broadcasting a message via a channel, the IRC EDC only needs to know the name of the channel. Generally an IRC EDC is tied to a particular IRC chat server and having the name of a channel is sufficient information to send a broadcast message. Unfortunately, there is no guarantee with this method that a message will reach an intended target, or that the message is read at all. This limits the usefulness of IRC broadcast messages to non-critical but interesting messages. Only protocols external to the PISR System can alleviate this issue.

For private messaging, the IRC EDC only needs to know the IRC nickname of the user for which the message is intended. Generally, an IRC EDC is tied to a particular IRC chat server and having the nickname of a user is sufficient information to send a private message to that user. Unfortunately nicknames can be hijacked easily, so unless sufficient protocols are in place outside of the PISR System, there is no guarantee of delivery to the intended user.

4.2.3.2 Simple Mail Transfer Protocol (SMTP)

SMTP, more commonly known as email, allows users to get notified via an email address. SMTP allows notification to be sent directly to a user's email inbox. Access to email is generally password-protected and can be securely encrypted to be utilized as a secure base for guaranteeing notification to a user. SMTP also allows for additional protocol options such as read receipts which allow a system to confirm delivery of a message to a user and to verify that the user has actually opened the email for reading. Email messages have a very large payload limit and allow for additional attachments of information such as images to convey a great amount of information.

An IRC EDC needs the email address of a particular user to deliver a message. Depending on the capabilities of the receiving client mail system, many additional pieces are available for the system to determine if messages are read in a timely manner. Since email addresses can be tied securely to an individual user, concepts like non-repudiation of receipt of an alert can be accomplished.

4.2.3.3 Short Message Service (SMS)

SMS, commonly known as text messaging, allows users to get notified via a text message on their mobile phone. For immediate notification of an issue in a communication-rich environment, SMS is a potential solution and a good capability to have. It does not offer state feedback such as a read receipt like email, but most mobile phones support the reception of text messages. Unfortunately, SMS only offers an extremely limited payload of 144 characters per message, so SMS alerts must package their information concisely.

An SMS EDC requires the mobile phone number of a user. However, there is no guarantee with this method that a message will reach an intended target user, the message is read at all, or that the user who is accessing the SMS is the correct user (stolen phone). Due to the speed and immediacy of alert notification, it does allow for critical messages to potentially get to the target user or users as soon as possible. If that speed saves lives, it may be worth utilizing.

4.2.3.4 Cursor on Target (COT)

Cursor on Target (COT) is a specific system protocol that allows for "dots on a map" to show up on a FalconView application and other COT-enabled systems. It is not targeted for a particular user; rather, alerts that target COT are generally for any user assessing the tactical situation in operations centers. COT messages require a location, confidence, observed event time, and a description of the event to work properly. One COT EDC is the FalconView server. A FalconView server requires one or more FalconView applications to be running. Each FalconView application provides a view of all COT messages that have been generated. FalconView applications assume that interested users are monitoring continually the FalconView application. Users are then responsible for reacting to those messages in accordance with established procedures.

5 Management and Control Layer Subsystem

5.1 Introduction

The purpose of the Management and Control Layer (MCL) Subsystem is to optimize the employment of resources across a PISR System. A PISR System operates under an objective of delivering the greatest quantity of highest value information when and where it is needed for the warfighter by leveraging the ability to take into consideration all information needs, capabilities, capacities, mission priorities, unit priorities, and any other defined constraints. The PISR System MCL determines how to best employ the PISR resources for each period of time. The PISR MCL will issue guidance in the form of plans, configuration, prioritization, and rules to the components of the PISR System. Those components then have a responsibility to understand and implement each element of guidance.

There are existing systems that analyze how information flows through a system or set of systems. What is missing is the feedback to those systems to prioritize their information flow based on a global set of objectives rather than their local set of objectives. MCL addresses this flaw by requiring subsystems within the PISR System to understand the global information objectives and optimizations to reach those optimizations, or at least to obey an external information manager that knows those objectives and optimizations. Technology exists to do this global management; MCL expands the scope of that technology to incorporate heterogeneous subsystems.

Limited and constrained resources may make it impossible to process and deliver all PISR information to the warfighter immediately. Human processing limitations make delivering all captured information undesirable. The MCL mitigates both these problems by optimizing data collection, information analysis processes, and information dissemination towards delivering the highest value information without overwhelming the human user(s). The MCL manages the resources as three optimization sub-problems: (1) collection resource allocation; (2) process control; and (3) information dissemination. Each of these optimization sub-problems is serviced by a specialized optimization module appropriate for that problem type. A fourth optimization module provides oversight of the three specialized optimization modules with the objective of balancing and tuning their operation to achieve optimized global performance. Global performance is measured by the satisfaction of information needs of the PISR System operators against the resources available, in an attempt to give the best value based on a set of defined constraints, priorities and policies.

MCL addresses several different requirements for a successfully operating PISR System. These requirements include:

- Ability to discover the failure of an internal or external component to perform as expected.
- Ability to know the activities and processes the PISR System can accomplish.
- Ability to know the resource utilization of a process.
- Ability to evaluate the value of information gained from a process.
- Ability to know current resources and their utilization.
- Ability to plan data distribution for a component to perform its function.
- Ability to orchestrate the PISR System activities to optimize the flow from data collection to distribution.
- Ability to evaluate the progress of a process within the PISR System and know if it is unable to complete its function due to lack of required resources.
- Ability to start and stop an activity in the PISR system.
- Ability to prioritize resource allocation to support highest perceived value activities.
- Ability to analyze an activity or process to update estimates of resource consumption and information value gain.
- Ability to disseminate alerts to the appropriate users in the appropriate timeframe.

The above requirements have been driven by both the defined A-priority QAs in Appendix A and the implied QAs necessary to get the system functioning properly. Specifically, the above requirements have been driven by QAs 13, 34, 114, 20, 93, 17, 109, 97, 60, 56, 75, and 110. All of these requirements deal with how the system delivers

information to users in response to their interests, as well as how to make communications between different systems robust and effective. Effectiveness is measured by how well the system satisfies users' information needs.

In order to accomplish its goals, the MCL oversees the development and execution of plans and guidance to perform collection, processing, and dissemination of information accessible to the PISR System. Optimization of these targets requires continuous monitoring and adaptive planning. As such, the MCL constantly monitors, evaluates, and revises its guidance for various components as the overall situational picture evolves. As a part of monitoring the PISR System, the MCL is responsible for taking automated corrective action for issues discovered in the PISR System and notifying responsible parties if a corrective action requires human intervention. When information needs, missions, resource availabilities, or other key components within the PISR System are updated, the various guidance plans are adjusted to reflect the updated understanding of the situation. As new high priority supportable information requirements (SupIR) emerge, MCL may bump, preempt, or tailor existing tasking to seek optimized delivery of highest perceived value information. Figure 13 illustrates the optimization lifecycle. Key to the optimization is the ability for the MCL to analyze the big picture of various systems interacting with one another, knowing what kinds of information each system needs to perform its functions, and making sure that each system gets the information it needs while taking into consideration health information such as utilized bandwidth. For example, it the system knows that HVI detection is currently the most important goal of the PISR System, then the MCL is responsible for making sure that systems such as Progeny have a priority on bandwidth and processing power.

MCL must support a distributed infrastructure, since not all pieces of the subsystem need to or are desirable for them to run on a single computer. Each sub-subsystem may have independent dedicated resources. Additionally, pieces of each sub-subsystem of MCL can be distributed throughout the PISR System's network to aid in the collection of information and distribution of tasking. Each of these local instances of MCL logic is considered a MCL node. An example of this would be a node that knows how to parse health status messages and report those messages to the Health Management Subsystem (HMS) interface. This node would be collocated or closely located with a sensor that is reporting health information. Health status information can then be passed to the node; the node can then analyze the information, only forwarding on vital info to the main MCL servers.

In addition to the local optimizations, the PISR architecture supports different PISR Systems communicating with one another across regions, networks, and security enclaves. MCL will broker information between PISR Systems so that resources might be shared at a global level. Local optimizations would take precedence unless tasking from a higher echelon preempts the local requirements. The PISR systems will be able to negotiate, task, and evaluate different plans at a global level to allow the best value of information for all users given the constraints of time, bandwidth, and availability.

Section 5.2 provides an overview of the MCL design at the subsystem and component level. Subsections of Section 5.2 provide detailed information about the different sub-subsystems of MCL. Section 5.3 provides an outline of the external and internal interfaces between each system and MCL as well as among internal MCL components. Section 5.4 describes the functionality required by a user interface to fully support the MCL. Section **Error! Reference source not found.** describes how MCL can evolve in successive iterations as the capabilities described in the architecture are worked out.

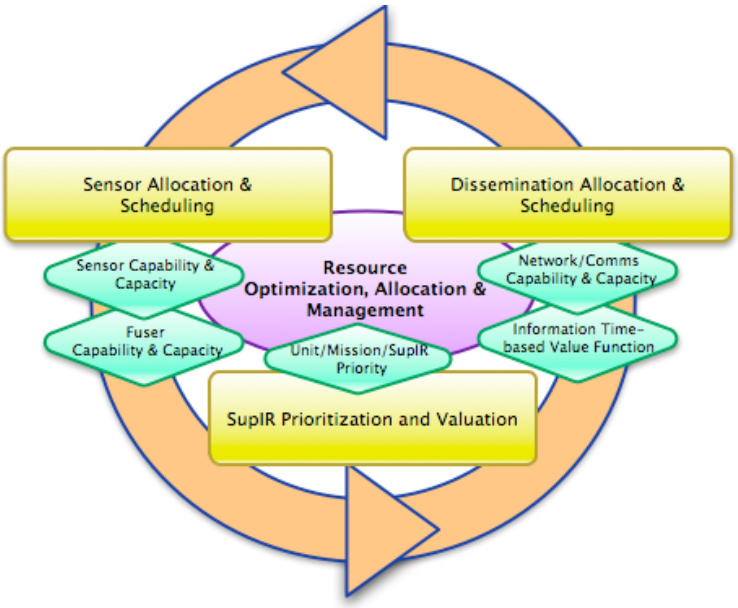

Figure 13. MCL optimization lifecycle

5.2 MCL Subsystem Architecture

The MCL architecture includes five sub-subsystems, with one sub-subsystem split into four additional modules for clarity. The logical separation of the MCL architecture is shown in Figure 14. Each sub-subsystem is described in greater detail within its subsection. The subsystems are listed below roughly in their order of dependency:

- Registration Management Sub-subsystem (RMS)
- Health Management Sub-subsystem (HMS)
- Policy Management Sub-subsystem (PMS)
- Alert Management Sub-subsystem (AMS)
- Process and Resource Optimization Management Sub-subsystem (PROMS)

PROMS is composed of four modules:

- Process Management Module (PMM)
- Collector Allocation Management Module (CAMM)
- Dissemination Management Module (DMM)
- Optimization Balance Management Module (OBMM)

Each of these sub-subsystems plays an important part in establishing the required contracts to handle the identified requirements smoothly.

The RMS defines a contract for each and every component in the PISR to notify the MCL of the capabilities, location, activities it can perform, information necessary to work, information produced, and potential status information of the component. Registration is the first step to a fully-functioning PISR system. Once each component of the PISR System has been registered, either directly or through a proxy, MCL can reason about that component in the other MCL sub-subsystems.

The HMS utilizes the information gathered by the RMS to query about the health or to monitor for certain expected health/status messages from components within the PISR. Monitoring the self-reported health of each component allows other MCL sub-subsystems to reason about the status and behavior of resources, or at the very least to send off alerts via the AMS to system administrators about abnormal behavior.

The PMS stores policies for the system so that the behavior of the system can be modified on the fly. Policies are answers to the questions such as: “What form of communication is the default?”, “What is our threshold for false positives?” or “How long since the last status update do we wait until we think a sensor is dead?” Policies allow operators to tune MCL behavior and performance at runtime to best match the mission and situation.

The AMS provides the ability to route alert communications to the operators based on their expressed interest and utilizing one or more designated communications channels. The AMS also has the ability to route alert communications based on roles, groups, or current operators, dynamically resolving who and how alerts should be delivered. If some event of interest happens that requires generation of an alert (e.g., an HVI is located), AMS can broadcast the notice, or alert a single individual (e.g., send a text message informing a unit that the HVI is near its location).

Finally PROMS allows us to optimize the process flow, allocation of collection assets, and internal messaging of information across the PISR System to disseminate highly valued information. The OBMM goal is to balance the PMM, CAMM, and DMM to make sure that they are collectively producing an optimized set of guidance that satisfies resource constraints while still delivering the near optimal system performance. Each module is interdependent upon the others, with OBMM negotiating and orchestrating the different optimization parameters and orchestrating the tuning of objective functions.

All communications to non-PISR specific systems are accomplished through messaging. Message specifications establish the contract that a system needs to conform to in order to be included in the PISR System. Intra-subsystem communications will also be established through messaging; however, there are also established interfaces with which the PISR Subsystems can interact. Messages are passed between systems through the PISR IB, leveraging the PISR IB’s publish/subscribe design. For example, when some component needs to send registration information to the MCL, it will publish its registration information to the PISR IB utilizing the augmented SensorML specification; the PISR IB will deliver that information to the RMS of the MCL through the subscription mechanism.

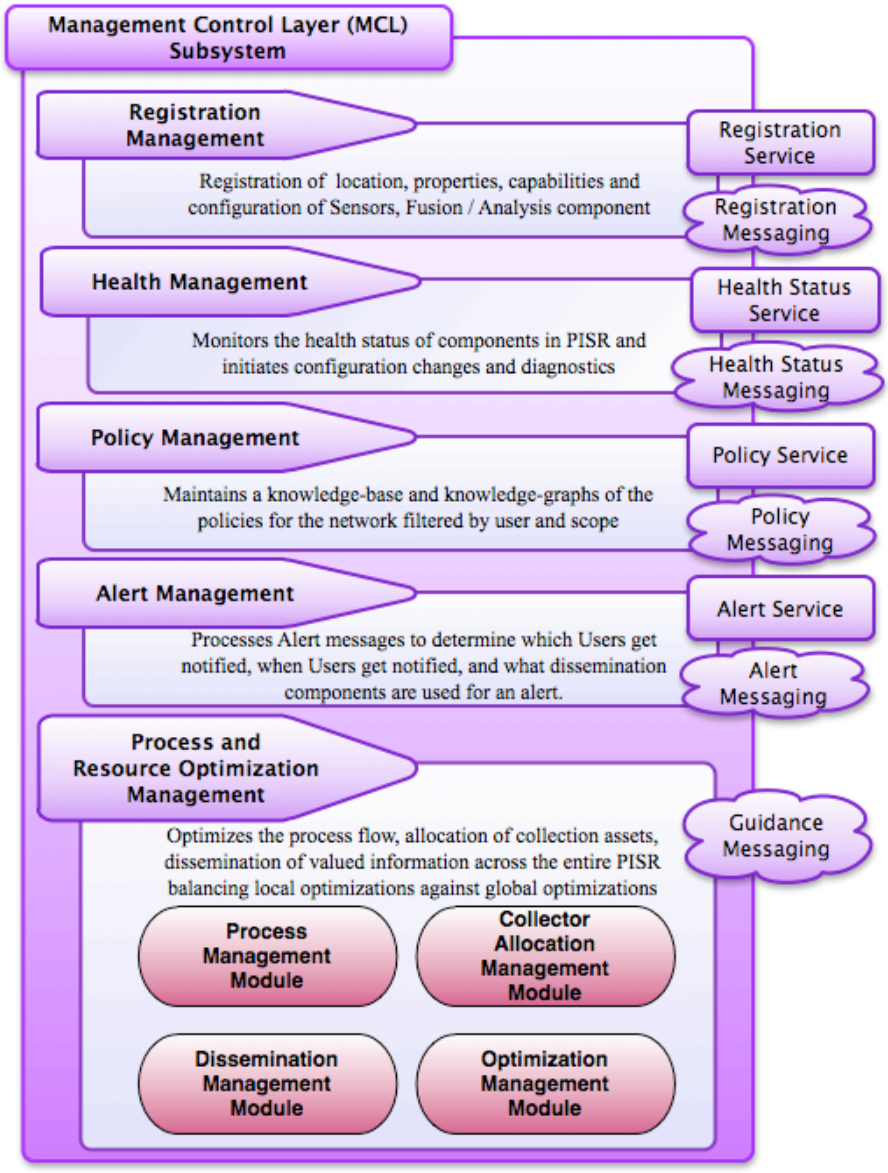

Figure 14. MCL Subsystem architecture diagram

5.2.1 Registration Management Sub-subsystem (RMS)

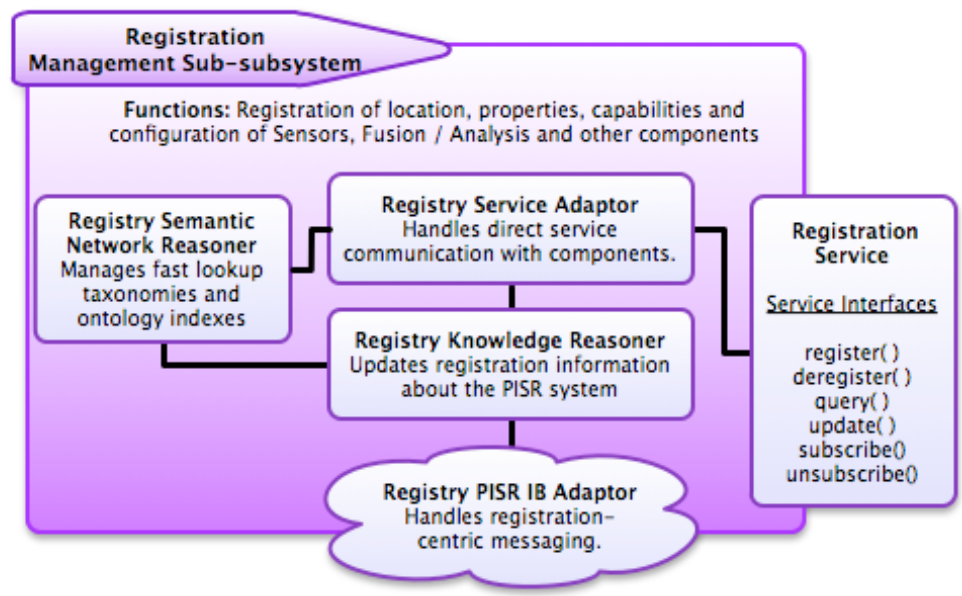

Figure 15. Registration Management Sub-subsystem architecture diagram

5.2.1.1 Component Description

The RMS (Figure 15) is responsible for maintaining a registry of all PISR System users, components, and resources with their capabilities in a searchable structure. This section refers to all users, components, and resources as "registered elements." The RMS aids in the collection of location, properties, configuration information, potential activities, and other capabilities of registered elements within the PISR System, collectively called attributes. This functionality is provided through a simple registration message posted to the IB. This registration message follows the SensorML specification with small augmentations to support the PISR System MCL requirements.

RMS provides an interface by which interested components, primarily internal to MCL, can query registration information. This allows subsystems to discover registered resources such as sensors, data processing modules, and physical users. Each registered resource has activities that it can perform. Each activity has its data requirements specified as well as the information that performing the activity will create. By registering potential activities, the PROMS will know what systems to task to get specific activities done as well as knowing the overall impact of trying to perform that activity. Having a centralized repository for users will allow the HMS to know which users are presently considered a part of the PISR system so that it might be able to manage dissemination to those users.

RMS also categorizes each registered element into a known taxonomy of sensors and systems. It does this based off the capabilities registered by a system. This allows unknown systems and sensors to registered and broadly categorized for utilization optimization purposes.

RMS can be split up into multiple instances to perform registration management locally to a subsystem that is accessing it. Leveraging the distributed nature of the PISR IB, different RMS instances can be optimized towards providing different registration information. For example, the SA subsystem might leverage the RMS to query and discover sensor capability information. The HMS would leverage the RMS to query external dissemination component capabilities. These two pieces of information are disjoint and the RMS may be divided and distributed to provide local access to locally important pieces of registration information. While all RMS pieces have access to the entire registration body of information, smart local caches would provide fast access to important registration information.

5.2.1.2 Key Component Functionality

The RMS has four primary functions:

- RMS manages the metadata about registered resources. This function enables a component within the PISR to perform capabilities-based and attribute-based queries for registered elements on the PISR System.
- RMS maintains addressing information to allow any component performing discovery to locate and initiate an interaction with the registered component or resource.
- RMS supports the automatic discovery of a newly registered resource or capabilities on an already registered component.
- RMS supports the removal of registered resources and notification of parties interested in their removal.

For example, a new sensor is installed in the PISR System. The sensor is a high-definition video camera with a limited range of view. It cannot be moved, but the direction it points in can be altered. The sensor (or the SA instance to which the sensor is tied) would be responsible for registering itself with RMS. The camera's configuration attributes such as resolution, maximum viewable distance, zoom, location, potential field of view, orientation, etc. are all registered. Along with those attributes, interaction capabilities also need to be specified (e.g., how the sensor status can be queried). This will allow any other resource within the PISR System to discover that this new resource exists. When a new information need is generated, this camera can be discovered and utilized to fulfill that information need.

Resource Metadata attributes for each registered element are provided through an extensible resource metadata description. Resource metadata comprises the following information, each component of which supports discoverable queries:

- Network location – This is where on the network a registered resource can be found. This can be something like the IP Address and communication port, domain name, network name, or some other identifiable information that can be mapped to an associated communication mechanism.
- Capabilities – These describe functions and benefits a particular registered resource can provide. For sensor information this will be something like 100m x 100m high resolution aerial video within a 10km x 10km area. Every registered component has a set of discoverable capabilities.
- Properties – These are attributes of the particular registered resource. These are things such as the specific camera model, the physical location of a static component, or the maximum capacity of a database.
- Configurable attributes – These are configurable attributes of a particular registered resource that can be modified by other resources in the PISR System. Each configurable attribute specifies how that attribute can be changed. For example if a camera is pointed in direction Y, a configurable attribute will be "Camera Direction", and it will state the message required to change the direction to X.
- Activities – These are what tasks or processes can be completed by the registered resource. Activities are defined in terms of what data is consumed and what data is produced. Activities have an associated cost and value that is modified as the system runs for optimization calculations. For example, sensors may have no required consumption data, so can effectively be the start point of any workflow process. Analytics would consume some data to perform their function and would produce data as an effect of running the activity. External Dissemination Components might consume data, but produce no PISR information artifacts. As the PISR system operates, cost and value are updated to reflect how each of the activities performs in respect to the overall goals of the PISR System.

5.2.1.3 Registry Semantic Network Reasoner Specification

The Registry Semantic Network Reasoner indexes various registered resources by capabilities and other functionalities to allow for fast lookup of registered resources conforming to some standard sets of attributes. It is responsible for deciding which attributes need indexing. Attribute indexing is a matter of policy established in the PMS as well as some intelligent reasoning over past queries to determine which attributes get utilized enough to require the initial overhead and additional space requirements of indexing to offset future query workloads.

While specific implementations of the RMS may use different ontology languages, an initial approach can adopt the Web Ontology Language Description Logics (OWL-DL) for the classification of various registered resources in the PISR System. This will allow for practical reasoning algorithms about the attributes and classification of various registered resources. Also it provides an easy way to grow the relational and classification mapping with several standard tools.

5.2.1.4 Registry Service Adaptor Specification

The Registry Service Adaptor is responsible for handling the transformation of registration information into the standard format that the Registry Knowledge Reasoner can utilize. It handles any inbound messages that conform to the Registration Service Interface. It is responsible for handling registry subscriptions and handling any outbound queries. Initially all queries will be attribute-based. For example "show me all registered resources where $attribute_i$ equals $value_j$". This allows for simple tuple syntax to query for various registered resources. Eventually this query language should be extended to fulfill any valuable queries that could be expressed using OWL-DL.

The Registry Service Interface is primarily intended to be an internal interface used to query the RMS directly for information. In general, external systems will interact with the RMS via messaging, specifically with the modified SensorML specification.

5.2.1.5 Registry Knowledge Reasoner Specification

The Registry Knowledge Reasoner manages the current state of registered resources in a PISR System. The Registry Knowledge Reasoner maintains a local cache of resources that are identified by the Registry PISR IB Adaptor or the local registry service adaptor and smartly maintains that cache for quick information access. The Registry Knowledge Reasoner is responsible for evicting, storing, and updating information for the registration cache as required. It leverages the PISR IB as necessary. The goal of the Registry Knowledge Reasoner is to perform the domain-specific reasoning required for RMS operation, primarily by adding in registration domain reasoning to local cache management of information provided by the PISR IB. The Registry Semantic Network Reasoner uses the Registry Knowledge Reasoner to create the various lookup indices.

5.2.1.6 Registry PISR IB Adaptor Specification

The Registry PISR IB Adaptor collaborates with the PISR IB Subsystem to subscribe to component registration messages. Whenever a component needs to register itself with a MCL (generally when they are connected to the PISR System), it sends a registration message to the PISR IB. The PISR IB then delivers that message to the RMS as a side effect of the subscription RMS has set up for that information. It leverages the distributed nature of the PISR IB Subsystem to make sure all RMS instances have access to all registry information in a PISR System.

5.2.2 Health Management Sub-subsystem (HMS)

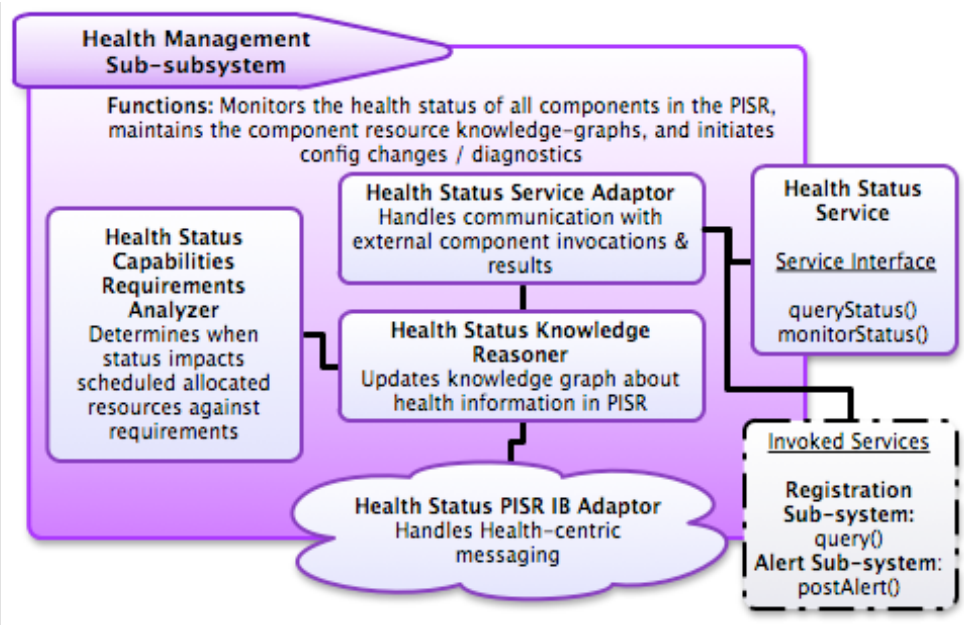

Figure 16. Health Management Sub-subsystem architecture diagram

5.2.2.1 Component Description

The Health Management Sub-subsystem (HMS) (Figure 16) monitors the health and status of all components and resources within the network. It maintains a logical grouping graph, linking various status-reporting resources within the PISR System. Registered resources are hierarchically ordered for summarization of information so that issues can be identified in an easily consumable manner. This enables the HMS to bring critical and abnormal situations to the attention of interested systems and users. It supports passive reporting by resources as well as active queries regarding the health status of specific resources as specified in the original registration of that component. Finally, HMS can send configuration request changes to registered resources. For example, HMS would be responsible for knowing how to send a control message to modify the quality of images returned by a camera to help bandwidth consumption issues.

HMS is one of the core functionalities of the MCL. Optimizations within the PROMS require the ability to know how the different components of the PISR System are performing. In addition, HMS develops alerts about infrastructure issues; if a server or sensor stops reporting or is being tasked more than it normally is, HMS can issue alerts via the AMS to a system administrator to rectify the problem.

5.2.2.2 Key Component Functionality

HMS has three primary functions:

- HMS interacts with the RMS to discover the resources which it is responsible to monitor and to determine how to interact with those resources so that it can acquire their status indicators and push configuration information out to them.
- HMS analyzes and summarizes status information focusing on reporting abnormal conditions or trends.
- HMS ensures that abnormal condition notifications are pushed to interested parties via the AMS (e.g., administrative users, commanders)

5.2.2.3 Health Status Capabilities Requirements Analyzer Specification

The Health Status Capabilities Requirements Analyzer is responsible for determining if reported health status information can or is having a detrimental effect on the PISR System. It examines the current data collection, analysis, and dissemination plans to see if there are any components being utilized that are reporting failing health or that have problematic health trends. Detection of any failures results in the Health Status Capabilities Requirements Analyzer notifying the Health Status Service Handler to post an alert to the dissemination sub-subsystem.

5.2.2.4 Health Status Service Handler Specification

The Health Status Service Handler processes incoming and outgoing messages from the service. It is responsible for interpreting the service interfaces method calls so that the Health Status Knowledge Reasoner can accurately reason about the health information requested and quickly provide that information to the requesting systems.

The Health Status Service is primarily intended to be an internal service to the MCL so that different subsystems can query the health of registered components directly. If other systems are interested in the health of a component, they can leverage this service to find out that information.

5.2.2.5 Health Status Knowledge Reasoner Specification

The Health Status Knowledge Reasoner is responsible for taking all health data being added through the PISR IB adaptor and the Health Status Service Handler to collate, summarize, and store it for later queries. It also is responsible for notifying the Health Status Service Handler when health conditions being monitored occur. Finally the Health Status Knowledge Reasoner acts as a gateway for the Health Status Capabilities Requirements Analyzer to reason over status to detect abnormal conditions that may affect the operational ability of the PISR System.

5.2.2.6 Health Status PISR IB Adaptor Specification

The Health Status PISR IB Adaptor is responsible for subscribing to and interpreting messages about status from various components. It is also responsible for posting configuration or status request messages to the PISR IB for a sensor or system to respond to.

5.2.3 Policy Management Sub-subsystem (PMS)

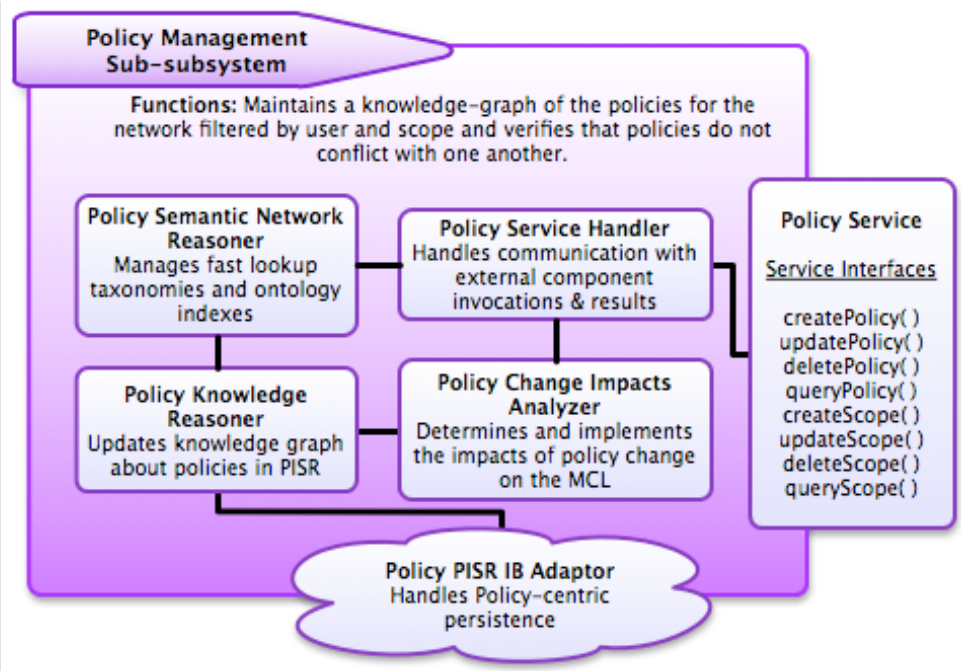

Figure 17. Policy Management Sub-subsystem architecture diagram

5.2.3.1 Component Description

The Policy Management Sub-subsystem (PMS) (Figure 17) is responsible for the management of various policies and constraints that affect the PISR System. These are policies that affect the runtime environment of the PISR System and not policies that deal with security. Those policies are separately managed in the IA framework. This is where the guidelines for various management tasks are created and managed. These policies have a defined scope as well as a value indicating for how strictly a policy must be followed. At one end of the spectrum, policies recommend how things should be done. At the other end of the scale, the PISR System is prohibited from violating certain hard constraints. For example, the policy service is responsible for determining which optimization parameters are used for the various modules within a PISR System. Any customizable options that happen within the system should query PMS to determine if there are any policies that are in effect for the function or capability they are performing.

Policies are a distributed resource within the scope of that policy instance. Global policy values can be changed on one MCL node and reflected across the network, assuming the user had such authority. Local policies can be in effect for a particular node within the network. Policy scope is highly flexible, but typically defined as global, organizational, community, or local. Examples of scope utilization are as follows:

- Global Policies – These are policies that affect every component within the PISR System. For example, one such policy could be a kill, capture, or either policy for Bin Laden. When a unit reports finding Bin Laden and tries to establish a kill plan for him, an active capture policy may say that the unit should not execute its kill plan.
- Organizational Policies – These are policies that affect a particular organizational unit. For example, when laying out audio sensors, some battalion may make it a policy that all sensors must be placed no more than 10 meters away from a road. A user belonging to that organization, standing on a road, makes a request to see if placing a sensor at his current location is okay. According to that organization's policy the user is notified that

the location is not okay. However, another unit that is utilizing the PISR System may be able to place that sensor on the road.

- Community Policies – Community policies are a dynamic policy scope that can be inclusive of particular users or units. These are intended to be policies that cut across multiple units or PISR sub-networks. For example, an audio specialist is logging onto the system. A standing policy for all Audio Specialists across the PISR might be to automatically bring them to an audio analytical screen so that they can begin their work.
- Local Policies – These are policies that only affect a single component within the PISR System. This could be a policy such as use a particular optimization engine (e.g., OE-x1) for this computer.

Policies should conform to a standard policy description language (PDL). Standard PDLs are being researched for inclusion into the PISR PLA. One such candidate is AMORD In RDF (AIR).

5.2.3.2 Key Component Functionality

The PMS supports three primary functions:

- PMS stores policies to the PISR IB Subsystem.
- PMS allows for queries of policies that affect various systems.
- PMS analyzes of the impact of policies for which there is a registered analysis engine.

The PMS does not enforce policies, rather it acts as an efficient indexing and querying engine of policies based on the scope and policy attributes. The policy manager can also analyze the impact of a policy if the system to which the policy applies has registered a resource that can analyze those policies. It is the responsibility of each system that has policies established for it to enforce those policies.

5.2.3.3 Policy Change Impacts Analyzer Specification

The Policy Change Impacts Analyzer is intended to analyze the impacts of a particular policy and see how it may affect or supersede other policies within the PISR System. It is able to detect conflicts and notify the policy maker of such potential problem areas. It allows for conflicting policies at different scopes, but not within the same scope. Scopes are verified to not conflict with any other currently created scopes and passed through to the Policy Knowledge Reasoner.

5.2.3.4 Policy Semantic Network Reasoner Specification

The Policy Semantic Network Reasoner indexes various policies by systems that they affect to allow for fast lookup of policies affecting some sub-system. All policies should be associated with a scope, with conflict resolution of policies happening by how hard the policy actually is. For example, there is a global policy to Kill Bin Laden on proper identification with a hardness value of 0.8 (high on an interval scale of [0, 1]). A local policy is in effect to Capture Bin Laden on proper identification with a hardness value of 0.9. The policy that takes precedence, all other things being equal, is the one with the higher hardness value. If hardness is equivalent, the precedence of policies is as follows: global policies, organizational policies, community policies, and finally local policies. This precedence should be configurable for a particular PISR System configuration.

5.2.3.5 Policy Service Handler Specification

The Policy Service Handler is the middle man between the policy service and the policy change impacts analyzer. It translates the various policies to the language that the network reasoner is utilizing and returns any feedback to the policy creator. For any queries it communicates with the Policy Semantic Network Reasoner in order to retrieve them as quickly as possible.

The Policy Service Interface is intended to be directly interacted with by PISR Subsystems and especially MCL sub-subsystems.

5.2.3.6 Policy Knowledge Reasoner Specification

The Policy Knowledge Reasoner is responsible for storing creating a local cache of policies for the Policy Semantic Network Reasoner to process. It receives various policies and scopes from the PISR IB adaptor and the policy change impacts analyzer.

5.2.3.7 Policy PISR IB Adaptor Specification

The Policy PISR IB Adaptor performs the persistence of various policies for future retrieval. The primary purpose of the PISR IB adaptor is to archive any policy changes for auditing purposes as well as creating a store of sample policies to choose from.

5.2.4 Alert Management Sub-subsystem (AMS)

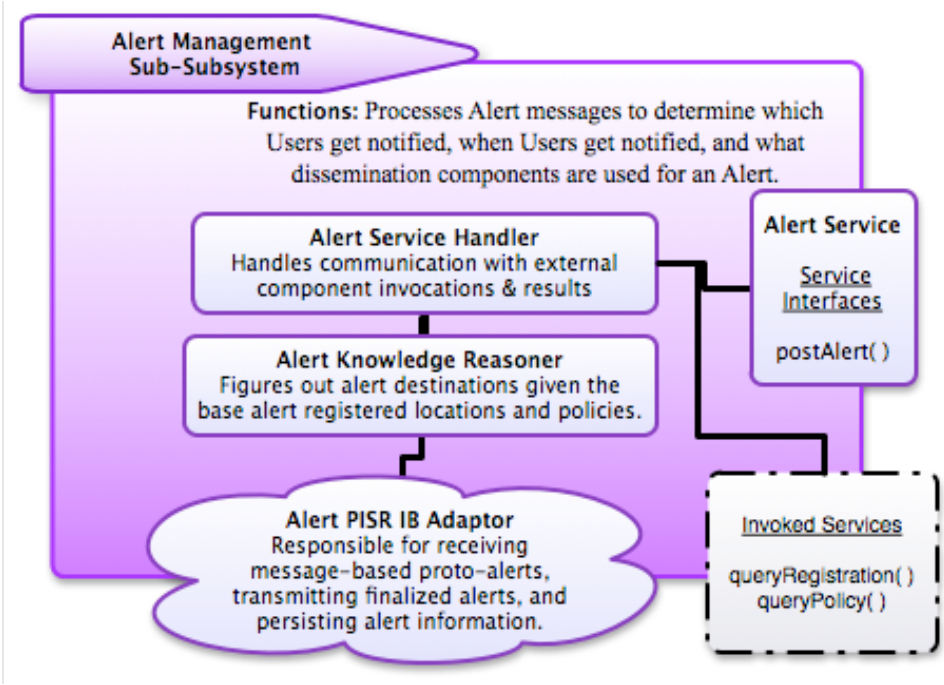

Figure 18. Alert Management Sub-subsystem architecture diagram

5.2.4.1 Component Description

The Alert Management Sub-system (AMS) (Figure 18) provides a framework to determine how to deliver alert messages to various actors within a system. The AMS receives a proto-alert from the PISR IB due to long standing subscriptions for that information. AMS then provides the where, when, and how a proto-alert needs to be delivered. This processed proto-alert becomes an actual alert that needs to be disseminated to the users. AMS provides several different handlers that act as end points to which an alert can be delivered. As described earlier, these end points can be systems such as an IRC chat server, COT Server, or an email server. AMS also determines the scope of notification, from a broadcast to all interested parties in a IRC chat room, or a single text message to a system administrator. In addition to the aforementioned end points, the AMS could have a distribution target of another subsystem inside the PISR Subsystem, such as a custom UI. Delivery is accomplished through registered dissemination components, registered as every other component of PISR through the RMS. The AMS framework allows users to register preferred means of alerting based on keywords, alert type, priority, and severity. It also supports the linkage of alerts to user groups and roles, providing dynamic alert routing for roles. For example an alert might need to be delivered to the current Watch Officer rather that the original person who created reason for that alert (such as a COI in the COI Sub-subsystem).

5.2.4.2 Key Component Functionality

The AMS has four primary functions:

- AMS specifies a well known messaging format for it to receive an unprocessed Alert or proto-alert from the PISR IB.
- AMS utilizes RMS and PMS to find out which users needs a proto-alert, when that Alert needs to be sent to them, and what dissemination components need to be sent the processed Alert.
- AMS utilizes the IB to push messages to internal and external dissemination components.
- AMS verifies the delivery of a processed alert and adapts to exceptions as necessary. Additionally, AMS supports additional alert states such as delivered, read, and acted upon if the underlying dissemination mechanism supports it.

5.2.4.3 Alert Knowledge Reasoner Specification

The Alert Knowledge Reasoner processes any alerts received and determines who needs to receive the alert as well as the delivery mechanism to get the alert to that person or persons. It does this through a combination of the alert information itself, what the currently registered user dissemination resources on the PISR are, where those dissemination resources are, and what policies have been established for this type of message.

For example, a new low-level alert (LLA) about the health of the system is received. The current policy is to queue all medium level and below health status alerts into a daily digest email and send them out at 0:00 GMT. Another policy is that health alerts should be sent to all users in the network administration group. The AMS takes the alert, reads the policies, and puts the alert on a queue to be processed later. At or just after 0:00 GMT, the deferred message queue gets processed. At that time, all deferred alerts are processed. The medium-level and below alerts that the policy had original excluded from being sent immediately, including the original LLA, are processed. In accordance to policy, all the alerts are bundled together into one alert (the digest). The AMS then figures out all the users that are now associated with this collated alert. Since the alert is supposed to be an email, all the network administrators are recorded as the alert recipients (in accordance to another policy set forth). The AMS then determines where the nearest open SMTP server is. It puts the location about the recipient component (the SMTP server) in the alert message in a way that the IB can understand. Finally it sends the newly packaged alert with component destination and user lists to the PISR IB adaptor so that it can post it to the proper dissemination component. The IB routes the alert to where it needs to go. Finally the adaptor for the SMTP unpacks the alert, creates the email digest, and sends out the email.

5.2.4.4 Alert PISR IB Adaptor Specification

The Alert PISR IB Adaptor is responsible for subscribing to the IB to receive any new proto-alerts that need to be handled. The Alert PISR IB Adaptor will also publish alerts that have been processed by the Alert Knowledge Reasoner to the IB for it to deliver those alerts to the appropriate dissemination component. The Alert PISR IB Adaptor is subscribed to additional state notifications that the various dissemination components may be able to provide (in accordance to their registered capabilities). With alert state notification, the AMS can potentially reprocess an Alert in a different way in accordance to some policy set up in the PMS (e.g., send the alert to an SMS server if the email server reports no delivery).

5.2.5 Process and Resource Optimization Management Sub-subsystem (PROMS)

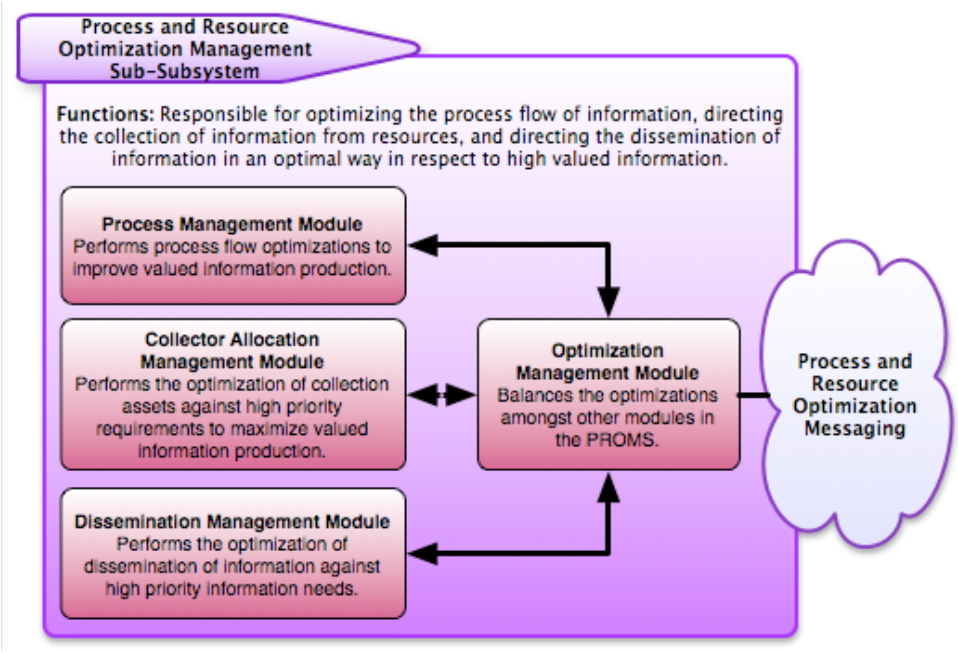

Figure 19. Process and Resource Optimization Management Sub-subsystem architecture diagram

5.2.5.1 Component Description

The Process and Resource Optimization Management Sub-subsystem (PROMS) (Figure 19) is responsible for orchestrating the PISR System in terms of the end-to-end process flow. This includes collection of information from resources, data analysis, and directing the dissemination of information. PROMS orchestrates the which processes should be utilized to deliver the Highest Valued Information (HVInfo) to the right person at the right time. HVInfo is defined through the Information Value of Information Needs against policies defined in PMS. This sub-subsystem is broken into 4 major modules, each with their own set of functionality:

- The Process Management Module (PMM), which handles the development of workflows to produce the near optimal set of valued information in a resource constrained environment.
- The Collector Allocation Management Module (CAMM), which handles the development of collection guidance for information from various sensors in the PISR System, optimizing the collection of perceived high value raw data for analysis.
- The Distribution Management Module (DMM), which handles the development of distribution guidance in order to optimize the flow of information across the PISR System.
- The Optimization Balance Management Module (OBMM), which organizes, coordinates, and balances each of the other modules in regards to one another to make sure they are cooperating in their goals.

The PROMS is not a singular system, but rather the logical grouping of interconnected and dependent products focused on providing the near optimal plans guidance for PISR support of the operator.

5.2.5.2 Key Component Functionality

The key component functionality for the PROMS is separated out into four different modules. The following subsections detail out the respective responsibilities of the PROMS Modules. Guidance communication with subsystems leverages the PISR IB publish/subscribe paradigm. Any new component that is to be involved with the

PISR System is required to either conform to workflow guidance given to the system or have a proxy be able to handle that workflow guidance in its stead. There should be no additional work necessary within the core PISR MCL subsystem to handle any system introduced into the PISR System biosphere.

Process Management Module (PMM)

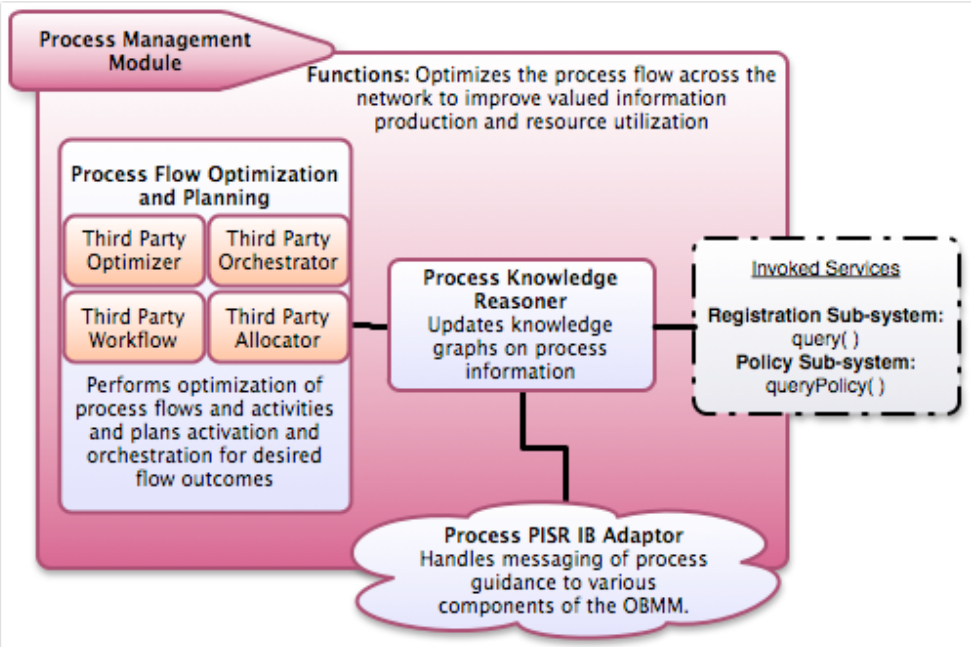

Figure 20. Process Management Module architecture diagram

Component Description

The Process Management Module (PMM) (Figure 20) acts as a high-level process flow controller and optimizer for all top-level processes within the PISR System. It is responsible for defining which activities are executed by which systems or components in response to user and system-defined information needs and processing policies. An activity is a description of a unit of work that can be done that relies on a set of preconditions (usually data requirements) and generates a set of postconditions (usually data produced). PMM is not interested in micromanaging all parts of an activity (which may be processes themselves); it just informs systems what activities need to be accomplished in order to satisfy HVInfo goals. The PMM can inform subsystems of activities to perform in two ways. First, the PMM is allowed to invoke an activity directly through a message placed on the IB. This message simply lists the activity to be accomplished, as predetermined by the activities registered within the RMS. Second, the PMM establishes guidance in the form of enabled activities for a system or component to perform automatically if the right preconditions exist. The second scenario is the primary way information will flow through the system as this is the same thing as guidance for which processes are executed. Sensors can be set up to always kick off the "start" activity of collecting some piece of information, analytics can be turned on or off, and the alert subsystem can focus on certain classes of alerts. For example, a TRSS sensor's proxy server should always have the start activity triggered when some piece of data has been discovered that needs to be forwarded through the IB to the corresponding analytics.

The PMM utilizes PMS to establish which third-party engines will be used for processing data in support of information needs. There are four different types of engines that might be selected from third-party vendors (each described below): Optimization, Orchestration, Workflow, and Allocation. Each engine is responsible for optimizing its

goal against the high-value needs defined by the users of the PISR System. Based on the PISR System needs and available resources, the different optimization engines allow for the reconfiguration of reasoning, workflow, allocation, orchestration, and other process elements to ensure both performance and results. Within the PISR System, a single vendor may have some or all engine categories as a part of their product, so these distinctions may not apply. The PMM consults the OBMM to determine the HVInfo goals balanced against the capabilities of other modules within PROMS. The PMM improves the production of HVInfo and resource utilization through some combination of workflows and allocation to make sure the system as a whole is operating towards some defined optimum. Namely, the PMM can select which workflows get invoked in various scenarios and set that up as guidance for the PISR System.

Key Component Functions

PMM accomplishes its goals through four primary functions:

- PMM continually monitors for changes of information needs and process management policies. As information needs change, the processes to service those needs evolve as well. By monitoring the information needs of a PISR System, current, and projected goals can be planned for.
- PMM manages the various engines required for producing near optimal process plans. This is accomplished through the use of policies stored in the PMS.
- PMM produces process execution guidance for systems and components to utilize which emphasize valued information.
- PMM notifies various components of the PISR System with their process plans to generate valued information though the use of a standard process definition language, such as the Business Process Execution Language (BPEL).

Process Flow Optimization and Planning

The Process Flow and Optimization Planning module is a pluggable interface for various third-party vendors to accomplish different parts of process flow planning. Each of these components could be a standalone component or combined in various combinations. For example, optimization could be accomplished by a vendor's workflow process planning module; in this case, the Optimization Component would exist in part within the workflow component.

Optimization

The Optimization Module controls which optimization techniques are utilized in regard to process management in the PMM. There are three different optimization targets that need to be accounted for when dealing with PMM:

- Workflow – What is the best way to sequence a set of processes to produce HVInfo?
- Allocation – What is the best combination of resources to satisfy processor needs?
- Orchestration – What is the best way to split up a process over several different areas of the PISR System?

While each optimization could be a standalone optimization routine that analyzes outputs from the other engines to validate the output, usually this will be an integrated component within the other engines. Policies within the PMS drive the configuration of each optimization engine selected. Optimization targets the satisfaction of HVInfo goals.

Workflow

The Workflow Module controls how workflows are created towards delivering HVInfo in the PISR System. HVInfo, policies, and registered activities (corresponding to the registered resources in RMS) drive the creation of workflow options to guide the overall system. Given a set of HVInfo goals and the analytical processes that can potentially fulfill those goals, the Workflow Component will develop a plan to satisfy those goals. A workflow should be considered a prioritized plan of attack to satisfy some information needs. Several different workflows may be created to satisfy a plan, with each workflow being prioritized against the HVInfo goals it satisfies. The Workflow Component needs to work with the support of the Allocation Component to determine the availability of processing resources, essentially how many nodes could satisfy a given process requirement. After doing its analysis, the Workflow module comes up with a set of workflow process guidance plans that can get distributed to the components and systems that the Workflow Module is invoking.

Allocation

The Allocation Component controls how resources are allocated in regards to processing subcomponents, or nodes. It determines what processing resources can be allocated for the given requirements and supports the Workflow Component to produce viable workflows. In order to accomplish its purpose, the Allocation Component needs to know the capabilities of various analytical components within its domain and the health of those components. The former is gathered from the RMS and the latter from the HMS. Through this information the Allocation Component knows where processes can be performed and can help drive the Workflow Component to generate doable workflows in terms of allocation of processing resources. For example, there are 20 nodes in the PISR System that can perform a particular process. Currently 10 of them are being overtaxed, and five of the others are displaying a large amount of latency. The Allocation Component would select one of the remaining five nodes and let the workflow module know that it is available.

Orchestration

The Orchestration Module is responsible for taking near optimal workflows produced by the workflow management and allocating pieces of the workflow amongst several systems and components as needed. It is intended to only be used if a particular workflow needs to be addressed by breaking it up amongst several different logical nodes in a network, or parts of the workflow can be deconflicted against time and resources such that they can be processed in parallel. For example, assume there is a high-level process for analyzing an image. Each node in the network can handle a 1MB image in reasonable time. The image received is 10MB. A workflow has been generated which has a sequence of activities for the 1MB image processing. The orchestrator would be responsible for allocating these activities of the workflow to multiple available resources for parallel execution. In general, the orchestration component is primarily useful for analyzing workflows produced by the Workflow Module and seeing if different processes identified by the Workflow Module can be split up into parallel activities for a more efficient distribution of work across available resources.

Process Service Adaptor Specification

The Process Service Adaptor handles the interaction with external services. The Process Service Adaptor knows how to invoke the required functionality of the invoked services. It also knows how to handle any incoming messages that conform to the service interface. It has a two-way flow, working as a buffer for any influx of commands from external sources as well as processing any request to work with external interface requirements.

Process Knowledge Reasoner Specification

The Process Knowledge Reasoner is a processor to manage the current state of the world for each of the process flow optimization and planning components to work with. It is responsible for taking information from the Process Service Adaptor and the Process PISR IB Adaptor and creating a consistent information base for each engine to utilize. The Process Knowledge Reasoner keeps the state of the world in slices of time, allowing an optimization routine to look at any slice up to the current time to support planning the near optimal process in terms of evolving HVInfo goals.

Process PISR IB Adaptor Specification

The Process PISR IB Adaptor is intended to work with the PISR IB to notify the PMM of new HVInfo. It is intended to work as a state blackboard that the PMM can work with. The PMM does process planning continuously, working on a constant state of the world as it does its process planning and forecasting. The Process PISR IB Adaptor is intended to work as a buffer that stores all changes to the world as the PMM is creating a new plan. Once a planning cycle is finished, it is responsible for telling the Process Knowledge Reasoner about the changes in the understanding of the world via the buffered changes.

Collector Allocation Management Module (CAMM)

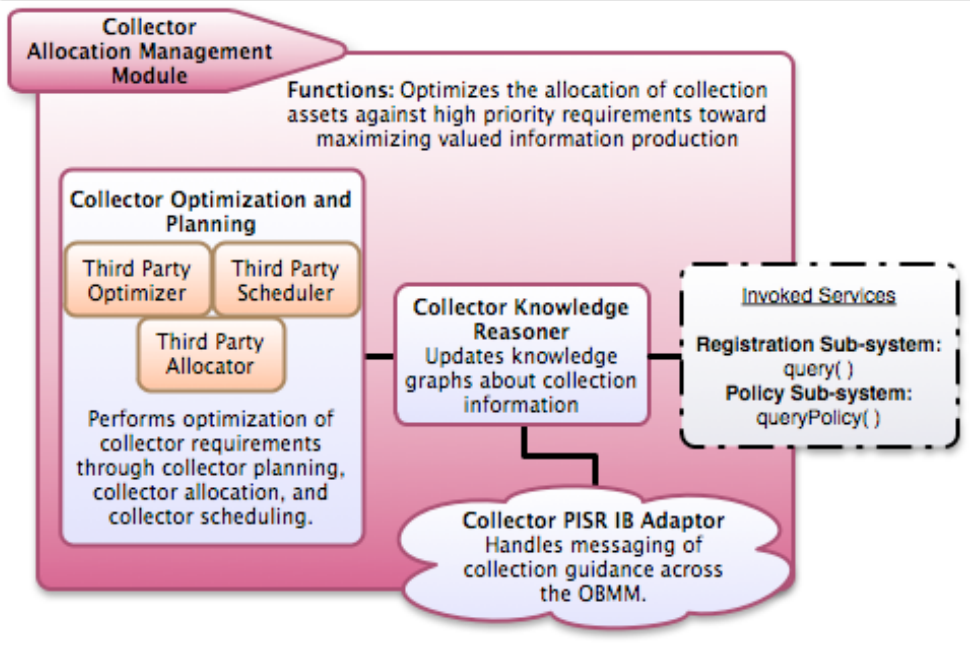

Figure 21. Collector Allocation Management Module architecture diagram

Component Description

The Collector Allocation Management Module (CAMM) (Figure 21) collaborates with the PMM to optimize collection of HVInfo through registered resources taking into consideration network constraints. Through the RMS, the CAMM knows the location, capabilities, and limitations of various resources within the PISR System. CAMM takes these attributes and the guidance provided by the OBMM to produce an optimized collection plan that satisfies HVInfo goals currently established in the PISR System. Similar to the PMM, it utilizes PMS to select the third-party engines to do Optimization, Scheduling, and Allocation, as described above. As the PMM does, it continually reassesses its collection policy and makes modifications as necessary to keep near a global optimum as defined by the information needs and policies.

Key Component Functions

This system accomplishes its goals through four primary functions:

- CAMM continually monitors information needs and collection allocation management policies. This allows CAMM to determine if changes in the desired collection plan of PISR Systems need to occur to support changing HVInfo goals.
- CAMM manages the various third-party engines required to produce a near optimal collection plan. This is accomplished through the use of policies stored in PMS.
- CAMM produces collection plans for external systems to utilize which emphasize the collection of data towards producing HVInfo.
- CAMM orchestrates internal PISR System components' use of resources to conform to generated collection plans by notifying these systems of collection plan changes.

Collector Optimization and Planning Specification

The *Collector Optimization and Planning* module is intended to be a pluggable interface for various third-party vendors to accomplish different parts of collection management planning. Each of these components could be a standalone component or combined in various combinations. For example, optimization should probably be accomplished by a vendor that does schedule planning. So the Optimization Component would exist in part within the scheduler component.

Optimization

The Optimization Component controls which optimization techniques are utilized in regard to the CAMM. Policies within the PMS drive the configuration of the optimization engine selected. There are two different optimization targets that need to be accounted for when dealing with the CAMM:

- Scheduler – When should resources collect information in response to HVInfo?
- Allocation – What is the best combination of resources to satisfy HVInfo goals?

While each optimization could be a standalone optimization routine that analyzes outputs from the other engines to validate the output, usually this will be an integrated component within the other engines. Optimization should primarily be targeting the satisfaction of HVInfo goals.

Scheduler

The Scheduler Component determines when resources should collect information that maximizes the production of HVInfo while taking into consideration PISR resource constraints. For example, directing a camera to take a picture every 2 seconds rather than every 1 second in order to reduce the bandwidth a resource is utilizing. The Scheduler Component works in tandem with the Allocator Component to know what resources are being utilized to cover a particular area. In the same example, such a scenario could allow for two cameras on separate links to schedule picture-taking at 2-second intervals with a 1 second offset from one another in order to maximize information collected from an area while minimizing the bandwidth resources utilized by any one node.

Allocator

The Allocator Component controls how collection resources are allocated. It determines what collection resources can be allocated for given information requirements. The Allocator Component works in tandem with the Scheduler Component so that the Scheduler Component knows what resources it has available to schedule in the first place. In order to accomplish its purpose, the Allocation Component needs to know the capabilities of various collection resources within its domain as well as the health of those components. The Scheduler Component utilizes information gathered from the RMS and the HMS to discover this information. Once the Scheduler Component knows the situation with its resources, it can task the proper set of collection resources towards the HVInfo goals. For example, there may be a HVInfo goal to know if there are vehicles approaching a particular area of interest. The Allocator Component is responsible for figuring out what healthy sensors are available in that particular area and telling the scheduler to task these sensors in the collection of the HVInfo.

Collector Service Adaptor Specification

This Collector Service Adaptor manages the interaction with external services. The Collector Service Adaptor knows how to invoke the required functionality of the invoked services. It also knows how to handling any incoming messages that conform to the service interface. It supports a two-way flow, buffering commands from external sources as well as processing requests to work with external interface requirements.

Collector Knowledge Reasoner Specification

The Collector Knowledge Reasoner manages the current state of the world for each for the collector optimization and planning components. It is responsible for taking information from the Collector Service Adaptor and the Collector PISR IB Adaptor and creating a consistent information base consisting of collection resources for each collection optimization engine to utilize. The Collector Knowledge Reasoner keeps the state of the world in slices of

time, allowing an optimization routine to look at any slice up to the current time in support planning the near optimal collection plan in terms of evolving HVInfo goals.

Collector PISR IB Adaptor Specification

The Collector PISR IB Adaptor is intended to work with the PISR IB to notify CAMM of new HVInfo through subscriptions. It acts as a state blackboard against which CAMM can reason about collection plans. The CAMM does collection planning continuously; it works on a constant state of the world as it executes an iteration of collection planning and forecasting. The Collector PISR IB Adaptor buffers all changes to the world as the CAMM is creating a new plan. Once a planning cycle is finished, it is responsible for telling the Collector Knowledge Reasoner about the changes in the understanding of the world via the buffered changes.

Dissemination Management Module (DMM)

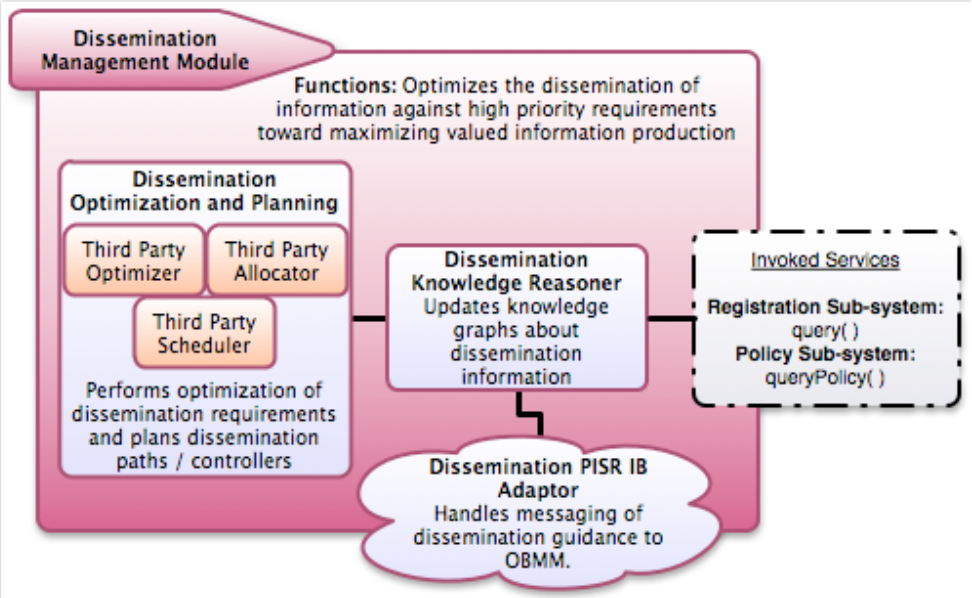

Figure 22. Dissemination Management Module architecture diagram

Component Description

The Dissemination Management Module (DMM) (Figure 22) works with the PMM to optimize the distribution of HVInfo through registered resources against network and other constraints. Through the RMS, it knows the location, capabilities, and limitations of consumers within the PISR System. It takes these attributes and the guidance provided by the OBMM to produce an optimized distribution plan that is suited to optimizing the flow of HVInfo to the consumers of that information on the PISR System. This distribution plan is targeted towards how a piece of information given its value should flow through the system from any node to any other node. Similar to the PMM, it utilizes PMS to select the third-party engines to do Optimization, Scheduling, and Allocation, as described earlier. As the PMM does, it continually reassesses its distribution policy and makes modifications as necessary to keep near a global optimum as defined by information needs and policies.

Key Component Functions

The DMM accomplishes its goals by managing four primary functions:

- DMM continually monitors HVI goals and dissemination management policies. This allows DMM to find potential changes to potentially change current dissemination plans to better satisfy HVI goals.
- DMM manages the various third-party engines to produce near optimal dissemination plans. This is accomplished through policies stored in PMS.
- DMM produces dissemination plans for external systems to utilize which emphasize HVI goals.
- DMM orchestrates internal PISR Subsystems to conform to generated dissemination plans by notifying these systems of the dissemination plan changes.

Dissemination Optimization and Planning Specification

The Dissemination Optimization and Planning module is intended to be a pluggable interface for various third-party vendors to accomplish different parts of dissemination management planning. Each of these components could be a standalone component or combined in various combinations. For example, optimization should probably be accomplished by a vendor that does schedule planning. So the Optimization Component would exist in part within the scheduler component.

Optimization

The Optimization Component controls which optimization techniques are utilized in regard to the DMM. Policies within the PMS drive the configuration of the optimization engine selected. There are two different optimization targets that need to be accounted for when dealing with DMM:

- Scheduler – When should collected information be disseminated in response to its perceived value?
- Allocation – What is the best combination of resources to satisfy the dissemination of information against HVInfo goals?

While each optimization could be a standalone optimization routine that analyzes outputs from the other engines to validate the output, usually this will be an integrated component within the other engines. Optimization primarily should be targeting the satisfaction of HVInfo goals.

Scheduler

The Scheduler Component determines when resources should distribute information in regards to the perceived value of that information against HVInfo goals while taking into consideration PISR resource constraints. Potentially this could be something like throttling the volume of information that is flowing to a particular node because that node is out of theater at the moment and sending information to that node may take away vital bandwidth that blocks delivery of HVInfo to a user in theater.

Allocator

The Allocator Component controls how dissemination resources are allocated in regard to the dissemination of HVInfo. It determines what dissemination resources can be allocated for given information requirements. The Allocator Component works in tandem with the Scheduler Component so that the Scheduler Component knows what resources it has available to schedule. In order to accomplish its purpose, the Allocation Component needs to know the capabilities of various collection resources within its domain as well as the health of those components. It utilizes information gathered from the RMS and the HMS to discover this information. Once the component knows the situation with its resources, it can task the proper set of dissemination resources towards the HVInfo goals. For example, there may be an HVInfo goal to notify users of an IED emplacement. It would be the responsibility of the dissemination allocation engine to make sure that the fastest communication channels from the user's perspective are allocated to send out this information.

Dissemination Service Adaptor Specification

The Dissemination Service Adaptor handles the interaction with external services. The Dissemination Service Adaptor knows how to invoke the required functionality of the invoked services. It also knows how to handling any incoming messages that conform to the service interface. The Dissemination Service Adaptor has a two-way flow, working as a buffer for any influx of commands from external sources as well as processing any request to work with external interface requirements.

Dissemination Knowledge Reasoner Specification

The Dissemination Knowledge Reasoner is a component that manages the current state of the world for each of the Dissemination Optimization and Planning components to work with. It is responsible for taking information from the dissemination service adaptor and the Dissemination PISR IB Adaptor and creating a consistent information base consisting of dissemination resources for each dissemination third-party engine to utilize. The Dissemination Knowledge Reasoner keeps the state of the world in slices of time, allowing an optimization routine to look at any slice up to the current time to support determining the near optimal dissemination plan in terms of evolving HVInfo goals.

Dissemination PISR IB Adaptor Specification

The Dissemination PISR IB Adaptor works with the PISR IB Subsystem to notify the DMM of new HVInfo through subscriptions. It acts as a state blackboard against which DMM can reason about dissemination plans. The DMM does dissemination planning continuously; it works on a constant state of the world as it executes an iteration of dissemination planning and forecasting. The Dissemination PISR IB Adaptor is intended to work as a buffer that stores all changes to the world as the DMM is creating a new plan. Once a planning cycle is finished, it is responsible for telling the Dissemination Knowledge Reasoner about the changes in the understanding of the world via the buffered changes.

Optimization Balance Management Module (OBMM)

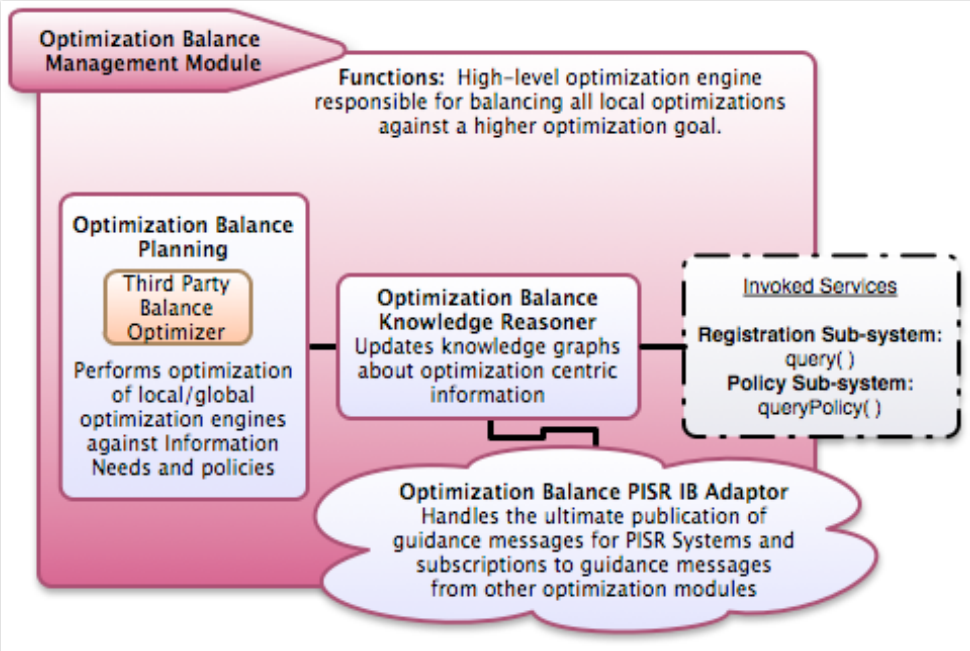

Figure 23. Optimization Balance Management Module architecture diagram

Component Description

The Optimization Balance Management Module (OBMM) (Figure 23) is the high-level optimization management engine that dictates what optimization engine configurations, objective functions, parameters, and constraints are to be used. It interacts with other OBMMs in other MCLs to determine local/global optimization trade-offs. OBMM is primarily responsible for the orchestrating the various other planning components so that there is an

near optimal balance between process, collection, and dissemination planning. It makes sure that all optimization components work in harmony to produce HVInfo. It is also responsible to reconcile global optimization planning with local optimization planning. OBMM accomplishes this by choosing optimization engine configurations ensuring local optimizations are performed within bounds of the global optimization, dynamically modifying them if possible and when necessary.

Key Component Functions

The OBMM has two primary functions:

- OBMM selects an appropriate set of local and global optimization engine configurations, objective functions, parameters, and constraints that can ensure that each optimization engine's output makes sense in terms of the objectives and situation, changing existing strategies if necessary.
- OBMM can specify that the PMM, CAMM, and DMM run multiple local optimization engine solutions with different settings or priorities to pick the near optimal solution set using a set of policy defined performance selection criteria.

Optimization Balance Planning Specification

The Optimization Balance Planning module consists of a single Balance Optimizer third-party component which is responsible for balancing the HVInfo goals amongst the process, collection, and dissemination planning components. The PISR IB adaptor sends information about other optimization goals throughout the PISR to make sure that the local planning optimization takes into consideration global concerns for HVInfo production.

Optimization Balance Service Adaptor Specification

The Optimization Balance Service Adaptor handles the interaction with external services. The Optimization Balance Service Adaptor knows how to handle the required external functionality of the invoked services. It also knows how to handling any incoming messages that conform to the service interface. It has a two-way flow, working as a buffer for any influx of commands from external sources as well as processing any request to work with external interface requirements. Primarily the optimization service handler only interacts with other optimization service handlers and the HMS. MCL external systems should never invoke services provided by the Optimization Balance Management Service.

Optimization Balance Knowledge Reasoner Specification

The Optimization Balance Knowledge Reasoner is a component that manages the current state of the world for each for the optimization planning components to work with. It is responsible for taking information from the Optimization Balance PISR IB Adaptor and creating a consistent information base consisting of optimization resources for the optimization third-party engine to utilize. The Optimization Balance Knowledge Reasoner keeps the state of the world in slices of time, allowing an optimization routine to look at any slice up to the current time in support balancing the various optimization components in terms of evolving HVInfo goals.

Optimization Balance PISR IB Adaptor Specification

The Optimization Balance PISR IB Adaptor is intended to work with the world model to notify the OBMM of new HVInfo. It is intended to work as a current state of the world blackboard that the OBMM can work against. The OBMM does Optimization Component balancing continuously; it should works on a constant state of the world as it does an iteration of balance planning and forecasting. The Optimization Balance PISR IB Adaptor is intended to work as a buffer that stores all changes to the world as the OBMM is verifying that the HVInfo goals of all optimization components are consistent. Once a balancing cycle is finished, it is responsible for telling the optimization knowledge reasoner about the changes in the understanding of the world via the buffered changes.

5.3 Subsystem Interfaces

5.3.1 Interfaces to Subsystems Internal to the PISR System Provided by the MCL Subsystem

Registration Service Interface

- **register** – Registers a resource with the RMS. Resource metadata should be present in the registration of a resource. Resources include sensors, user interfaces, data sources, and data consumers, essentially everything that should be discoverable within the system. Gives the resource a unique id that it can use to deregister itself. This is primarily an internal to MCL function.
- **deregister** – Removes a resource from the RMS. This should only happen if the resource is no longer available within the PISR System. An example would be when a user logs out of the system, the resource associated with the user's desktop is no longer available. This is usually only invoked during a clean shut down of a resource. The RMS utilizes the HMS to discover resources that have been unintentionally removed from the system and puts them in an inactive state. This is primarily an internal to MCL function.
- **subscribe** – Creates a long-standing query for which the RMS will test any new, updated, or newly removed registered resources for a potential match and notifies the subscriber of those changes. Each subscriber receives some unique id associated with the subscription for unsubscribe purposes. For example, a resource doing audio analysis wants to know if any new audio sensors are added to the PISR System. It creates a subscription with the RMS to notify it whenever a new resource with the capability of audio sensor is added to the PISR System. Whenever an audio sensor is added, the subscribing resource is now notified of the new sensor's existence. This is primarily an internal to MCL function.
- **unsubscribe** – Removes the subscription from the system so that the subscriber is no longer notified of new, updated or removed objects matching the subscription. For example, if an information need no longer is looking at a particular area of interest, a component associated with that information need can unsubscribe from notices about new sensors in that area. This is primarily an internal to MCL function.
- **update** – Modifies the metadata associated with a previously registered resource. Fires any subscriptions whose queries match the new attributes. For example, if the location of a registered resource changes, that information is then updated for the RM. Any resources interested in that registered resource (through subscriptions) are then notified that the resource has changed. This is primarily an internal to MCL function.
- **query** – Queries the RMS for registered resources that match specified capabilities at the current moment in time. Inactive systems are ignored unless explicitly requested in the query. For example, a user queries the RMS for all sensors in a particular location. The RM returns all information about registered sensors in that area.

Health Status Service Interface

- **reportStatus** – Allows a registered resource to report its current status information to the HMS in accordance with the metadata associated with the resource. This could be in response to either a request for status or a periodic status update that a resource is giving. For example, if a sensor gives a periodic heartbeat that it is still functional, the adaptor responsible for this sensor (represented by proxy in the SA) will periodically call this function for that sensor and report that the sensor is still there. This is primarily a function internal to MCL.
- **queryStatus** – Queries the current status of the resources described in the query. Will summarize if possible and if the policies are in place to do that summarization. For example, a network administrator wants to know the status of a database that has been registered in the PISR System. The HMS will collect all information about that database and return that information in response to this query. Another scenario would be where a network admin wants the overall status of all audio sensors. The HMS would collect and collate that information and return the overall status of that request. In general, the HMS always summarizes to the level of the query made to it, for specific data, specific queries must be made.
- **monitorStatus** – Creates a subscription to notify the given interested party of a change in the health of the resources described in the query. This creates a notification feedback loop when the health status on a particular resource changes. For example, a network administrator wants to know whenever a resource fails to give a heartbeat within a timeframe of 4 times the length of time between expected heartbeats. First a policy is created that whenever a heartbeat is not reported for that length of time; it considers the resource dead or

unavailable. The monitorStatus would create a subscription that, for whenever a resource reports dead, the subscribing component is notified. In this case the subscribing component should probably be some component that uses AMS to post alerts in the case of a subscription being fired.

Policy Service Interface

- **createPolicy** – Creates a policy for the PISR System. Each policy defined must state the scope associated with the policy. Policies should conform to a standard policy description language.
- **updatePolicy** – Updates an existing policy for the PISR System. Changes the definition of a policy. For example, a policy that states Kill Bin Laden on identification may need to change to Capture Bin Laden on identification.
- **removePolicy** – Removes an existing policy from the PISR System.
- **queryPolicy** – Returns all policies described by a given query. For example, this would allow all policies that affect a given system to be returned for enforcement of those policies. A query could be narrowed to only return policies that affect a given capability as well.
- **createScope** – Creates a scope. This primarily services Community Policies as global, organizational, and local policy scopes are defined and static. This would allow a user to specify that a new community scope that states every user with the capability of Video Analyst should be included.
- **updateScope** – Updates a scope. Redefines the group of users or organizations that should be included in a scope.
- **removeScope** – Removes a scope. Removes a scope and all policies associated with that scope.
- **queryScope** – Returns all scopes that conform to the given query. For example, some user may want to know of all scopes that have Video Analyst as one of their required capabilities.

Alert Service Interface

- **postAlert** – Internal interface method for the MCL's subscription of a proto-alert to be posted to the AMS for route processing. Any component within the PISR can post a proto-alert message to the PISR IB. Eventually that message will be translated into a format appropriate for this function to do the routing behavior for an alert to be delivered in the appropriate manner(s) to a user. For example, SA needs to notify a user that a particular COI has been triggered. SA posts a proto-alert to the PISR IB. Previously, the AMS Alert Knowledge Reasoner subscribed to proto-alerts in the PISR IB. The PISR IB delivers the proto-alert to the AMS Alert Knowledge Reasoner to be translated and posted via this method for alert routing to take place. AMS determines based on the properties of the proto-alert and policies in place how the message is to be delivered. AMS determines the appropriate routing for the message so that the PISR IB can deliver the alert to the proper External Dissemination Component(s), such as an SMTP server. The alert is delivered to the SMTP server via the PISR IB or another established messaging protocol. The SMTP server then delivers the email(s) to interested users. This is primarily a function internal to MCL.

5.3.2 Interfaces to Systems External to the PISR System

5.3.2.1 Interfaces Provided by the MCL Subsystem to Systems External to the PISR System

The MCL Subsystem makes extensive use of the PISR IB Subsystem to publish and subscribe to messages. Rather than have a dedicated interface, MCL takes the approach that communication should happen via message payloads. As such MCL simply publishes messages to the PISR IB for interested parties to pick up on via their subscriptions.

MCL publishes the following messages to the PISR IB for other components and subsystems to pick up in their subscriptions:

- **Alert Messages** – These are messages that conform to the alert message specification. These are messages for the PISR IB to disseminate to the appropriate dissemination components, such as a SMTP server, COT server, or other target components that provide users with information.

- **Status Request Messages** – These are messages that conform to the status request message specification. These are messages that can target any system to provide some status message to the MCL. Generally status request messages are made only to systems if they are not automatically posting status messages due to some expense involved in calculating the status message.
- **Guidance Messages** – These are messages that conform to the guidance message specification. These are messages that relay some bit of guidance to a component within the PISR System. These messages contain guidance on which processes to run now, which processes to run automatically when data is available to them, what information to disseminate, and finally what information to collect. Guidance messages come out of the PROMS and are the optimized plan for how each component of the PISR System should run.
- **Policy Messages** – These are messages that conform to the policy message specification. Policy messages generally only originate from other MCL subsystems. They are the way for policies to be distributed in a global fashion.

5.3.2.2 External Interfaces Used By the MCL Subsystem

The MCL Subsystem also makes extensive use of the PISR IB Subsystem to publish and subscribe to messages. The MCL Subsystem utilizes the all the exposed internal interfaces to translate messages into a format that the MCL can utilize. While it is possible to directly invoke an interface method, the preferred manner should be to post a message on the PISR IB to be routed to the MCL via the publish/subscribe architecture.

MCL subscribes to the following messages from the PISR IB:

- **Registration Messages** – These are messages that conform to the augmented SensorML specification**.** Every system when attached to the PISR System should post a Registration Message detailing what it is, its location, data requirements, data artifacts produced, activities it can do, and other capabilities. This message is then parsed and is used in the register method of the Registration Service Interface to register the object with the MCL.
- **Status Messages** – These are messages that conform to the health status message specification. Systems should periodically report some set of status messages to the PISR IB to be delivered to the MCL's HMS. The message will be parsed and the reportStatus method of the Health Status Service interface will be utilized to report the status.
- **Proto-alert Messages** – These are messages that conform to the alert message specification. Systems may at any time post an alert to be disseminated to the appropriate parties. This allows the MCL's AMS to pick up on said alerts, route them appropriately, and then disseminate them to the appropriate components that can handle them.
- **Policy Messages** – These are messages that conform to the policy message specification. Policy messages generally only originate from other MCL subsystems. They are the way for policies to be distributed in a global fashion.

5.4 Required User Interfaces

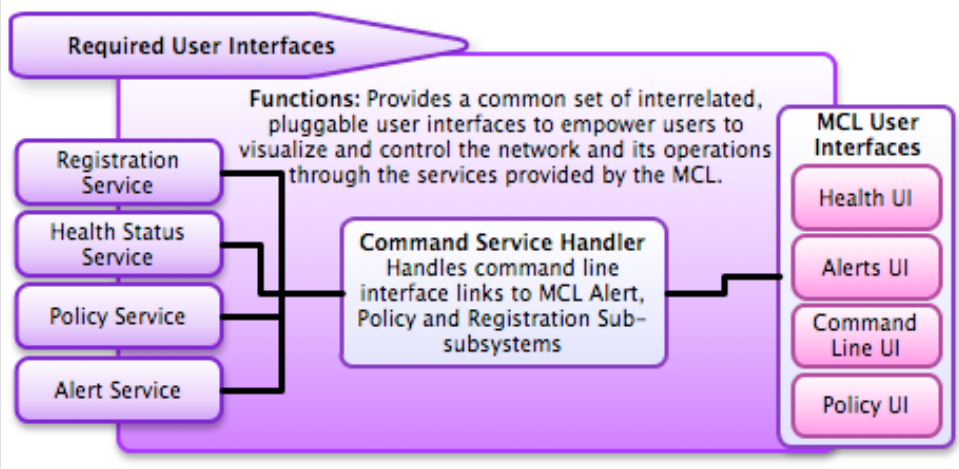

Figure 24. MCL UI architecture diagram

5.4.1 Component Description

The required user interfaces shown in Figure 24 should be developed to allow various users to interact with the functionality of MCL in a common way. Each user interface potentially interacts with different classes of user, but they should have a common look and feel, ideally common to all UIs in the PISR System.

5.4.2 Health UI Specification

The Health UI is intended for PISR administrators and developers. It is a way to view a summarization of the various health aspects of the PISR System. Ideally it should have topological map of the PISR, with the ability to view different types of summarized data in visual form to quickly identify issues that are present in PISR. Health issues include node performance, database performance, sensor health, network latency and throughput, and others. All summarization information should support drilling down into additional details in order to identify particular trouble spots.

5.4.3 Alerts UI Specification

The Alerts UI is intended to be a heads-up display of various noteworthy events in the PISR System. In regard to MCL, these will specifically be health violations and as such should primarily be targeted towards administrators and developers. This interface should be able to sort alerts via any combination of priority, categorization, and time. It should also allow for developers to link back to the systems that generated the alerts, or at least another UI that can give more information about a particular alert. For example, if a health alert comes in this UI should support launching the Health UI and directing the Health UI to the component that generated the alert.

5.4.4 Command Line UI Specification

The Command Line Interface (CLI) is intended for power administrative users to issue a set of commands to various parts of the PISR network through a powerful command line interface. Instead of using the point and click interface, which could be cumbersome to some users, the CLI will allow users to write text commands. Potentially this will also allow for some moderate scripting of tasks as well.

The scripting language for this is yet to be determined, but potentially will be something like Python or Ruby. Each component of a PISR System that wishes to have commands that can be driven by the CLI should register their commands with the RMS as a part of their command control. CLI will use RMS to determine the available commands for a CLI user.

5.4.5 Policy UI Specification

The Policy User Interface (PUI) is intended to be a point and click way to create various scopes and policies that affect the PISR System. The PUI will allow for the creation of new policies through a policy wizard or policy by example interface. The PUI is able to show conflicts of various policies so that a user might be able to resolve them manually. The PUI can display all policies and is able to sort them by system and different attributes that they affect. The PUI could be used by all levels of users; however, the primary users of the PUI will be PISR System administrators and developers.

5.5 Technology Readiness Level

In order to support the PLA, an assessment of the Technology Readiness Level (TRL) needs to be made of products intended to conform to the PLA System architecture. A current product working towards supporting the MCL specification is ActiveEdge produced by Cougaar Software. ActiveEdge is an agent-based distributed information system that performs intelligent orchestration of information, providing various optimizations towards different information goals. While most of the sub-subsystems of MCL are present in ActiveEdge, there is additional work needed for it to incorporate other PISR products into its management scheme. ActiveEdge is the proof of concept for a functioning MCL Subsystem; it is currently at TRL level 6 with respect to the specifications outlined in this document. Below, we describe the TRL of ActiveEdge's MCL implementation with respect to each of the individual sub-subsystems outlined in this section.

5.5.1 HMS

The TRL of the HMS capabilities in ActiveEdge is 6. The system has been tested and demonstrated; however, the system currently only manages health internal to the ActiveEdge product. The health status messaging system for external components to report their current health exists; however, there are currently no systems other than demonstration systems leveraging those health reporting mechanisms. ActiveEdge can report on all metrics that are collected both internally and externally. ActiveEdge can also perform basic analyses and summarization of those metrics.

5.5.2 RMS

The TRL of the RMS capabilities in ActiveEdge is 7. The system has been tested and demonstrated; however, the system currently only registers internal components to the ActiveEdge product. The external interface for registration utilizes the well known and PISR PLA conformant sensor registration architecture called SANY (Sensor Anywhere integrated project; see http://sany-ip.eu). SANY is fielded and currently operates at a TRL of 8. SANY's registration schema, SensorML, is the standard upon which sensors and systems within the PISR will register their capabilities and properties. Due to the utilization of SANY's SensorML as the registration language, ActiveEdge is well positioned to handle subsystem registration outside of MCL.

5.5.3 PMS

The TRL of the PMS capabilities in ActiveEdge is 6. The system has been tested and demonstrated. ActiveEdge conforms to the established standards set in this document for providing policy information across the entire PISR. Policies are leveraged only internally to the ActiveEdge system; for the system to achieve a higher TRL, other PISR subsystems and products need to start leveraging the PMS.

5.5.4 AMS

The TRL of the AMS capabilities in ActiveEdge is 7. The system has been tested and successfully performs in accordance to the specification laid out in this document. It has room to grow by supporting more dissemination platforms. It currently supports external alert notifications through JMS and leverages SMTP, IRC, and CoT dissemination mechanisms, all of which operate at TRL level 8 or 9. The AMS of ActiveEdge is being leveraged by external sources and produces the PISR artifacts described by this document.

5.5.5 PROMS

The TRL of the PROMS capabilities in ActiveEdge is 5. The system is still evolving to match the needs of the PISR environment; however, much of the functionality for orchestrating a system is present. The MCL conforms to the

workflow specifications outlined by the PISR PLA. ActiveEdge can orchestrate the activities of a system; however, the external communication mechanisms described by this document are not implemented yet. All process and resource optimizations currently target and understand only those systems internal to the ActiveEdge product.

This page intentionally left blank.

6 PISR Information Base Subsystem

6.1 Introduction

The PISR Information Base (PISR IB) Subsystem provides for the smart push of actionable information—to those who need it most, when they need it most, and in the form they need it most. This is accomplished through the use of triggers (and other continuously running background queries), defined against a multi-level observation and hypothesis knowledge representation that is capable of ingesting and linking all kinds of relevant information—from sensor feeds and HUMINT, to predictions and plans—while managing believability of different pieces of ingested information.

The PISR IB architecture provides actionable insights from both "slow and fast moving data" as soon as corresponding data sources are ingested by the PISR IB. Figure 25 illustrates how the PISR IB is complementary to the Distributed Common Ground System Marine Corps (DCGS-MC) Integration Backbone (DIB), Generic Hub (GHub), MarineLink and other information servicing data hubs.

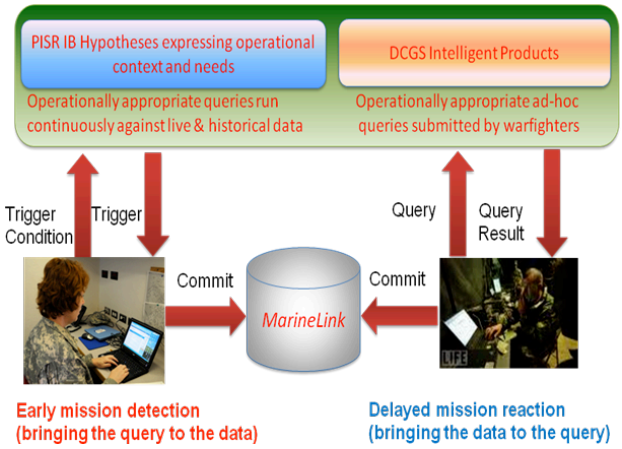

Figure 25. PISR IB supports intelligent delivery of information by integrating data from diverse sources and then pushing it to meet requirements specified by Marines

Near real-time data sources from Sensors/Analytics provide tracks on detected entities, which include observations of persons, vehicles, and facilities. Semantically interoperable Sensors/Analytics are further complemented by semantically integrated historical Marine-relevant data sources (e.g., MarineLink, DIB, GHub, and others). The combination of Analytics and PISR IB subsystems produce actionable intelligence by "connecting the dots," relating information from various sources using inductive and deductive inference.

The PISR IB Subsystem treats other PISR System subsystems as customers. Attending to customers' needs requires the PISR IB to provide PISR subsystems with a customer-friendly information environment. The MCL Subsystem, for dissemination planning purposes, requires PISR IB to have knowledge about the PISR System configuration, network topology, information assurance governance of organizations, organizational roles of users, user access control within organizations, and other data used to ensure information is passed to users needing that information. The MCL Subsystem manages the health of the PISR System (e.g., "Which of its components are working?", "What resources are available?", "What the highest priority information needs are?"). The PISR IB Subsystem provides necessary representation capabilities to understand PISR subsystems, sub-subsystems, components, capacity, constraints, etc. The ability of the PISR PLA to understand the availability, capability, constraints, and limitations of its own components (introspection) is important. Equally important is the ability of PISR PLA system to represent and understand the battlespace within which it operates (situational interpretation). Data in the PISR IB

become valuable when they feed a process that produces good outcomes for specific end-users. For the Marines, that usually means avoiding threats or effectively exploiting opportunities.

The SA Subsystem can focus on specifying relevant analytical routines without worrying about how data is physically represented or stored. This occurs by leveraging the logical interface that PISR IB provides to other components. The PISR IB also enables analytics to extend domain terms by defining new categorical meanings and aliases.

Sensors and analytics are producers and consumers of observations and hypotheses with inherent uncertainties. Modeling a dynamic situation consists of beliefs we have about the true state of affairs rather than objective facts or truths. In that sense, every statement in the PISR IB is an assertion corresponding to a hypothesis, and the PISR IB must enable the determination whether the assertion supports or rejects corresponding hypotheses. The PISR IB associates degrees of believability with all assertions (whether lower-level sensor outputs, seemingly objective human observation, or higher level human- or machine- generated hypotheses) and explicitly links those "believabilities" with their sources. Not only does the PISR IB track multiple sources for the same logical assertion, but over time. Moreover, based on data conflicts and their resolutions, the believabilities of sources may vary. There is no single model of error or uncertainty or degree of belief that works broadly or is universally accepted. The Open Geospatial Consortium SensorML Annex C has a model for detectors which talks about how a simple detector can be characterized (e.g., such as a thermometer). Other detectors (cameras, analytics) would have analogous models but they are not widely available. PISR IB is responsible for providing useful semantic templates for the representation of uncertainty (e.g., spanning spatial dimensions for a type of video-based tracking algorithm). The PISR IB accomplishes this by providing semantic types that track data sources and associated believabilities. These types are specified in the PISR IB schema definition.

Modern battlespaces are messy information environments. Information may be coming in: (1) at different levels of granularity; and (2) with different models of uncertainty. PISR IB must address both of these information management issues. To handle information granularity complexity, the PISR IB breaks with classic Business Intelligence architectures by decoupling the different levels of representation so that each level can directly ingest information. Once ingested, each level of representation checks surrounding levels for the presence of corroborating or conflicting data and takes appropriate actions as a result. Within the PISR IB, high-level information may influence how lower-level information is interpreted. Low-level information (e.g., Abdul Maswary is in Wazir Akbar Khan Mosque) may confirm or reject higher-level hypotheses (e.g., there are terrorists in Kabul). The loosely coupled hierarchies within the PISR IB enable the viewing and comparing of data patterns across various levels. These hierarchies enhance support for other PISR components as well as the PISR IB's core mission of smart push (knowing what information is most important to given users/systems and providing that information with priority to those users/systems) to provide enhanced functionality supporting Marine analysts (e.g., fusing, aggregating, pattern matching, network representation, including social network relationships, navigating, visualization, zooming in/ out).

To handle uncertainty, PISR IB supports the integration of multiple belief management frameworks. For example, there are "likelihood ratio" evaluations to establish a confidence level of inferred hypotheses. This believability management framework makes it easy to look at the likelihood ratio of an hypothesis and its complement (negation), which is the ratio of the probability of seeing the evidence obtained if the hypothesis were true divided by the probability of observing the same evidence if the hypothesis were false. Seeing these two probabilities side by side and their ratio can be useful in recognizing situations where the level of uncertainty could lead to unfortunate incidents, such as firing on innocent civilians.

The principal representational objective of the PISR IB is to record the "state" of the environment, relevant to the Marines, as it is observed and interpreted through the variety of PISR assets and users. Most of the PISR System focuses on the dynamic situation of blue forces, opposing forces, and interrelationships in the battlespace. Data types in asymmetric warfare are diverse, complex, noisy, and poorly formalized. For example, there is no definitive listing of the types of events that should be reported or all the necessary and sufficient data to record about those events. For this reason, events exemplify an open, partially structured, somewhat informal category of importance in our information model. The PISR System must be open to these types of categories, making good use of these data when deemed valuable to current and planned operations. The PISR IB semantically unifies diverse and complex data types and schemas by describing and representing dynamic situations comprising entities of various types, relationships, properties, attributes, and values. Instances of some entity types are static, while others are dynamic because instances and values for those entity types can change over time and space. Given a particular time and space, the values

constitute the situation description at that time and location. At any specific point in space, data may be available about past, present, and future instances of entity types. [23]

Most data values of interest in the PISR PLA have a spatial-temporal context, requiring 3- or 4-dimensional representation. The spatial dimension is characterized by (for example) the ground location plus altitude above or below the surface or sea level. The temporal dimension includes time or time interval when the observation has been collected or when some event apparently occurs, as well as the temporal validity of some piece of information (e.g., the position of an object only until its time of departure). Some of the sensors/analytics operate over dimensions other than spatial and temporal (e.g., hyper-spectral, defining utilized wavelengths organized into band channels and bands). Fusion across sensors/analytics requires using more dimensions (e.g., spatial resolutions), and groupings of sensors/analytics by classes, types, and other characteristics.

6.2 PISR Information Base Subsystem Architecture

The PISR System relies on the PISR IB Subsystem to provide a variety of data management capabilities. PISR IB Subsystem consists of two sub-subsystems to perform semantic data unification for the USMC. First, the Virtual Integration (VI) Sub-subsystem addresses the need for integrating diverse—in source, structure, and content—information. Second, the Distribution Sub-subsystem is responsible for providing publish/subscribe functionality to support distribution of data across the PISR System.

We detail the architecture for the VI Sub-subsystem in Section **Error! Reference source not found.**. This component is named *virtual* because it provides access to concepts of any number of external sources through a single set of concepts model elements and relationships. To accomplish this, the VI Sub-subsystem provides a means of mapping external data sources into a common set of formalized concepts over which computational reasoning can be performed. This includes information from both humans (e.g., from user interfaces) and machines (e.g., from sensors, analytics, and other data stores). For example, sensors may "detect" many people, but it is the previously captured facts regarding social networks and organizational affiliations stored in a database (e.g., DCGS) that determine which people Marines need to know about (e.g., people with known associations with terrorist organizations). It is through this computable logical combination of diverse information content from different sources that the PISR IB supports the delivery of high value information.

The PISR IB Distribution Sub-subsystem is responsible for utilizing interfaces and services provided by external systems to bring requested data into the PISR System. The PISR IB virtual information model integrated with the PISR IB Distribution Sub-subsystem delivers properly transformed, aggregated, and translated data to the warfighter. Removing a need for further transformations at the user side makes PISR IB-distributed data immediately usable.

6.2.1 PISR IB Virtual Information Sub-subsystem

The PISR IB is a *virtual information base*. This means that a user of the PISR system may access any data produced within the PISR system as well as data produced externally by other systems at the enterprise and tactical edge levels that have information relevant to the Marines. Virtualization is accomplished via a logical mapping of vocabularies corresponding to the diverse data sources (i.e., current ISR programs of record, such as DCGS) to the PISR IB conceptual model. Figure 26 depicts integration of diverse data models utilizing vocabulary management.

[23] Future states represent projections or predictions of the state of the battlespace.

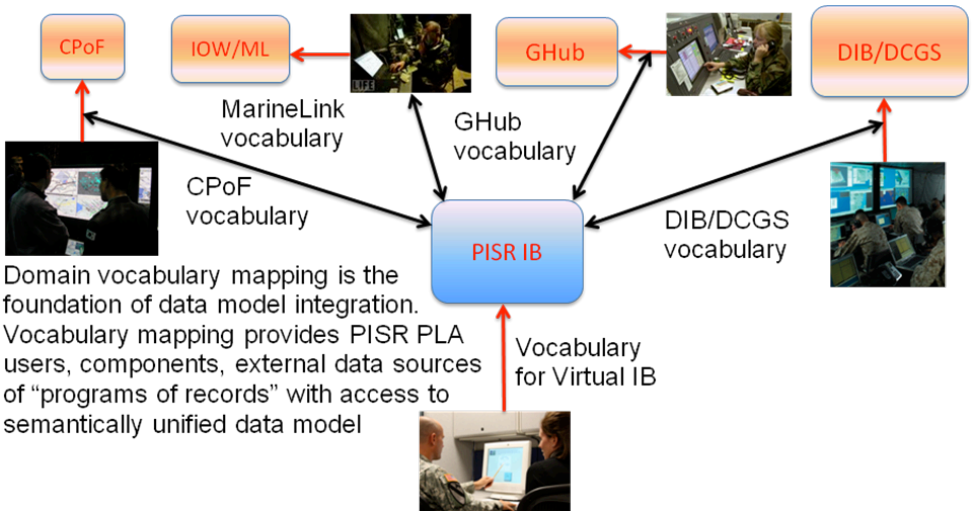

Figure 26. Vocabulary mapping in creation of a virtual information base

6.2.1.1 Operational Needs Serviced by PISR IB Virtual Information Sub-subsystem

The PISR IB capabilities support operational needs grouped into the following three categories to fulfill Quality Attributes of the USMC Intelligence operational community:

- Seamless unification of "just right" data to enhance user navigation in a quest to extract actionable intelligence
  - Requires data virtualization accomplished through semantic harmonization/integration across PISR systems/subsystems and relevant external systems at any tier in the enterprise or tactical edge.
- Facilitating decision making by cueing users to "most critical" situations, which they are capable to attend to
  - Implies "machine-based cueing", requiring a "smart data push" to alert the operators to critical conditions observed/perceived in the battlespace
- Maintain "quality of data products" tailored for the range of operational tempos to fit the operational roles
  - Users at different roles (e.g. Intelligence Officers, Marines on the ground, etc.) need different products at different speeds to do their activities well. Operational context may determine or influence users' needs.

Figure 27 depicts the functional thrusts of this architecture document. Each of the blocks corresponds to one of three operational needs categories listed above. The two top major blocks are further enhanced by the third block. It should be noted that "Near real-time scalable COTS/GOTS Open Architecture" defines the choice of frameworks to support PISR System operational needs. The PISR IB Subsystem is positioned at the intersection of data processing framework and dissemination middleware (or fully fledged dissemination frameworks).

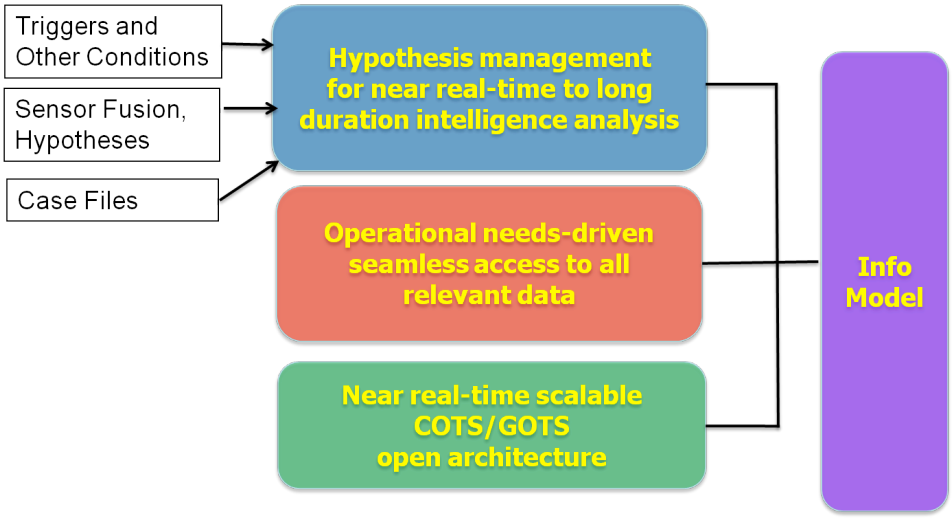

Figure 27. PISR IB architecture servicing operational needs: scalable hypothesis management with unified seamless virtual data integration

The data processing framework supports particular strategies for storage of the PISR IB information model. For instance, one framework could support a classical disk-persisted database. Another data processing framework could support a stream-relational database without a disk persistency. A third hybrid data processing framework could support a stream-relational database with a disk persistency and SQL-based native integration between streaming and relational layers of the database. The PISR IB Subsystem enriches any combination of data processing framework by semantically enabling an information data model supported by the data processing framework. The PISR IB Subsystem is at the intersection of the PISR System data processing framework with dissemination middleware (or fully fledged dissemination framework), enabling the PISR IB Subsystem to enrich dissemination/distribution capabilities for information of greatest value to the users.

6.2.1.2 PISR IB Virtual Information Sub-subsystem Architecture

Figure 28 depicts the functional subcomponents of the PISR IB to be instantiated within the data store, generally as a collection of tables and user-defined functions. The primary goal of the PISR IB is to enable smart push of intelligence based on operational requirements. This goal is enabled through explicit logical coupling of four distinct levels of information. First, *Symbol Definition* enables the mapping of physical vocabularies into specific data *Types* promoting interoperability with external PISR IB Stakeholders. Next, *Type Definition* provides explicit capture of foundational data structures to support the representation of different dimensions of data (e.g., spatial, temporal, political, etc.) and logical comparison. Then, *Schema Definition* provides complex combinations of *type* structures to support operational concepts that will be captured within facts, hypotheses, and observations. Finally, *Schema Instance Storage and Retrieval* supports the capture of and management of data as it is collected and ingested into the PISR IB through the *PISR IB Interfaces*.

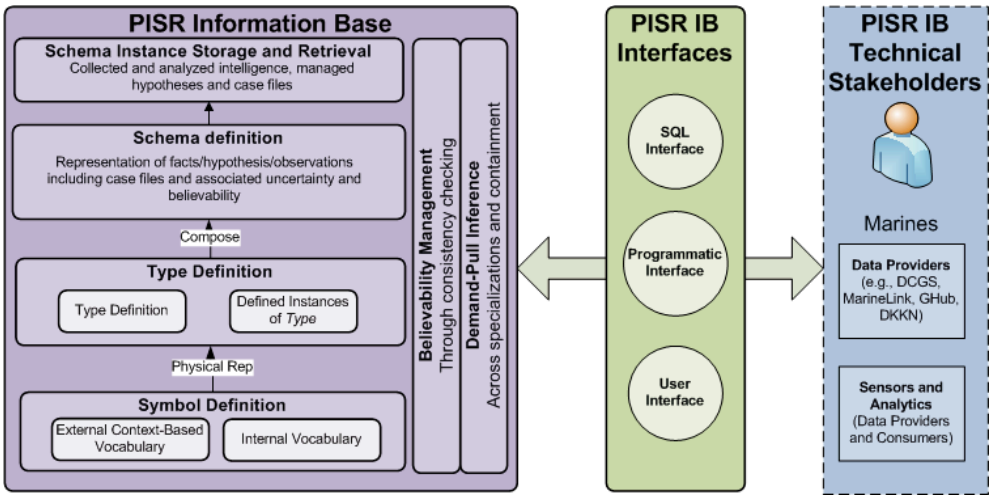

Figure 28. PISR IB functional overview

In addition to the core knowledge representations, PISR IB includes two primary functions: *Believability Management* and *Demand-Pull Inference. Demand-Pull Inference* provides demand-driven inference based on generalization and specialization relationships defined between the concepts defined within the PISR IB. *Believability Management* supports the management of uncertainty externally (by sensors and analytic components) as well as internally (through consistency checking).

6.2.1.2.1 Conceptual Data Model

Low-level information is received from the sensor interpretation subsystem on a frequent basis. We anticipate hundreds or more discovered entities to be incoming during a single analytical period. This assumes that incoming information refers to an entity and contains an observed time-space location. This assumption might change as system understanding, instantiation, and employment evolve.

The low-level information is then compared with information being globally accumulated about the contents of locations in the world; e.g., more stationery and more movable nature; more and less believable; and historical and predicted. Internal to the PISR System, the SA Subsystem is the primary source of such data. Locations of natural terrain features, buildings, and other stationary assets such as roads and bridges are examples of static content. The locations of vehicles, persons, weather are examples of more dynamic information.

Dynamic information may have the form of a feature found for some time range at some location (e.g., an HVI observed at some space-time coordinate) or a motion vector (e.g., a vehicle observed at time *t* to be traveling in direction *d* with or without speed information). Some facts are critical by virtue of what they denote (e.g., two individuals caught discussing the specifics of where to place an IED); others are critical by virtue of their location. Other observations are a function of both. One of the critical derived attributes of a location we are calling "Location Value." The purpose of the Location Value variable (and there are several distinct functions we envisage implementing) is to classify low-level sensor data into two buckets: (1) high information value because it signals a possible triggering, cueing and alerting condition; and (2) less than high information value because it does not signal a possible triggering, cueing and alerting condition.

The underlying concept is that at any point in time, there are many entities being tracked/sensed/observed in a course way and only a small percentage of them are in need of significantly scarcer and more expensive fine-grained sensors/observation/information. So the goal is to implement an efficient process to produce that classification. Subsequent iterations of this design may, if need be, implement a more sophisticated classification logic that takes into consideration such factors as the sensor utilization ratio—when resources are relatively more available, then finer-grained tracking of otherwise less-valued low-level data may be possible (and could be useful in discovering new or low probability events).

Classification criteria are defined and stored in the system. Here is an example of a classification that might be defined and persisted in the PISR IB Subsystem:

> *If a group of entities make a coordinated approach to a high-valued location, it represents a possible ambush threat and is in need of finer-grained information to decide whether it is or is not, in fact, such a threat.*

Determining if a location is or is not high-valued is a function of both the static and dynamic aspects of the location. Say the system determines that an entity is stopped at the side of a minor road—not high enough valued by itself. However, when combined with data from the dynamic world model that includes information about blue forces moving on that same road and projected to pass by the location in question in 90 minutes, the location is calculated to be a high-valued location.

The SA Subsystem provides situational triggers to the PISR IB Subsystem for calculation of particular "triggering, cueing and alerting condition" sub-expressions. One can think of the calculation as inserting a new "hypothesis" row into a table. Each entry in this table represents a distinct possible "triggering, cueing and alerting condition"; for instance, an IED "triggering, cueing and alerting condition". PISR IB schemata will define a distinct table for each kind of possible "triggering, cueing and alerting condition". The source of the possible "triggering, cueing and alerting condition" is the one or more underlying entities from whence it came. PISR IB schemata will allow for the combination of two or more low-level tracked entities, not suspicious in themselves, to constitute a suspicious aggregate entity; e.g., five or more persons crossing a field converging on a possible ambush point.

Diversity of Specialized Things, Actions, and Functions

The PISR IB will provide robust support for a wide variety of distinct kinds of things/actions: entities as well as actions, both concrete (persons, places, things) as well as abstract (e.g., knowledge products). Each distinct kind of thing/event (where by distinct is meant that the non-key attributes of the thing/event are distinct), has a separate collection of table structures. Physically, things and actions of all kinds can be represented by SQL tables. Depending on context, these may be called schemas, frames, or relations.

Generalization/Specialization for Things, Actions, and Functions

Both data and rules are naturally differentiated across multiple levels of abstraction. Whereas the PISR IB enables each separate abstraction level (e.g., person, friendly soldier, marine) to ingest data independently and therefore the possibility that world interpretations may not be consistent across levels of interpretation (e.g., sensors may report no persons present in an area where marines are known to be), consistency is enforced prior to the triggering of any rules. For example, a system-generated alert that might result in the dropping of ordnance on an area would first resolve any potential conflicts regarding presence of persons (or specializations or subtypes of persons) in an area.

The PISR IB supports data entry into the IB at different levels of abstraction (e.g., the output of a sensor may indicate the presence of persons; HUMINT may indicate the presence of "friendlies"). Rules are also abstraction level specific. Some rules may trigger based on the presence of persons, others based on non-hostile persons, and still others based on specifically named marines.

In the PISR IB, generalization-specialization data relationships can be captured through the use of separate tables and foreign keys for each thing/event/schema along the spectrum, connected through a "specialization" table. For example, separate tables could be defined for Marine troops, US troops, Friendly forces, persons and concrete entities. For example:

```
CREATE TABLE wm.concrete_entity
( … );
CREATE TABLE wm.person
( … );
CREATE TABLE wm.friendly
( … );
CREATE TABLE wm.us_military
( … );
CREATE TABLE wm.marine
( … );
CREATE TABLE wm.specialization
(
```

```
…
  concrete_entity        integer references wm.concrete_entity,
  person                 integer references wm.person,
  friendly               integer references wm.friendly,
  us_military            integer references wm.us_military,
  marine                 integer references wm.marine,
…
) ;
```

This can enable the PISR IB to ingest information at whatever level of abstraction at which it comes in. For example, intelligence may indicate the presence of persons in an area adjacent to an intended strike zone. The decision to hold off the strike might be made as a function of whether the persons are hostile or not hostile. Further analysis might show the presence of friendly forces in the region thus barring the use of certain strike methods. So, in this case, all that was needed was to know whether there were persons in an area and if so whether they were friendly.

Mapping data across levels of abstraction (e.g., ingesting a fact about a Marine and linking that fact to facts about friendly forces) is a form of logical inference. There is more than one way to implement this kind of reasoning. Principal approaches include pushing data up the abstraction ladder (e.g., a new fact about a Marine triggers a new fact about a friendly force which in turn triggers a new fact about a person) or pushing rules down the abstraction ladder (e.g., a rule defined in terms of persons also applies to Marines). The optimal way or balance of ways will be a function of the specific patterns of schemas such as fan-out rates and numbers of rules. In either event, the PISR IB can use a generalized need or pull-driven model of inference rather than an open-ended push model which can generate large numbers of useless inferences.

What follows is a representative example of PISR IB schemas in a specialization-generalization hierarchy.

Thing-event

Entities

Concrete

Persons

Friendly Forces

US troops

Marines

Navy

Non-US troops

Civilians

Hostile

Organization

Large stationary produced assets

Fixed sensors

Buildings

Large mobile assets

Armored Vehicles

Tanks

Planes

Small mobile assets

Abstract

Plans

Reports

"triggering, cueing and alerting condition"

Actions

Enemy Attacks

Ambushes

IED explosion

Other

Friendly troop movement

Battles

HVI tracking

Suspicious

Reconnaissance

Container/Contained Relationships for Things, Actions, and Functions

Large things are composed of smaller things. Countries combine to form continents; rooms combine to form buildings. Combining larger numbers of smaller facts about things/actions into smaller numbers of larger facts is a natural part of any interpretation process and is supported by the PISR IB regardless of where the interpretation logic is defined or executed.

But the container/contained relationships are not always clear. For example, two persons are observed in an altercation in a city market. Is one the aggressor and the other the victim? If so, which is which? Could both be insurgents attempting to create a diversion while some other event takes place? Because container/contained relationships are not always known with a high degree of certainty, it is important for the PISR IB to be able to handle uncertainty in the links between smaller things and the roles they play in a larger thing/action.

Additionally, it is not always the case that information is first collected about small things and then rolled up into larger things. For example, HUMINT may hypothesize the presence of a large scale action such as moving enemy insurgents before more detailed sensors can establish, assuming it is true, the composition of that action such as the persons and assets involved.

So the PISR IB will support both probabilistic linkages between smaller things/actions and their roles in larger things/actions and the ability to observe larger things/actions in the macro before identifying the component things/actions of the larger thing/action. It is expected that these linkages will be authored outside the IB through the template and programmatic interfaces.

Relationships of containment/containedness are captured through the use of arrays of foreign keys each of which maps one schema into a component (or container) of a second. For example:

```
CREATE TABLE wm.containment
(
   id                serial primary key,
   …
   concrete_entity              integer references wm.concrete_entity,
```

```
    person                  integer references wm.person,
    friendly                integer references wm.friendly,
    us_military             integer references wm.us_military,
    marine                  integer references wm.marine,
    …
    Action                  integer references wm.action,
    enemy_attack            integer references wm.enemy_attack,
    ambush                  integer references wm.ambush,
    …
) ;

CREATE TABLE wm.contained
(
    …
    concrete_entity         integer references wm.concrete_entity,
    person                  integer references wm.person,
    friendly                integer references wm.friendly,
    us_military             integer references wm.us_military,
    marine                  integer references wm.marine,
   …
    action                  integer references wm.action,
    enemy_attack            integer references wm.enemy_attack,
    ambush                  integer references wm.ambush,
    …
) ;
```

In light of the use cases described earlier in this document, most links will map larger numbers of smaller entities into smaller numbers of larger actions, and there will be entity-entity and action-action links as well. For example, as illustrated below, an action schema "Ambush" might, in its definition, include attributes for time, location, and class/type information for the most likely participating entities in this example; i.e., hostile persons, hostile assets, friendly persons, and friendly assets.

Conceptually, the "Ambush" schema would look like the following:

Ambush

Time

Location

List of contained Thing-event schemas

hostile persons

hostile assets

friendly persons

friendly assets

The core of the "Ambush" schema (ignoring containing links):

- observed_time timestamp(3) without time zone not null,
- start_time timestamp(3) without time zone not null default -infinity,
- end_time timestamp(3) without time zone not null default infinity,
- actors integer[] not null, -- references wm.containment
- objects integer[] not null, -- references wm.containment,
- location geometry

The actors are a list of (thought to be) hostile entities (persons and assets), while the objects of the ambush are the (thought to be) friendly entities (persons and assets).

An observed ambush would include values for core attributes, any contained schemas, and any additional schemas observed. In this example, the "civilians" schema is observed to be contained also in the "Ambush" schema.

Conceptually, the "Ambush" schema in use (when instantiated) might look as follows:

Ambush	
Time	xx/xx: 0800
Location	X-Y
Hostile persons	
	count = 10
	Pashtun count >=1
Hostile assets	
	> 0 Stingers
	2+ building
Friendly troops	
	10 Marines
Friendly assets	
	2 Humvees
civilians	
	count = 20

The PISR IB Subsystem supports inferring the presence of contained things/actions or containing things/actions given ingested information about a thing/action and one or more extant rules of inference, regardless of whether the rule specification and/or execution occurs outside or inside the PISR IB Subsystem itself.

Run Time Semantic Extensibility

Through vetting with stakeholders, the specification of things/actions will be strengthened prior to implementation. It is anticipated, however, that new kinds of things or attributes will arise on a regular enough basis that the PISR IB needs to be able to support the ingestion of information that does not match any existing semantic categories.

The PISR IB will support run time semantic extensibility in the following example-illustrated way. Suppose that the PISR IB has a schema for small mobile assets and that HUMINT is providing data about a new small UAV but that the extant definition for small mobile assets presumed ground-based assets and so had no attributes for altitude.

The PISR IB would recognize the presence of an attribute "altitude" that was not present in the schema for small mobile assets. This failure to find a matching attribute would trigger three events:

1. A secondary table associated with the small mobile asset table would be instantiated that contains the new attribute and its value. Note that this table extension for new attributes is a common element to all IB schemas.
2. A new type would be registered in the type definition space of the PISR IB with a set of possible values consistent with the ingested value.
3. An analyst report would be generated indicating the presence of new attributes. The analyst might then create a new specialization of type small mobile asset, say small UAV.

Consistency Checking

The need to use logical inference to provide consistency checking occurs whenever it possible for the IB to ingest two or more facts or rules which, while not identical as stated, nonetheless have interdependent truth values.

For example, a sensor may indicate the absence or presence of persons at a location. A Blue Force tracking system (e.g., FBCB2) may indicate the absence or presence of Marines at the same location. Logic tells us that if there are zero persons at the location, there also must be zero Marines. However, since the detection of humans is occurring via an independent sensory channel from the detection of Marines, it is possible for sensors to assert that no persons exist at a location while an independent source (e.g., HUMINT) asserts the presence of two Marines. To uncover and highlight these inconsistencies within the PISR IB, there are two basic ways to perform consistency checking: fact propagation and formula propagation.

Using fact propagation, observed facts about a thing/action would propagate to higher or lower abstraction levels. For example, the presence of a Marine would generate the fact that there is a person. The absence of any persons would generate the absence of any Marines. The problem with fact propagation is the risk of generating massive amounts of low value inferences. Instead, the PISR IB can perform background consistency checking using a demand-pull model. Specifically, this means that functions (including *read* calls from external processes) are understood to be the consumers of facts. If for example there is a function that triggers a strike order based on the absence of any Marines in an area, even if no Marines were directly observed in the area, the function would follow the PISR IB's semantic pathways (generalization/specialization, containment/containedness, projection) to test whether there are any logically related facts that might be inconsistent with the given fact and which would impact the execution of the function. If it is then discovered that two persons were observed by a sensor in the relevant location, a potential inconsistency would be discovered (unless the two persons observed were also further identified as being hostile). For another example, if there were a function defined to trigger on the absence of any persons in an area and two Marines are observed, this would also be flagged as an inconsistency.

6.2.1.2.2 Historical, Projected, and Planned World States

Although it is possible to think of the world as a really big thing/event (this being a specialization of the root thing/event rather than either thing/entity or event/action), its importance to the IB is such that it is worth calling out separately. The historical world (all history up until now) is a combination of land, air, and sea models where land includes relatively fixed derived types, both natural (e.g., rivers) and manmade (e.g., buildings), and links to the location-indexed views of "every thing/event" schema (all thing/event schemas generate location-indexed views). Users can see historical world views based on any subsets of locations or thing/events.

Regardless of how or where prediction (or projection) functions are specified or executed, the projected world has all the dimensions/types and schemas of the historical world plus an additional scenario dimension because most thing/events can be projected in multiple ways. As with the historical world, projections are supported at the entity and entity group levels. This is accomplished by mirroring the historical and current implementation in a 'projected' mirror:

```
CREATE TABLE wm.projected_thing_event
(
   id            serial primary key,
   …
) ;

CREATE TABLE wm.projected_extended_thing_event
(
   projected_thing_event_id  integer not null references wm.projected_thing_event,
   …
) ;

CREATE TABLE wm.projected_thing_event_locator
(
   …
   projected_thing_event_id  integer not null references wm.projected_thing_event,
   …
) ;
```

The projected world is intended to represent projections of the current state and trend in affairs; in other words, absent the execution of any endogenous plans. The planned world at any point in time is a set of goal states for the projected world at some relatively future time that differs from the projected values for that same time. In this sense, plans may be thought of as intended deltas to otherwise exogenously projected states.

6.2.1.2.3 Believability Management for Facts/Hypotheses/Observations

All facts are not created equal. Neither are they tense-less or without source attribution. Multiple sources may disagree about a fact. The same source may provide conflicting observations over time. These observations may provide evidence supporting multiple hypotheses. Even when there is no conflict (whether because there is only a single source or all sources agree) there may be significant uncertainty in the facts/observations/hypotheses. Combine this with the fact that different actions require differing degrees of certainty in the underlying observations (e.g., how sure must one be that there are no civilians or friendly forces in an area before ordering an air strike) and there is a need in the PISR IB to support robust belief management that leverages the IB's semantic richness.

While the intention is to support multiple belief-management frameworks, the PISR IB Subsystem will provide for reasonably sophisticated native belief management in the following way.

- Believability (starting off as a 2 digit rank ranging from 0.01 to 0.99 but capable of extension if needed, and also capable of being bucketed with multiple vocabularies) will be able to be associated with a source, a source-time, a source-time-schema, a source-time-schema-attribute or even a source-time-schema-attribute-attribute_value on an as needed basis.
  For example, the believability of data entering the PISR IB from sensors will most likely be sensor-specific whereas the believability of observations coming from HUMINT may also vary by the kind of thing being observed. For example, can a given person reliably distinguish between different kinds of IEDs or munitions. Or, the believability of a specific HUMINT source can vary as a function of whether the observation was made by day or night.
- Every data source (e.g., specific sensors, HUMINT) can be given its own believability function
  For example, the believability of track data entering the PISR IB from sensors may vary from 0.90 to 0.95 depending on the sensor type, its working condition, and the context. The believability of a specific HUMINT source may be equal to 0.99 for direct observations by day (saw person 'x' enter building y at 0900), but only be 0.80 by night.
- Different components of a single thing/action may carry different believabilities
  It may be believable to 0.99 that a certain friendly troop movement is occurring at a particular space-time. And it is also known that the troop movement contains, say, 50 local troops and 25 US forces. The question is what is the likelihood that there is an insurgent hidden amongst the local troops? Depending on the situation, whether each local serviceman is individually identified or whether each is simply an enumerated individual within a group, the PISR IB can represent the belief that any specific local serviceman is actually an insurgent or the belief that there exists at least one insurgent within the group as a whole.
- The believability of any source-observation can change over time based on conflicting beliefs with other source-observations.

The decision tree presented on Figure 29 below is an example method for altering the beliefs of sources based on conflicts with other sources, offered to illustrate and provide an initial tool for developers.

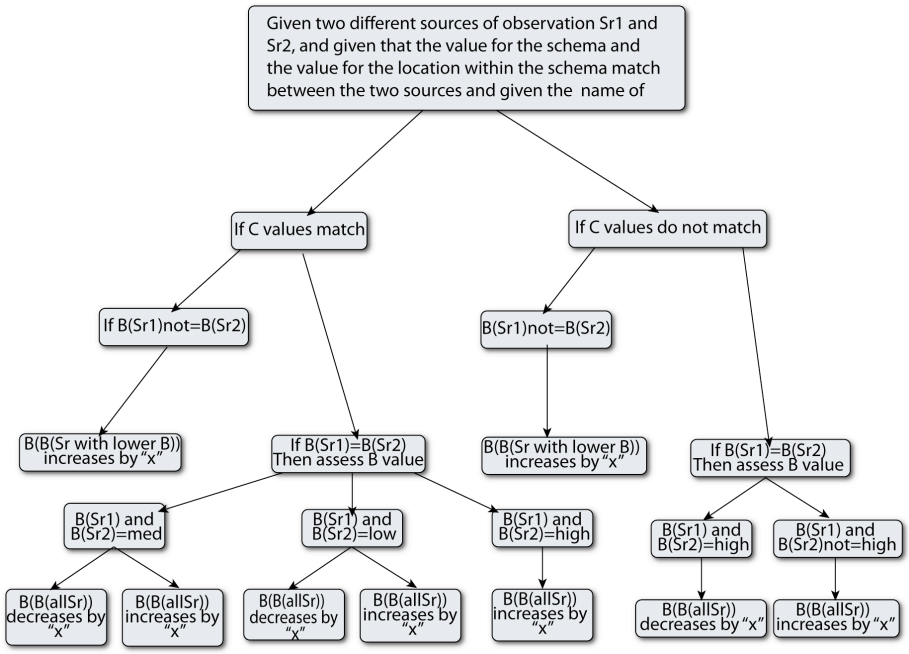

Figure 29. Flexible computational logic for altering the beliefs of sources based on conflicts with other sources

6.2.1.2.4 Decision/Planning Process and Justification Memory

Knowledge products representing intended or taken resource allocation decisions, including plans and new orders within plans, need to include references to the myriad thing/events (and their states) whose observation supports the intended action or decision.

The PISR IB maintains information that can support a basic planning UI in the form of a window on top of historical and projected world states to facilitate users' selection of key entities and actions as supporting evidence for some user-specified action/decision. Such an interface could be used, for example, to easily capture and instantly view all of the accumulated evidence for and against calling a strike on a particular target before it (as an action/decision) is taken and becomes irrevocable, or it could be used ex-post facto to figure as a debriefing tool to help figure out what went wrong or right with a plan/decision.

The PISR IB requires no additional effort to capture and replay the relevant things and events. Users, however, must be willing to record or have recorded their plans and decisions taken.

6.2.1.2.5 Human and Programmatic Interface

In this section, interfaces for PISR IB technical stakeholders are described. It should be noted that the human interfaces defined in this section are specifically targeting *engineering* stakeholders leveraging PISR PLA. Operational stakeholders will access the PISR IB through the interfaces described in the PISR PLA UI Environment Subsystem (Section 2). One notable exception to this is the consistency checking interface which could provide utility to an analyst mining through data in the field.

Adding new type and schema definitions

Although the flexibility of the underlying implementation can be manipulated with SQL, it is advantageous for the PISR PLA to provide other subsystems and performers with a means to alter or expand the existing types, schemas, and functions/rules within the IB. The subsystem provides a template-based user interface that allows a knowledge

engineer to define within the IB new specializations of existing schemas and types (and keeping in mind that all new definitions are specializations of existing definitions). Some new schemas will be new leaves and thus only specializations; others inserted into the hierarchy of schemas will be a specialization of a "parent" type and a generalization of one or more "children" (e.g. inserting a NATO schema in between a friendly forces schema and a collection of Marine, Navy, and Army schemas). Additionally, the knowledge engineer can link the newly defined schemas via contained and containment relationships to other schemas. This interface also allows for the manipulation of the vocabulary entries to support further interoperability. A notional Add Schema screen for a schema design interface is shown in Figure 30 below.

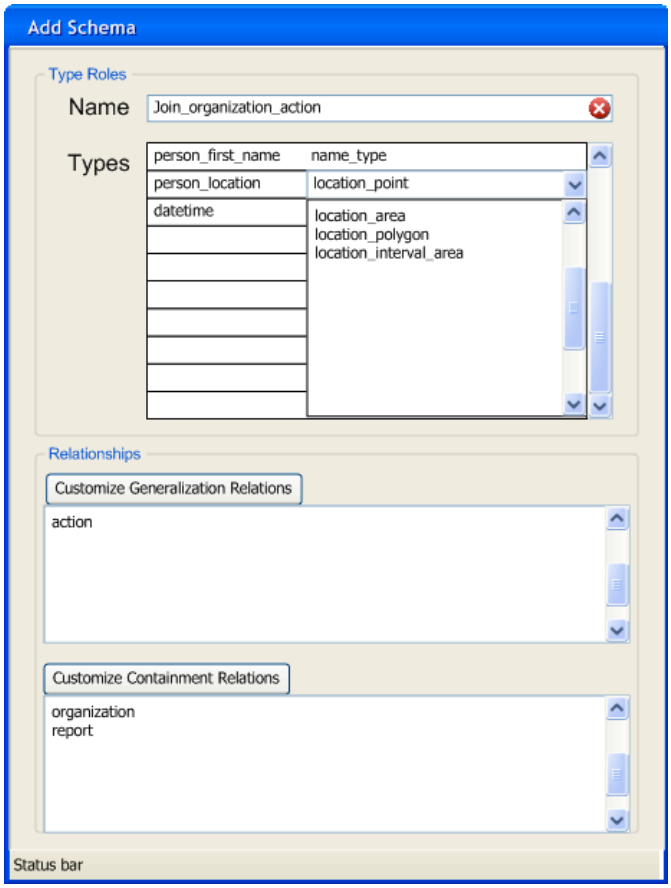

Figure 30. Notional PISR IB editing interface

In this mock-up, a user is defining a new schema within the IB. After determining the schema name (e.g., Join_organization_action), the user must identify the attributes of that schema and select the *Types* that define the concept. Additionally, the user must also specific the *generalization* relationships to other schema and the *containment* relations supporting PISR IB consistency checking and inference.

Consistency Checking

The human interface for consistency checking allows the PISR IB operator to highlight any fact in the IB and query whether there are any other facts that are inconsistent with the highlighted fact. The consistency checking pathways follow the same topology as the PISR IB's built-in semantic structures; namely:

1. Source inconsistency
    a. Query for all conflicting facts attributable to a different source
        i. E.g., two different HUMINT sources disagree about the organizational affiliation of a known local person (note that believability management works to resolve source inconsistencies)
2. Abstraction inconsistency
    a. Query for all conflicting facts at higher or lower levels of abstraction
        i. E.g., if two marines were observed at a location, abstraction inconsistency would occur if an independent observation recorded zero persons at the same location
3. Containment inconsistency
    a. Query for all conflicting facts in contained or containing schemas
        i. E.g., if an IED is found at a location, containment inconsistency would occur if that location is contained within an area defined as secure (i.e., satisfying a "secure location" schema)
4. Projection inconsistency
    a. Query for all conflicting facts that can be projected from existing facts
        i. E.g., if a HVI is asserted not to be at a particular safe location, projection inconsistency would occur if the most recent prior observation of the location and potential travel velocity of the HVI can be projected so that the HVI is currently at the specified safe location

Programmatic PISR IB Interface

The underlying functional logic used to support the PISR IB template interface will be bundled (e.g., in a .jar file) to support programmatic access to PISR IB. This will allow the other subsystems to access PISR IB functionality rather than solely through SQL, although full SQL query processing functionality will be maintained. Functions supported by the programmatic interface include:

- Add/update fact/observation/hypotheses
- Add/update types
- Add/update schemas
- Add/update/associate vocabulary
- Add/update/associate rules
- Generalization / Specialization maintenance
- Containment maintenance
- Consistency checking functions

6.2.1.2.6 PISR IB Examples

In this subsection, we provide example schemas and supporting types to illustrate PISR IB support for dynamic situations. Rather than providing a series of schemas, we present these examples using the following representation:

```
schema_name:(schema1 (generalization), role1 (type||schema), role2
(type||schema),…,roleN (type||schema))
```

In this representation, schema1 is a previously defined schema with which the schema being defined possesses a generalization relationship. The roles defined for the new schema (i.e., role1,…, roleN) can consist of any number of attributed types or any number of contained schemas. Similiarly, instances of schema, in which records are created in the database, use the following representation:

```
schema_instance:(record_id(int), value1(type||schema,
value2(type||schema),…,valueN(type||schema))
```

Also, as described earlier in the chapter, PISR IB supports four explicit linkages between schemas—including generalization, specialization, container, contained—that can be maintained (for example) as arrays of foreign key references in the relational data store. For the purpose of legibility, assume that schema with designated roles within these example schema definitions have explicit container/containment relations defined in the database. Also, assume that the specialization linkages (0…N) are also maintained. All examples presented here represent a given interpretation of data sources, which could be altered or extended to support any number of PISR IB data producers and consumers.

Entities, Attributes, and Values:

The core of the PISR IB is the `Entity` schema and the various specializations (e.g., 'Person', 'Vehicle', 'Equipment'). Each level of specialization provides the attribution of different data types and schemas, which are supported by internal vocabularies. For example, let us consider the person specialization of the concrete_entity schema:

```
entity: (thing_event (generalization), description(String))

concrete_entity:(entity (generalization), description(String), mass(Real))

person:(concrete_entity (generalization),forename (String), surname (String),
gender (Enum), height (Real), age (Integer))

combatant: (person (generalization), gender (Enum), height(Real), weapon
(Equipment), affiliation (Organization))
```

Any number of attributes can be added to this description of person to support the functionality required by PISR consumers and producers.

Actions and Events:

Another core element of the PISR IB as architected is the `Action` schema. Coupled with `Entity`, external software interfacing to the PISR IB can manage information pertaining to entities, their actions, and other entities that receive the action. With other representations, such as RDF or Case Frames, the relationship between action and entity is predefined (e.g., subject, predicate, object or entity, attribute, value). Any combination of relations between activity and entity can be captured and recalled with specialization of this `thing-event` and `action` schema. For example, let us consider a specialization of the `maneuver` schema:

```
action:(thing_event (generalization), description (String), actor (entity),
start_time(Time), end_time(Time))

maneuver: (action (generalization), description (String), actors (entity[]),
geometric_characteristics(geometry[]))

reinforcement_maneuver: (maneuver (generalization),actors (entity[]),
start_positions(location[]), reinforced_positions (location[]))

dismounted_reinforcement_maneuver: (reinforcement_maneuver(generalization),
reinforcers (combatants[]), start_positions (location[]),
reinforced_positions(location[]))
```

Through `abstract_entity` containment schemas—such as reports, observations, and plans—assertions regarding planned actions and events can be properly sourced and managed with respect to actual observations being ingested. Consider a generic IPB process in which hypotheses regarding an enemy COA is developed. First, we must support the representation of a COA involving some number of expected actions which possess some notion of actors involved (in the example below, `ordered_actions` is meant to represent a sequential list of actions required to support the COA; more complex manifestations of connected actions are envisioned as the system evolves):

```
coa:(abstract_entity(generalization), description(String),
ordered_actions(actions))

convoy_coa: (coa (generalization), description(String),
maneuver_actions(actions[]))

convoy_ambush_coa: (coa (generalization), description(String),
ordered_actions({positioning_for_convoy_attack, attack}(Action)),
targets(convoy_coa[]))
```

An analyst or analytic algorithms may then establish indicators—with logical constraints—regarding hypotheses surrounding convoy ambush COAs, which would also be represented within the PISR IB; such as:

```
positioning_for_convoy_attack:(maneuver (generalization), "enemy's moving
into attack position around roads", actors(enemy_combatants[]),
start_positions(location[]), end_position(location(road_geometry+100.0))
```

Finally, let us consider a report within a SIGACT that an RPG was fired on a convoy at time `T` and location `L`. The instance of the schema would resemble:

```
Attack:(12345, "rpg attack" , unknown (person),  rpg (weapon_type), T(time),
L(location))
```

Through the constraints defined in the schema definitions, and through deductive logical inference across generalizations and containment, the instance of the attack schema can be inferred as satisfying part of `convoy_ambush_coa`. Specifically, this includes the containment relationship between the `convoy_ambush_coa` and attack, as well as the constraints on `positioning_for_convoy_attack` regarding `road_geometry`.

States, Projected State, Goal States, and Plans:

The maintenance of dynamic, projected, and planned schema instances within the PISR IB supports the creation and maintenance of comparable notions of state. For example:

- Radar generates an 'observation' *containing* a 'tracked-vehicle' instance denoting the position of the sensed vehicle.
- A 'tracked-vehicle attack' capability is defined with an 'attack-type' and an 'effect-radius'.
- An external analytic creates an instance of 'hypothesis' *containing* an array of projected 'tracked-vehicle' instances denoting the state of the vehicle for some projection interval of time moving forward (in this example, the projected intervals may be seconds and minutes, but for more strategic considerations the intervals could be days, weeks, or months).
- A planned route for a convoy is extracted from the MarineLink 3.0 schema and stored within PISR IB as a 'plan' *containing* an array of 'Maneuver' actions by 'Marine-Convoy' group of entities containing spatial information from the route.

```
convoy_coa: (12345, "weekly convoy along route 36",
{3567,3568}(maneuver_action[]),…, start_time(Time)…)

maneuver_action: (3567, "travel from FOB Detroit to FOB Chicago",…,
{waypoints}(Location))
```

Beliefs:

The maintenance of beliefs and inherent uncertainty is another core element of PISR IB. The attribution of believability metrics can be applied to any type of information for a given schema, providing the flexibility for a PISR System analytic, internal PISR IB logic, external data source (e.g., CIDNE or MarineLink reports), or Marines to attribute believability as a function of source and subject matter.

Actors, Capabilities, Responsibilities, Roles, and Duties:

In defining an 'action' specialization, the entities—both the actors and those entities receiving the effect of the action—can be specified through containment. This construct can be used to form a plan/projection perspective to represent the mechanisms to define capabilities (seen as potential actions; duties extend this concept with deontic logical constructs supporting the representation of obligation). For example:

```
attack: (action(generalization),…, attacker (person[]), target (entity[])))

convoy_coa: (coa (generalization),…,fueler(person[]), driver(person[]),
convoy_elements(equipment),…, start_time(Time), end_time(Time))
```

Sensor Capabilities, Observations, and Measures:

Much like the relationship between actor and action within a 'thing-event' schema, a sensor's capabilities and information collection capabilities can be expressed within the PISR IB. Measures are generally represented by types defined within PISR IB (e.g., 'height' can be a measure collected by a soft biometric sensor/analytic, and is defined as a numeric with associated units (meters) in the schema).

Functions, Relations, and Relationships:

The generalization/specialization and container/contained relationships in the core PISR IB architecture provide the ability to connect, through foreign key reference, schemas and types across different hierarchies of concepts. This connectivity enables logical operations within the IB and supports input and output from automated reasoning within PISR Analytics.

6.2.1.3 Near Real-time Scalable COTS/GOTS Open Architecture

The PISR System data architecture is based on semantic integration plus information logic (i.e., linking new facts/observations/hypotheses with entity/action states and user information requirements) and virtualization, which are all the ingredients of the "smart data push" paradigm. Smart data push is not a feature on the same level as its ingredients; rather, it is the final/highest valued capability and thus presupposes semantic integration, information value analytics, and virtualization. Smart data push accomplishes automated cueing of operators based on triggers computing on semantically harmonized and integrated data sources. In the idea of "valued information at the right time," "right time" does not mean "real time". Operational tempo might dictate a need to disseminate "valuable information" in near real-time. Processing of information provided by near real-time sensors and by near real-time analytics requires a data analytics system capable of generating valuable insights in near real-time. Demanding real-time performance while processing historical data (e.g., from MarineLink), which could be a day or more old, does not necessarily require real-time characteristics. Identifying hidden patterns characterizing behavioral changes of the insurgency requires discovering relationships across information that may span significant periods of time. In this case speed is not a primary requirement, but quality of identified patterns is. Additionally, the amount of data generated by sensors and analytics continues to grow exponentially. These combined forces are pushing even the fastest data warehousing technologies beyond the limits of their batch-processing design, with increasingly higher hardware and operational costs just to maintain the same performance.

A hybrid of stream processing and relational database technologies provides necessary scalability by covering the spectrum of needs to process near real-time data, historical data, and a combination of both at any processing speed required to provide valuable insight, from simple triggering to more intricate characterizations of enemy behavior providing a significant advantage in asymmetrical war.

Marines are organized into different hierarchical tiers and different roles at each of the tier. Marines operating at COCs and Marines on the ground require information with different levels of complexity. The rule of the thumb is that simpler information ought to be delivered with minimal latency. More complex information, tilted towards strategic decisions at the tactical edge, tends to be more complex and to some degree, not as time-critical. The PISR PLA COTS/GOTS Open Architecture (PPCGOA) has been architected to operate at different levels of aggregation of the information.

What separates PPCGOA from Complex Event Processing (CEP) or Analytical Event Processing (AEP) architectures is that the former follows Business Intelligence (BI) data modeling principles. PPCGOA is built to keep the state of incrementally updatable materialized views in near real-time. This capability is fundamental to maintain the state of the world model and to react to the changes required within various operational contexts. Near real-time analytics requires availability of any combination of features since anticipating availability of all features at any point and time is just not possible. That need is addressed by supporting a concept of derived materialized views. PPCGOA includes support of "read" and "write" continuous queries providing a scalable ability to archive incoming data at a detailed level, which is a must for the Marines. The other critical feature is the ability to support "joined queries" resulting in elimination of redundancy in subscription requests to support sensors, analytics, users, and external systems via the PISR IB Distribution Sub-subsystem. Figure 31 depicts the PPCGOA.

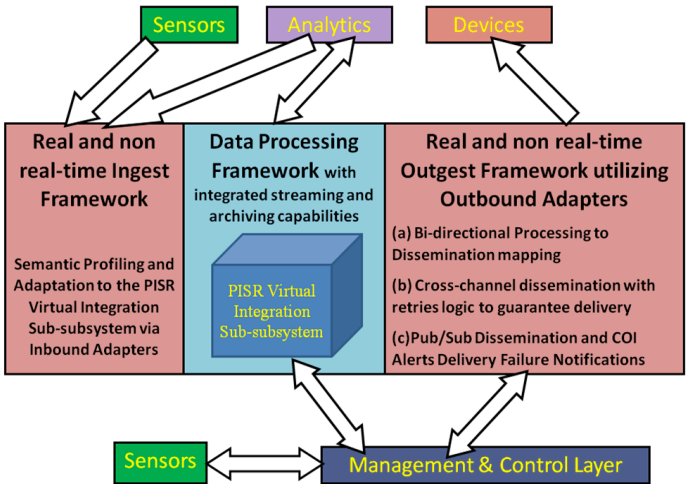

Figure 31. PISR IB near real-time scalable COTS/GOTS open architecture

PPCGOA also supports scalable file data management. This is accomplished by distribution between a high-performance database file management system, internal to PPCGOA, and external file management systems. For example, moving large video files is an operation requiring large data transfers. An efficiency is introduced by moving this bulky data as soon as the warfighter expresses an interest in obtaining the metadata by clicking at a thumbnail. Figure 32 depicts this approach. Bringing streaming into and from the Video Imagery Data Store and into the PISR IB will alleviate inefficiencies found in data copying such large content.

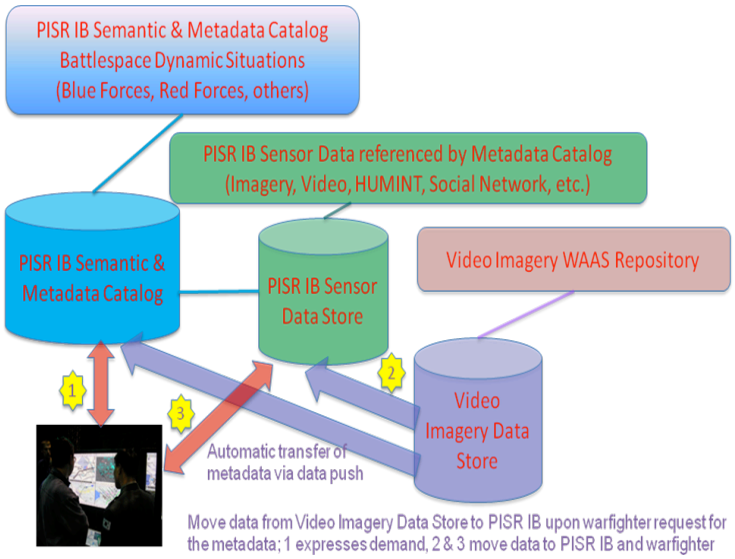

Figure 32. PISR IB storage management strategy

Truviso COTS Framework for Initial Reference Implementation of the PISR PLA COTS/GOTS Open Architecture

An initial reference implementation demonstrating PPCGOA can be created from Truviso's COTS framework[24] (see Figure 33). The Truviso COTS framework provides publish/subscribe capabilities, fundamental for PISR IB to provide a blackboard functionality. Together with MCL-generated "dissemination plan", the Truviso-based PPCGOA is capable of supporting the needs of the PISR IB Distribution Sub-subsystem. The following product descriptions from Truviso illustrate key component capabilities that can be realized for the PISR PLA.

[24] Refer to http://www.truviso.com/continuous-analytics-architecture.php

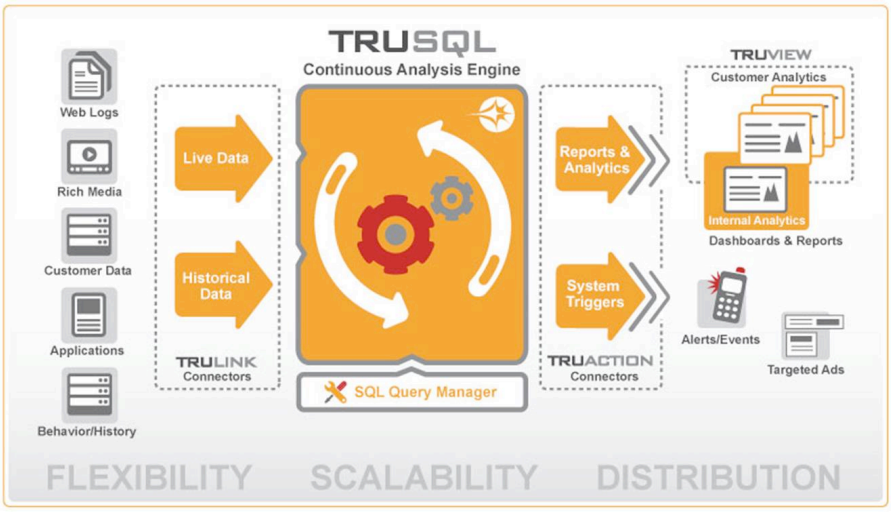

Figure 33. Truviso stream-relational processing framework

TruSQL Engine

The TruSQL flexible high-availability engine is a stream-relational processing technology that combines the real-time speed of stream processing with SQL language queries on stored data. Data is pulled or pushed into the TruSQL engine from relational tables, files, or streaming data sources. The data is correlated and runs against queries that update in real-time, based on changes to the data. There's no need to select data, then read and query the data later. Queries run in parallel as data comes in, resulting in continuously up-to-date reports and a dramatic reduction in server resources and infrastructure costs.

TruAction Triggers

The TruLink Triggers provide a framework to create External Dissemination Components (EDCs) to perform distribution and management of the information via various distribution channels (e.g., sending an email, text message, voice mail, IP-blocking, etc.). TruAction connectors support implementations of "retry logic" over other TruAction connectors. If information cannot be delivered via e-mail, "retry logic" might retry the same channel additional times before switching to any alternate channels while attempting to ensure that valuable information was not only delivered but actually read. TruAction connectors publish query results as alerts or programmatic actions to a source system, and support setting up notifications to be alerted instantly via SMS or email when a parameter or guideline is met, or an exception is found.

TruLink Data Source Connectors

The TruLink Connector framework leverages open standards to enable the system to connect with any data source. Connectors clean, filter, and transform data as needed, and handle the field mapping from schema to schema. Additionally, web services APIs can connect to enterprise software systems, and live data from message busses or extract-transform-load (ETL) tools. A library of pre-built connectors supporting most common data formats is available (including JDBC, XML, JMS, CSV, flat files, SOAP, REST, web services, and more). TruLink Connectors support the full connection lifecycle – start, stop, pause, remove, and add.

Distributed Deployment

Multiple Truviso instances can be deployed across distributed systems, enabling edge, grid, and "in-network" data processing. Clustered servers provide redundancy, high-availability, and distributed processing power. Distributed deployment enables zero-latency, edge data processing at the point of capture, aggregation, filtering, transformation, cleansing across multiple sources, and live monitoring of inter-system business processes and complex events. In a grid deployment, Truviso executes the most complicated analysis instantly as new data arrives from one or more sources, and then moves the aggregates and computed data to central management nodes for rollup and comprehensive reporting.

6.2.2 PISR IB Distribution Sub-subsystem

6.2.2.1 Registration of PISR IB Distribution Sub-subsystem with MCL

Registration Service Interfaces (RSI) require all resources (i.e., sensors, user interfaces, data sources, and data consumers; essentially everything that should be discoverable within the system) to register in accordance with six RSI operations: deregister, register, subscribe, unsubscribe, update, and query.

PISR IB Distribution will register the following resources with the MCL RMS via RSI:

I. PISR IB Subsystem and PISR IB Distribution Sub-subsystem.
II. MarineLink 3.0 over any supported type of the interface, including Web Services

Add/Delete/Get/Update Files
Create/Delete/Get/Update Associations
Get/Update Properties
Get AssociationTemplates
Create/Delete/GetAll/Get/Update AssociationTypes
Create/Delete/Get/Update Entities
Get EntityCounts
Get EntityResults
Get EntityTemplates
Get/GetAll EntitySubtypes
Create/Delete/Get/GetAll/Update EntityTypes
Create/Delete/Get/GetAll/Update PropertyGroupTypes
Create/Delete/Get/GetAll/Update PropertyTypes
Create/Delete/Get/GetAll/Update PropertyTypesFormats
Create/Delete/Get/Unlink/Update RelationalPropertyGroups
Create/Delete/Get/GetAll/Link/Update RelationalPropertyGroupTypes
Get/GetAll RelationalPropertyGroupSubtypes
Get RelationalPropertyGroupCounts
Get RelationalPropertyGroupResults
Get RelationalPropertyGroupTemplates
Get/Update RelationalProperties
Create/Delete/Get/GetAll/Update RelationalPropertyTypes
Get RelationalPropertyGroupTemplates
GetAll RelationalPropertyGroupSubtypeDefinitions
GetAll EntitySubtypeDefinitions
Create/Delete/Get/Update SubtypeDefinitions
Create/Delete/Get/GetAll/Update UserAccounts
GetAssociatedEntities
Logoff Logon
Register/Unregister Application

NOTE:
Interfaces with "A*dd*", "C*reate*", "*Logon*", *Register*" prefixes register with RMS via **register** RSI.
Interfaces with "*Delete*", "*Logoff*", *Unregister*" prefixes register with RMS via **unregister** RSI.
Interfaces with "*Update*", "*Link*", "*Unlink*" prefixes register with RMS via **update** RSI.
Interfaces with "*Get*", "*GetAll*" prefixes register with RMS via **query** RSI.

III. GHub interfaces over any supported types of the interface, including XML, Web Services

getAllChildren
getAllChildrenByID
getChildFolders
getChildDatasets
create/delete Folder
createFolders
delete/add Dataset
create/delete FolderByID
delete/add DatasetByID
getFolderInfo

getDatasetInfo
getFolderInfoByID
getDatasetInfoByID
get/set FolderPermissions
get/set DatasetPermissions
get/set FolderPermissionsByID
get/set DatasetPermissionsByID
get/add FolderComments
get/add DatasetComments
get/add FolderCommentsByID
get/add DatasetCommentsByID
get/set FolderDescription
get/set DatasetDescription
get/set FolderDescriptionByID
get/set DatasetDescriptionByID
addDatasetInit
addDatasetByIDInit
updateDatasetInit
updateDatasetByIDInit
datasetFileUpload
datasetInitFileUpload
datasetGetFileOffset
datasetAppendFileChunk
datasetCommit
getDatasetConfig
get/set FolderRepositoryMetadata
get/set DatasetRepositoryMetadata
get/set FolderRepositoryMetadataByID
get/set DatasetRepositoryMetadataByID
getDatasetFileList
getDatasetFileListByID
getDatasetFileInfoListByDatasetID
getDatasetFileInfoListByDataset
getDatasetFileListByDateRange
getDatasetFileListByDateRangeByID
getDatasetFileContents
getDatasetFileContentsByID
getDatasetFileChunk
getDatasetFileChunkByID
getServiceLinks
getServiceLinksByID
getDownloadLinks
getDownloadLinksByID
getDatasetsByDateAndType
getDatasetsByDateAndTypeByID
getDatasetsByDateAndTypeByIDFull
getDatasetsByDateAndTypeFull
searchForDatasets
searchByBoundingBox
add/remove GHubPathToDataset
add/remove GHubPathToDatasetByID
publish/unpublish Dataset
publish/unpublish DatasetByID
executeSPARQLQueryByID
executeSPARQLQueryByPath
executeVersionedSPARQLQueryByID
executeVersionedSPARQLQueryByPath
getVersionsByID
getVersionsByPath
addFileToDataset
addFileToDatasetByID

NOTE:
Interfaces with "a*dd*", "c*reate*" prefixes, datasetInitFileUpload: register with RMS via **register** RSI.
Interfaces with "delete" prefix register with RMS via **unregister** RSI.
Interfaces with "update", "set" prefixes, datasetAppendFileChunk: register with RMS via **update** RSI.
Interfaces with "get", "search", "execute" prefixes, datasetGetFileOffset: register with RMS via **query** RSI.
Interfaces with "publish" prefix, datasetFileUpload: register with RMS via **subscribe** RSI.
Interfaces with "unpublish" prefix, datasetFileUpload: register with RMS via **unsubscribe** RSI.

IV. DKKN interfaces over any supported type of the interface, including Web Services

addRoute
createInterestFolder
createKeywordQuery
createSpatialQuery
createTypeQuery
expandQuery
publish
publishToAgent
query
queryBOLO
queryFreeText
queryHVI
queryIED
queryInterestFolder
queryObservations
register
registerForInterestFolderUpdates
registerForSubscriptionUpdates
registerMany

NOTE:
Interfaces with "a*dd*", "c*reate*" prefixes register with RMS via **register** RSI.
Interfaces with "delete" prefix register with RMS via **unregister** RSI.
Interfaces with "expand", "set" prefixes register with RMS via **update** RSI.
Interfaces with "query" prefix register with RMS via **query** RSI.
Interfaces with "publish", "register" prefixes register with RMS via **subscribe** RSI.

6.2.2.2 PISR IB Subsystem Information for IA and Support to Dissemination Planning and Execution

Whether specified by a commander, or self-specified, or through default specifications in the application logic, the PISR IB links situational triggers, as defined and as triggered, with those recipients for whom the information has the greatest value. Recipients include persons, manual receptors (i.e., specific communication devices registered to an action or entity function such as battalion commander's radio or pilot's radio), and automated processes.

The value of a "triggering, cueing, and alerting condition" to a recipient is calculated in terms of the comparison between the specific impacts of the "triggering, cueing, and alerting condition" and the relationship (for example, as implemented through foreign keys on schemas) between those impacts and the potential recipients. The value of a "triggering, cueing, and alerting condition" to a potential recipient is thus a function of at least the following:

- The space-time locations associated with the "triggering, cueing, and alerting condition" and the space-time locations of the potential recipients
- The impact of the "triggering, cueing, and alerting condition" on any actions (completed, being executed or in the planning stages) and the relationship between those impacted actions and any potential recipients
- The impact of the "triggering, cueing, and alerting condition" on any non-person things and the relationship between those impacted non-person things and any potential recipients
- The impact of the "triggering, cueing, and alerting condition" on any persons and the relationship between those impacted persons and any organizations to which they might belong and the relationship between those directly impacted persons and indirectly impacted organizations and any potential recipients
- The impact of the "triggering, cueing, and alerting condition" on any knowledge products and the relationship between those impacted knowledge products and any potential recipients

The "partial triggering, cueing, and alerting condition" can impact anything. The locations or thing/events that are directly impacted propagate activation signals outward using the built-in semantic relationships of abstraction, containment, and projection. These signals ultimately reach communication devices where there are either people or processing systems that are ether directly or indirectly impacted by, or are likely to be directly or indirectly impacted by, the perceived occurrence of that condition.

The PISR IB Distribution Sub-subsystem cannot deliver valuable information to the devices without adherence to the IA policies defined at the enterprise level. The type of IA policies relevant to the distribution falls into the practice of User Access Control (UAC). UAC imposes restrictions on the distribution of data to the devices used by the warfighters. To support UAC, the PISR IB needs to support schemas for at least the following artifacts:

a. Organization Hierarchies

b. Users with user profiles, which should include organization, user role, user device, etc.

c. Network topology, including communications networks with their profiles

d. Security enclaves and security guards

Considering that all thing/event schemas generate location-indexed views, UAC will be capable of providing read/write access to different kinds of UAC views corresponding to any chosen UAC-related IA policy. For instance, one IA policy may define access to particular information which is role-based. Another IA policy could further restrict access to the information based on the role and the location of the warfighter device. A third policy might restrict access based on user roles and organization echelons for particular platoons and squads for a group of companies, which, in turn, belong to a group of battalions.

MCL is responsible for the optimization of available resources. As seen earlier, MCL is responsible for the optimization of the dissemination resources and pathways. The properties of dissemination resources are under the control of IA policies. MCL dissemination planning needs to have control of the following resources:

a. PISR IB System configuration, which includes the PISR IB Subsystem and its sub-subsystems.

b. User interfaces, data sources, and external systems (e.g., MarineLink, GHub, DKKN, DIB).

Finally, dissemination planning needs to be able to represent the planning results in the PISR IB through the use of an associated schema and corresponding guidance (constraints, limits, priorities, etc.).

6.3 PISR Information Base Subsystem Interfaces

6.3.1 Interfaces to Subsystems Internal to the PISR System Provided by the PISR IB Subsystem

PISR IB will support a programmatic interface (e.g., a set of Java libraries usable by applications and/or services) that supports the functions underlying the template interface (refer to subsection 6.2.1.2.5 for more details) as part of a common software component.

At the initial stages of the PISR PLA, PISR PLA subsystems (e.g., the SA Subsystem) can use the programmatic interfaces to PISR IB directly to alter its content based on changes occurring at runtime. These functions include:

- Query of existing PISR IB concepts and relations
- Query of PISR IB instance data
- Creation of a new types, schemas, functions, including specialization/generalization and containment/contained
- Addition of PISR IB instance data
- Feedback for operational success and consistency checking

It is anticipated that the evolution of PPCGOA will decouple PISR PLA subsystems from using the PISR IB programmatic interface directly. Decoupling is accomplishable by extending the stream-relational processing framework with PISR IB. Decoupling allows the SA Subsystem and other subsystems to post continuous queries directly or, if necessary, to do so indirectly via the inbound adapters, or by standing up continuous queries after performing necessary transformations.

6.3.2 Interfaces to Systems External to the PISR System

6.3.2.1 Interfaces Provided by the PISR IB Subsystem to Systems External to the PISR System

Refer to subsection 6.2.1.2.5 on human and programmatic interface to the PISR IB Subsystem.

6.3.2.2 External Interfaces Used by the PISR IB Subsystem

Refer to subsection 6.2.2.1 on registration of the PISR IB Distribution Sub-subsystem with MCL.

7 Key Internal Messaging

7.1 Subsystem to Subsystem Messaging

Figure 34 identifies principal message interactions occurring across subsystems of the PISR System. Each of the interactions is described below (presented in alphabetical order of the link labels).

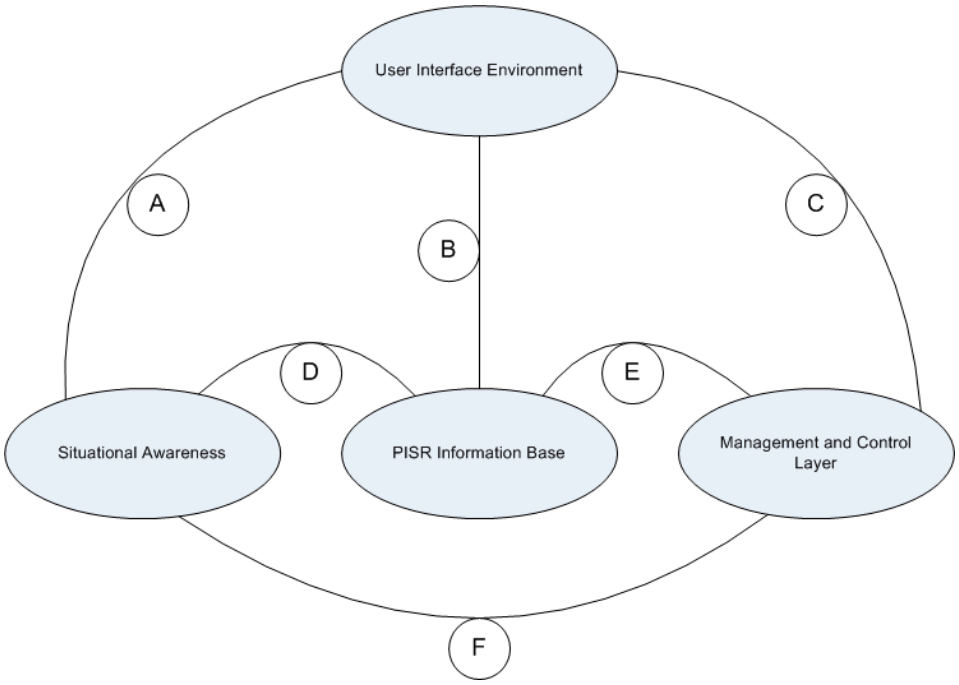

Figure 34. Principal PISR subsystem interactions

7.1.1 Situational Awareness to/from UI Environment (A)

Communication between the SA Subsystem and the UI Environment Subsystem consists of the PISR Information Requests (COIs) and sensor collection planning requests, status, and priorities.

7.1.2 PISR Information Base to/from UI Environment (B)

Communication between the PISR IB Subsystem and the UI Environment Subsystem consists of data base queries in support of Case File Management, Video/Audio/Image stream data from sensors, and Alert messages.

7.1.3 Management and Control Layer to/from UI Environment (C)

Communication between the MCL Subsystem and the UI Environment Subsystem consists of new user registration, sensor configuration, sensor status requests/reports, policy management configuration, alert message configuration, system health status alerts, system configuration, system performance information, and command line commands.

7.1.4 PISR Information Base to/from Situational Awareness (D)

Each piece of information that the SA Subsystem handles is persisted in some form in the PISR IB Subsystem. The SA Subsystem section described these particular pieces of information in greater detail, but consists of detected entities, detected attributes (features), detected behaviors, detected states, detected indicators, and recommended actions. Each of these pieces of data can flow from the PISR IB to the Situational Interpreter or from the Situational Interpreter to the PISR IB.

7.1.5 PISR Information Base to/from Management and Control Layer (E)

There are several messages that go between the PISR IB Subsystem and the MCL Subsystem. Messages from PISR IB to MCL include:

- Registration Information – These are the information messages about different capabilities, location, properties, and attributes of different components within the PISR System. The intended components needing registration within the PISR IB are databases and different computers that are handling the storage of data whether it is real-time or long time.
- Performance Information – These are the variety of messages that deal with the performance mechanisms of the different data stores. These are information such as database usage and other computer resources.

Messages from MCL to PISR IB:

- Policies – These are the policies that need to be used by the MCL to process different optimization, process, collection, and dissemination plans.
- Collection, Dissemination, and Process plans – These are different plans determining performance metrics over the system. These are for archival purposes to potentially analyze how the various plans worked.

7.1.6 Situational Awareness to/from Management and Control Layer (F)

The Situational Interpreter Sub-subsystem and MCL need to communicate with one another constantly. All messages between these systems flow to each other through the Dissemination Subsystem. Messages from the Situational Interpreter to MCL are as follows:

- Registration Information – These are the information messages about different capabilities, location, properties, and attributes about different components within the PISR System. The intended components needing registration within the Situational Interpreter are each sensor component, any Situational Interpreter analytic engine computer, or any other component that should be discoverable by other components.
- COI to Process Mappings – These are processes corresponding to COIs that the MCL uses to optimize what processes are being run when on a global level.
- COI – These are all the COIs that have been generated by the PISR System. These are used for COI optimization prioritization.
- Sensor Diagnostics – These are all the health information messages about each sensor.
- Performance metrics – These are all the metrics of the analytical component computers as well as bandwidth messages.

Messages from the MCL to Situational Interpreter are as follows:

- Process Plans – These are workflows for the Situational Interpreter to use as a plan of execution. Specifically when conflicts arrive, the process plan should be used as the prioritization scheme for execution.
- Collection Plans – These are suggestions for how sensors should be allocated for collecting data.

7.2 Pub/Sub-driven Data Flow for COI Alerting Example

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

7.2.1 Objective

The purpose of Figure 35 is to describe the drivers enforcing the dataflow across PISR System. Dataflow brings directional guidance to the interfaces. The various subsystems need to accept such guidance through guidance interfaces to support "smart push".

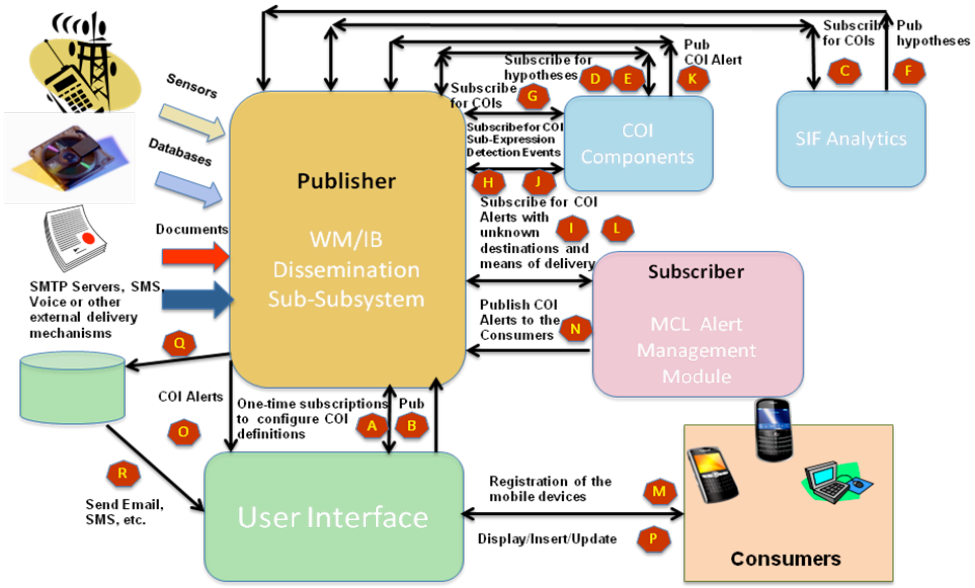

Figure 35. Pub/Sub-driven data flow for COI definition, COI processing, and COI alert dissemination

7.2.2 Step-by-step dataflow

Step A Specify COIs: COI Administrators use "one-time subscriptions" for obtaining information from the PISR IB., which is necessary to create COI specifications.

Step B Publish COI specifications: COI Administrators publishes COI specifications to the PISR IB.

Step C Subscribe to COI specifications: SIF Analytics subscribe to COI specifications, describing COI type-unique interpretation steps for SIF Analytics.

Step D Subscribe to hypotheses: COI Interpreter subscribes to hypotheses generated by CIF Analytics.

Step E Publish hypotheses subscription results: PISR IB publishes COI-produced hypotheses subscriptions results to the COI Interpreter.

Step F Publish hypotheses: SIF Analytics publishes hypotheses to the PISR IB for the COI Interpreter.

Step G Query for COI specification: COI Interpreter issues "COI specifications" subscription to the PISR IB to obtain COI specifications, which are associated with hypotheses published by SIF Analytics. PISR IB publishes COI specification to the PISR IB.

Step H Subscribe to COI sub-expression detection events: COI Interpreter subscribes to the results of the queries it forwards to the COI sub-expression detectors, which run within the PISR IB. COI sub-expression detectors issues standing queries against the PISR IB.

Step I Subscribe to MCL AMS sub-subsystem unprocessed alerts: MCL Alert Management Sub-subsystem subscribes to the PISR IB for COI alerts.

Step J Publish COI sub-expression detection events subscription results: PISR IB publishes COI sub-expression events to the COI interpreter.

Step K Publish COI AMS-sub-subsystem unprocessed Alerts: COI interpreter publishes MCL AMS sub-subsystem unprocessed alerts to the PISR IB.

Step L Publish COI AMS-sub-subsystem unprocessed Alerts subscription results: PISR IB publishes MCL AMS sub-subsystem unprocessed alerts to the MCL AMS.

Step M Discover events of availability of consumer devices: Register mobile devices though UI Subsystem. This activity is generally performed by a device to express a need in COI alerts.

Step N Publish COI Alerts: MCL AMS sub-subsystem publishes processed (with message destination and users addresses) COI alerts to the consumer devices.

Step O Publish COI Alerts: PISR IB publishes COI alerts to the UI subsystem.

Step P Display COI Alerts: UI subsystem's presentation supports streaming paradigm by displaying COI alert on the consumer device window.

Step Q Push Alerts to External Dissemination Devices: The PISR IB publishes processed alerts to the proper dissemination components.

Step R Propagate messages to external devices: These are emails, sms, phone calls, or other external messages corresponding to alerts.

7.3 Pub/Sub-driven data flow for IED Battlefield Activity 1

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

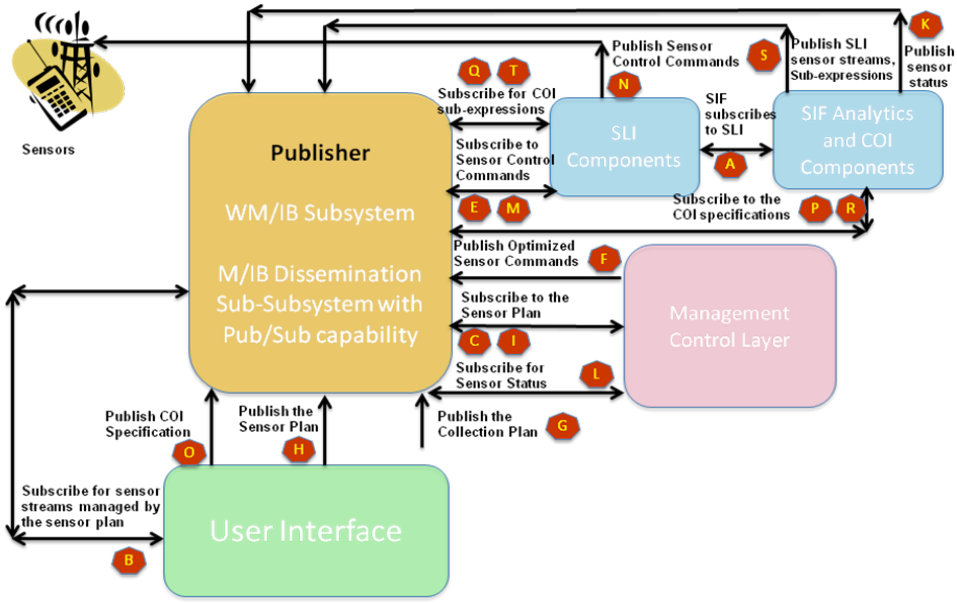

Figure 36. Pub/Sub-driven data flow for IED vignette battlefield activity #1

7.3.1 Objective

The purpose of Figure 36 is to describe the data flow across the PISR PLA Subsystems to perform Battlefield Activity 1 from the IED Use Case Vignette.

1a. Battlefield Activity: BN PISR / collection assets prioritized to support BN focus of effort. PISR assets focused on Named Area of Interest where 1st squad is currently patrolling.

1b. PISR System Activity: Previous to this point in time; an Intelligence Analyst has used the PISR System to build a Sensor Plan, sent that plan forward and the plan was approved. The Sensor Plan caused to GBOSS, WAAS, and Ground Sensors to be deployed to the NAI to provide continuous monitoring required in the Collection Plan. (Reference; Use Case for Case File and COI development) Analyst has entered into the PISR System a COI to alert the analyst if anomalous human activity is detected within the NAI along a route being used by a convoy or patrol.

7.3.2 Subsystem Data Flow and Processing Narrative

The analyst creates the Sensor Plan through the assistance of the User Interface Environment Subsystem. The developed plan (schema instance) is published to the PISR IB. Following approval of the plan (external to the PISR System), the analyst uses the User Interface Environment to access and activate the plan, causing publishing of the plan by the Dissemination Sub-subsystem to the MCL (which has continuous subscriptions for new collection plans). The MCL also has continuous subscriptions to the health status of sensor systems enabling it to compute an optimal allocation of resources to address the requirements of the new sensor plan. The MCL publishes sensor control commands as needed in accordance with the optimization. The SLI components have continuous subscriptions to commands for their respective sensor systems (GBOSS, WAAS, and Ground Sensors) and pass the control commands to the sensors as needed.

The analyst prepares the COI through the assistance of the User Interface Environment Subsystem (COI Editor Sub-subsystem). The completed COI is published to the PISR IB Subsystem, causing the Dissemination Sub-subsystem to publish the COI to the COI Translator Sub-Subsystem which has continuous subscriptions for new COIs. Subscriptions for alerts were also established based on the instructions in the COI publication (e.g., whom to alert and by what means). The COI Translator publishes requests for information (e.g., queries generated to address COI

subexpressions) to the PISR IB Subsystem. Certain SLI sub-subsystems may be subscribed to information requests published by the COI Translator Sub-subsystem. These subscriptions are based on the SLI's capability (based on associated sensor(s) and processing capabilities) to satisfy particular information requests (e.g., an SLI capable of analyzing GBOSS feeds to detect human activity in an area of interest), in which case the Dissemination Sub-subsystem publishes the information requests to those subscribers.

7.3.3 Step-by-step dataflow

Step A Subscribe to sensor streams: SIF Subsystem subscribes to SLI Subsystem for continuous streaming of sensor data with detected entities.

Step B Subscribe to sensor streams with detected entities: UI Subsystem subscribes to the SLI Subsystem for continuous streaming of sensor data with detected entities.

Step C Subscribe to the Sensor Plan: MCL Subsystem subscribes to the Sensor Plan.

Step D Subscribe to the Sensor Status: The MCL Subsystem subscribes for sensor status information.

Step E Subscribe to the Sensor Control Commands: The SLI Sub-subsystem subscribes to sensor control commands for its associated sensor(s).

Step F Publish Optimized Sensor Commands: The MCL Subsystem publishes sensor control commands to the PISR IB Subsystem.

Step G Publish Collection Plan: The MCL Subsystem publishes new Collection Plan to the PISR IB Subsystem.

Step H Publish the Sensor Plan: UI Subsystem publishes Sensor Plan to the PISR IB Subsystem.

Step I Publish the Sensor Plan subscription results: PISR IB Subsystem publishes Sensor Plan to the MCL Subsystem.

Step K Publish the Sensor Status: The SIF Subsystem publishes sensor status information to the PISR IB Subsystem.

Step L Publish the Sensor Status subscription results: The PISR IB Subsystem publishes sensor status information to the MCL Subsystem.

Step M Publish the Sensor Control Commands subscription results: The PISR IB Subsystem publishes sensor control commands to the SLI Subsystem.

Step N Publish the Sensor Control Commands: The SLI component(s) passes sensor control commands to its associated sensor(s).

Step O COI Creation and Publishing: UI Subsystem publishes COI Specification to the PISR IB Subsystem.

Step P Subscribe to the COI Specification: COI component subscribes to COIs.

Step Q Subscribe to the COI Sub-Expression: SLI component(s) subscribe for COI sub-expressions seeking information the SLI sensor(s) and processing can potentially provide.

Step R Publish COI Specification subscription results: The PISR IB Subsystem publishes COI Specification to the COI Component.

Step S Publish Sub-expressions: COI component publishes COI Sub-expressions to the PISR IB Subsystem.

Step T Publish COI Sub-expression results sets: PISR IB Subsystem publishes COI Sub-expressions to the SLI component(s).

7.4 Pub/Sub-driven data flow for IED Battlefield Activities 2 and 3

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

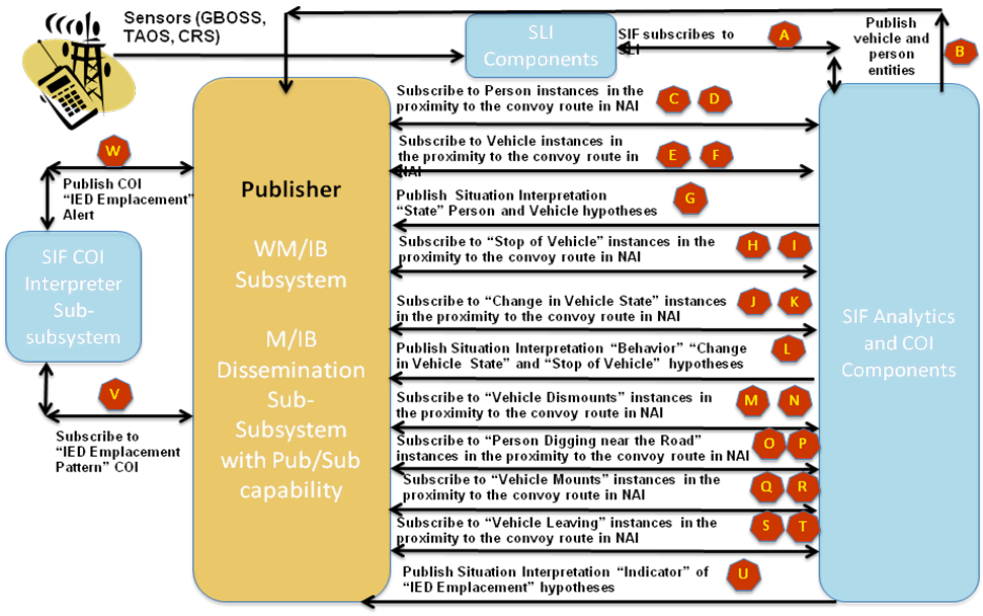

Figure 37. Pub/Sub-driven data flow for IED vignette battlefield activities #2 and #3

7.4.1 Objective

Figure 37 describes the data flow across the PISR PLA Subsystems to perform Battlefield Activities 2 and 3 from the IED Use Case Vignette.

2a. Battlefield Activity: GBOSS, WAAS, WAAS, TRIS, and other sensors continuously monitor anomalous human (foot/mobile) IR signatures "in view of" planned/current BN mission areas and feed info to various SLI Sub-subsystems, including the Tactical Switchboard .

2b. PISR System Activity: PISR System is connected to Tactical Switchboard through the Situational Awareness (SA) Subsystem. GBOSS and WAAS are also connected to the PISR System thorough SA. Other non-real time information feeds like MarineLink and Global Command and Control System (GCCS) are connected through the World Model/Information Base (PISR IB) Subsystem. PISR System is looking for information that will satisfy the COI created in Activity 1 (i.e., anomalous human activity detected within the NAI along a route being used by a convoy or patrol).

3a. Battlefield Activity: A vehicle stops at the side of the road for 20 minutes. Two individuals leave the vehicle, dig on the side of the road and reenter the vehicle. The vehicle leaves the area at a rapid rate of speed.

3b. PISR System Activity: Event Detected; The PISR System identifies the activity in 3a as activity that satisfies the COI created by the Analyst.

7.4.2 Subsystem Data Flow and Processing Narrative

The SLI Sub-subsystems of the SA Subsystem receive data streams from their associated sensor systems (Tactical Switchboard, GBOSS, and WAAS). Low-level data interpretations by the SLIs are published to the PISR IB. Various SIF Sub-subsystem components have subscribed for data interpretations from certain sensors based on the higher-level feature and behavior classification and pattern detection processing they perform. In this case, subscriptions of interest relate to data that could indicate human activity in the specific geographic area defined by some proximity to the convoy route in the NAI. Notifications of data meeting the subscription criteria are sent by the Dissemination Sub-subsystem to the subscribing SIF components.

A sensor provides low-level data interpretation indicating the presence of a vehicle in the area of observation. The associated SLI publishes information about this behavior to the PISR IB. A SIF component that has subscribed for information on vehicle detections is notified. The SIF component is interested in a behavior relating to a vehicle stopping for an extended period of time. The SIF component subscribes to the PISR IB for notifications of a change in the vehicle's state. Another SLI component has subscribed for information relating to an IED emplacement behavior pattern involving a vehicle stopping for a period of time (some threshold), individuals leaving the vehicle, individuals observed digging along the side of a road, individuals reentering a vehicle (could be the same one they arrived in), and the vehicle leaving the area at a high rate of speed (exceeding some threshold).

A sensor (could be different from the one above) provides low-level data interpretation indicating humans exiting a vehicle. The associated SLI publishes the information to PISR IB. The Dissemination Sub-subsystem notifies the SIF component that has subscribed for information relating to the IED emplacement behavior described above.

A sensor (could be different from the ones above) provides low-level data interpretation indicating humans are digging on the side of a road. The associated SLI publishes the information to PISR IB. The Dissemination Sub-subsystem notifies the SIF component that has subscribed for information relating to the IED emplacement behavior described above. The SIF component associates the observed digging behavior to the humans who exited a vehicle based on geographic and temporal proximity and other feature data describing the individuals provided in the published sensor data.

A sensor (could be different from the ones above) provides low-level data interpretation indicating humans entering a vehicle. The associated SLI publishes the information to PISR IB. The Dissemination Sub-subsystem notifies the SIF component that has subscribed for information relating to the IED emplacement behavior described above. The SIF component associates the observed behavior to the humans who had previously exited a vehicle based on geographic and temporal proximity and other feature data describing the individuals provided in the published sensor data (as well as the vehicle feature data provided in the initial published data on the vehicle).

Twenty minutes after the initial vehicle detection, a sensor provides information to its associated SLI that the vehicle is moving. The SLI publishes the information to the PISR IB. The Dissemination Sub-subsystem notifies the SIF component that subscribed for information on the change of state. The SIF component determines the vehicle is the same as the one that has previously been reported as stopped and computes the duration of time the vehicle had been stationary. The SIF component correlates the moving vehicle to the one that had previously stopped, and from which the individuals had exited and reentered. The SIF component publishes satisfaction of all its elements for IED emplacement behavior, together with data on the certainty of each finding and the overall conclusion.

The PISR IB Dissemination Sub-subsystem publishes to the COI Interpreter Sub-subsystem that has subscribed for IED emplacement behaviors in proximity to the particular convoy route in the NAI. The COI Interpreter Sub-subsystem determines that the behavior data and level of certainty in the findings are sufficient to send an alert that the COI has been met. The PISR System then performs the COI alerting steps as described in Figure 35.

7.4.3 Step-by-step dataflow

Step A Subscribe to sensor streams: SIF Subsystem subscribes to SLI Subsystem for continuous streaming of sensor data with vehicle and person schema instances.

Step B Publish sensor streams: SIF Subsystem publishes streams with vehicle and person schema instances to the PISR IB Subsystem.

Step C Subscribe to "person(s) in NAI": SIF Analytics Sub-subsystem subscribes to "person" schema instances in the proximity to the convoy route in the NAI

Step D Publish "person(s) in the NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "person" schema instances in the proximity to the convoy route in the NAI.

Step E Subscribe to "vehicle(s) in NAI": SIF Analytics Sub-subsystem subscribes to "vehicle" schema instances in the proximity to the convoy route in the NAI.

Step F Publish "vehicle(s) in the NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "vehicle" schema instances in the proximity to the convoy route in the NAI.

Step G Publish "state" hypotheses: SIF Subsystem publishes results of interpretation of "vehicle(s)" and "person(s)" schema instances to the PISR IB Subsystem.

Step H Subscribe to "change in vehicle(s) state in NAI": SIF Analytics Sub-subsystem subscribes to "change in vehicle state" schema instances in the proximity to the convoy route in the NAI.

Step I Publish "change in vehicle(s) state in NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "change in vehicle state" schema instances in the proximity to the convoy route in the NAI.

Step J Subscribe to "stops of vehicle(s) in NAI": SIF Analytics Sub-subsystem subscribes to a threshold-based "vehicle(s) stopping for a period of time" schema instances in the proximity to the convoy route in the NAI.

Step K Publish "stops of vehicle(s) in NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "vehicle(s) stopping for a period of time" schema instances in the proximity to the convoy route in the NAI.

Step L Publish "behavior" hypotheses: SIF Subsystem publishes results of interpretation of "change in vehicle state" and "vehicle(s) stopping for a period of time" schema instances to the PISR IB Subsystem.

Step M Subscribe to "vehicle(s) dismounts in NAI": SIF Analytics Sub-subsystem subscribes to "person(s) leaving the vehicle(s)" schema instances in the proximity to the convoy route in the NAI.

Step N Publish "vehicle(s) dismounts in NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "person(s) leaving the vehicle(s)" schema instances in the proximity to the convoy route in the NAI.

Step O Subscribe to "person(s) digging along the side of the road(s) in NAI": SIF Analytics Sub-subsystem subscribes to "person(a) digging holes" schema instances along the side of the road(s) in the proximity to the convoy route in the NAI.

Step P Publish "person(s) digging along the side of the road(s) in NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "person(a) digging holes" schema instances in the proximity to the convoy route in the NAI.

Step Q Subscribe to "vehicle(s) mounts NAI": SIF Analytics Sub-subsystem subscribes to "person(s) entering the vehicle(s)" schema instances in the proximity to the convoy route in the NAI.

Step R Publish "vehicle(s) mounts in NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "person(s) entering the vehicle(s)" schema instances in the proximity to the convoy route in the NAI.

Step S Subscribe to "vehicle(s) leaving NAI": SIF Analytics Sub-subsystem subscribes to "vehicle(s) leaving area" schema instances at a threshold-based high rate of speed in the proximity to the convoy route in the NAI.

Step T Publish "vehicle(s) leaving NAI": PISR IB Subsystem publishes subscription results to the SIF Analytics Sub-subsystem with "vehicle(s) leaving area" schema instances in the proximity to the convoy route in the NAI.

Step U Publish "indicator of IED emplacement" hypotheses: SIF Subsystem publishes results of interpretation of "person(s) leaving the vehicle(s)", "person(a) digging holes", "person(s) entering the vehicle(s)" and "vehicle(s) leaving area" schema instances to the PISR IB Subsystem.

Step V Subscribe to "IED emplacement pattern": COI component subscribes to "IED emplacement behavior pattern" schema instances.

Step W Publish COI Alert": COI component evaluates COI sub-expressions to determine whether COI Alerting is warranted. If condition is "true", COI Alert is published in accordance with Figure 35.

7.5 Pub/Sub-driven data flow for IED Battlefield Activity 4

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

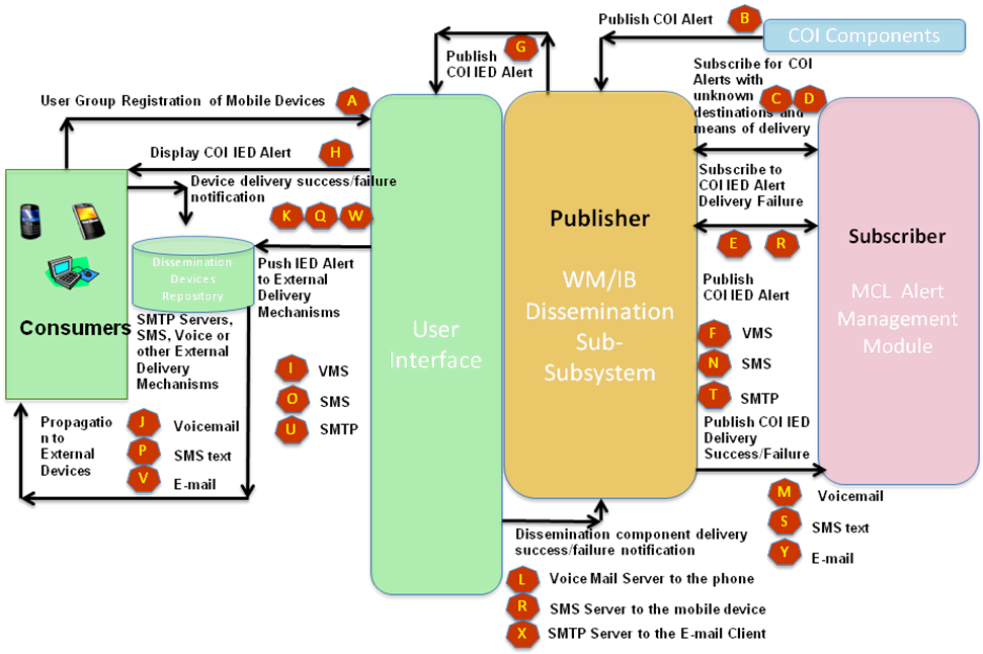

Figure 38. Pub/Sub-driven data flow for IED vignette battlefield activity #4

7.5.1 Objective

The purpose of Figure 38 is to describe the data flow across the PISR PLA Subsystems to perform Battlefield Activity 4 from the IED Use Case Vignette.

4a. Battlefield Activity: BN HQ and below are alerted to IED hazard and "pushed" sensor information to initiate analysis and target tracking.

4b. PISR System Activity: Analyst who created the COI is immediately alerted with a flashing message on his computer monitor that his COI has been satisfied. The COI requested that Battalion Headquarters (BN HQ) and below be Alerted when the COI is satisfied. PISR System prioritizes the Alert so that communication resources are available to send the Alert to BN HQ and all activities below that. COI also requested that a summary report of the sensor information be included with the Alert. PISR System packages a summary of the sensor information with the Alert. Several other COIs that are active requested that their authors be sent any Alerts dealing with IEDs. These users receive the Alert and begin doing additional research to understand and obviate the threat. Standing Operating Procedures (SOPs) for this Battalion require a call list of key personnel be alerted when a suspected IED is discovered. PISR System retrieves this list from PISR IB and uses the Dissemination Sub-Subsystem to phone all the people on the list. If person fails to answer the phone, the PISR System automatically sends a text message and an email to the person. The PISR System continuously monitors communication system to know when all persons on the list have been notified and they have responded to the Alert.

7.5.2 Subsystem Data Flow and Processing Narrative

The analyst who created the COI is sent an alert by the Dissemination Sub-subsystem via the User Interface Environment Subsystem. The COI includes an alert distribution profile specifying to whom and by what means the alert should be distributed; in this case, the BN HQ and subordinate elements were identified as recipients of the alert. AMS identifies the users associated with those particular groups and creates additional duplicate alerts for each of the users in that group in accordance to policies set forth and preferred message delivery style. According to the Alert specification, a summary of sensor information is included with the alert. Such information includes the source and evidentiary trail for the evidence supporting satisfaction of the COI that was published by the COI Interpreter Sub-subsystem when it determined that the COI was satisfied with sufficiently high confidence to warrant publishing to the PISR IB Subsystem. Refer to Figure 35 for more details.

In the satisfaction of the IED threat COI, the COI Interpreter also determined satisfaction of other COIs regarding possible IED threats. PISR users identified in the distribution instructions for these COIs are also alerted. In particular, the distribution instructions on one of the satisfied COIs (regarding discovery of a suspected IED) contains reference to a call list of personnel to be alerted. The MCL Alert Management Sub-subsystem (AMS) retrieves the content of the list from the PISR IB and notifies the PISR IB Dissemination Sub-subsystem to alert the users on the list by phone. The notification includes a protocol for positive acknowledgement of receipt of the alert by each user (part of the distribution instructions entered when the profile for COI Alert per each of the COI types was created). MCL AMS subscribes to the COI Alert Notification Delivery. If a user does not answer the call, the User Interface Environment Subsystem will publish to the PISR IB Subsystem "undeliverable" COI Alert Notification Delivery information. The MCL AMS Sub-subsystem will receive this information as a result of its subscription and will try to alert the person via a text message. If unsuccessful, the MCL AMS Sub-subsystem will try e-mail in accordance with previously set up policies. The MCL AMS Sub-subsystem continues to monitor interactions with the users to ensure all persons on the list have been notified and they have all responded to the alert.

7.5.3 Step-by-step dataflow

Step A Describe COI Alert Delivery Process Flow: User registers through UI Subsystem for COI alerts. User specifies the Distribution Profile defining the sequence of COI Alert Deliveries: Voice Message first, SMS message next, e-mail following SMS, etc. User optionally defines requirement for receipt acknowledgement.

Step B Publish "unprocessed" COI IED Alerts: COI Interpreter Component determined satisfaction of other COIs regarding possible IED threats. COI Interpreter Components publishes COI alerts to the PISR IB Subsystem.

Step C Subscribe to "unprocessed" COI IED Alerts: MCL Alert Management Sub-subsystem subscribes to the PISR IB for COI IED alerts.

Step D Publish "unprocessed" COI IED Alerts: The PISR IB Dissemination Sub-subsystem publishes subscription results unprocessed alerts to the MCL AMS Sub-subsystem.

Step E Subscribe to COI IED Alert Delivery Failure: MCL Alert Management Sub-subsystem subscribes to the PISR IB for COI IED alert delivery failure.

Step F Publish COI IED Alerts: MCL AMS Sub-subsystem publishes processed (with dissemination component, message destination and users addresses) COI IED Alerts to the PISR IB Subsystem. In accordance with the COI IED Alert Distribution Profile, the 1st dissemination component will be the Voice Messaging System (VMS). It is external to the PISR System.

Step G Publish COI IED Alerts: PISR IB publishes COI IED Alerts to the UI subsystem.

Step H Display COI IED Alerts: UI subsystem's presentation layer displays the COI IED Alert to the explicit subscriber to the COI Alert (e.g., Analyst at COC).

Step I Push Alerts to External Dissemination Devices: The PISR IB Subsystem publishes processed COI IED alert to the VMS.

Step J Propagate messages to external devices: VMS dissemination component distributes COI IED Alerts to the user's phone.

Step K Notify on device delivery success/failure: VMS dissemination component is notified by the device that it "was" or "was not" picked up by the user and went onto the recording.

Step L Notify on dissemination component delivery success/failure: The PISR IB Subsystem is notified by the VMS dissemination component of a success/failure to inform the user of a COI IED Alert over the phone.

Step M Publish delivery success/failure: The PISR IB publishes subscription result to the MCL AMS Sub-subsystem with a success/failure to deliver COI IED Alert to the user's voice mail.

Step N Publish COI IED Alerts: MCL AMS Sub-subsystem publishes processed (with dissemination component, message destination and users addresses) COI IED Alerts to the PISR IB Subsystem. In accordance with the COI IED Alert Distribution Profile, the 2nd dissemination component will be the SMS Server to deliver e-mails. It is external to the PISR System.

Step O Push Alerts to External Dissemination Devices: The PISR IB Subsystem publishes processed COI IED alert to the SMS Server with the timeout.

Step P Propagate messages to external devices: SMS Server dissemination component distributes COI IED Alerts to the user's mobile device.

Step Q Notify on device delivery success/failure: SMS Server dissemination component is notified by the SMS client that SMS text message "was" or "was not" viewed by the user within the specified timeout.

Step R Notify on dissemination component delivery success/failure: The PISR IB Subsystem is notified by the SMS Server of a success/failure of the user to view SMS text message with the COI IED Alert within the timeout.

Step S Publish delivery failure: The PISR IB publishes subscription result to the MCL AMS Sub-subsystem with a success/failure to succeed for to deliver COI IED Alert to the user's SMS device.

Step T Publish COI IED Alerts: MCL AMS Sub-subsystem publishes processed (with dissemination component, message destination and users addresses) COI IED Alerts to the PISR IB Subsystem. In accordance with the COI IED Alert Distribution Profile, the 3rd dissemination component will be the SMTP Server to deliver e-mails. It is external to the PISR System.

Step U Push Alerts to External Dissemination Devices: The PISR IB Subsystem publishes processed COI IED alert to the SMTP Server with the timeout.

Step V Propagate messages to external devices: SMTP Server dissemination component distributes COI IED Alerts to the user's e-mail inbox.

Step W Notify on device delivery failure: SMTP Server dissemination component is notified by the e-mail client that e-mail was not opened by the user within the specified timeout.

Step X Notify on dissemination component delivery failure: The PISR IB Subsystem is notified by the SMTP Server of a failure of the user to open the e-mail with the COI IED Alert within the timeout.

Step Y Publish delivery failure: The PISR IB publishes subscription result to the MCL AMS Sub-subsystem with a failure to succeed for to deliver COI IED Alert to the user's voice mail.

7.6 Pub/Sub-driven data flow for IED Battlefield Activities 5-7

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

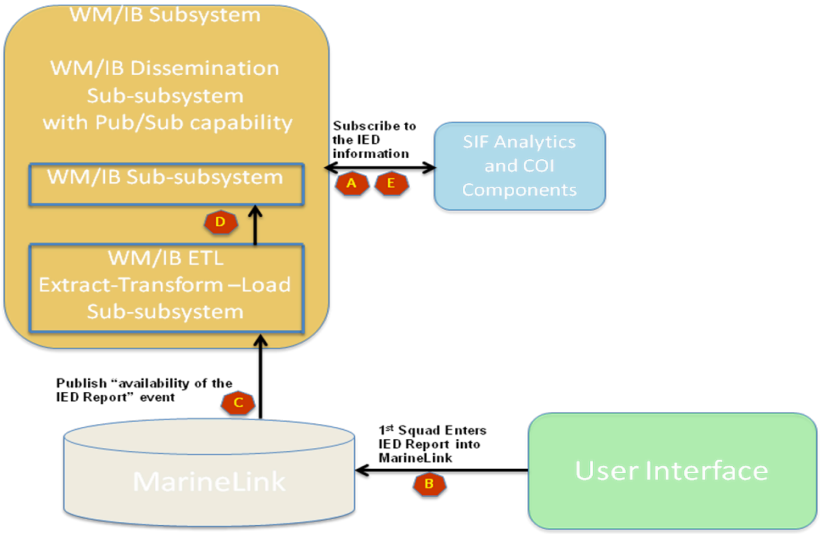

Figure 39. Pub/Sub-driven data flow for IED vignette battlefield activities #5-7

7.6.1 Objective

The purpose of Figure 39 is to describe the data flow across the PISR PLA Subsystems to perform Battlefield Activities 5, 6 and 7 from the IED Use Case Vignette.

5a. Battlefield Activity: While executing patrol mission, 1st Squad is pushed IED alert information & provided Situation Awareness (SA) of ongoing target.

5b. PISR System Activity: COI that has been satisfied by sensor information required that any Marine units operating within 1.5 km of the suspected IED site receive the Alert. 1st Squad is the only unit within the 1.5 km circle so they are pushed the Alert through the digital tactical radio system. PISR System monitors the communication system to determine if 1st Squad responds to the text message. 1st Squad does respond.

6a. Battlefield Activity: WAAS and ground-based sensor collection & analytics continue to feed HQ element's intelligence analysts with updated target case file information.

6b. PISR System Activity: Additional COIs are being created to look for additional enemy activity that may be associated with the IED emplacement. PISR System Users are monitoring WAAS and ground-based sensors looking for additional evidence of enemy activity.

7a. Battlefield Activity: 1st Squad EOD detachment locates & successfully disables IED and *pushes* information update to BN HQ and adjacent units via network.

7b. PISR System Activity: 1st Squad EOD sends out a message that IED has been destroyed. Analyst who initiated the COI is notified. Analyst used the PISR System to enter the outcome of his COI but decides to leave the COI open to detect additional IEDs in this important NAI. IED Report is filed by 1st Squad into MarineLink. PISR

System with a connection to MarineLink through PISR IB detects the information about the IED and adds the information to the PISR System IED database.

7.6.2 Subsystem Data Flow and Processing Narrative

The distribution instructions on the COI relating to possible IED activity along the convoy route in the NAI also required that any Marine units operating within 1.5 km of the suspected IED site receive the alert. The MCL Alert Management Sub-subsystem (AMS) issues one-time subscription to obtain message destinations fitting 1.5 km spatial criterion. 1st squad is the only one meeting this criterion. The AMS Sub-subsystem publishes the alert to the PISR IB Subsystem. PISR IB Subsystem transmits alert to the digital tactical radio dissemination component to 1st Squad. As above, the distribution instructions require positive acknowledgement of receipt of the alert by the 1st Squad, which the MCL AMS Sub-subsystem eventually obtains from the recipient. See Figure 38 for further details.

PISR users continue to create and publish COIs as described previously. In particular, SLI Sub-subsystems associated with WAAS and ground-based sensors continue to publish information about the battlespace. SIF Subsystem components continue to be notified by the PISR IB Subsystem of activities in the area of interest based on their subscriptions for feature, state information. SIF components continue to assess the data looking for and publishing behaviors and patterns of behavior that can indicate a threat to the convoy movement over the route. The COI Interpreter Sub-subsystem continues to receive subscription results from the PISR IB Subsystem based on subscriptions on active COIs. PISR users continue to be notified on satisfied COI alerts through the dissemination components originating from the MCL Alert Management Sub-subsystem and distributed via the PISR IB Dissemination Sub-Subsystem. Refer to Figure 37 for further details.

Receiving notification through tactical communications that the IED has been destroyed, the PISR user (intel analyst who had initiated the COI regarding suspected IED emplacement along the convoy route in the NAI) calls up the COI Editor Sub-subsystem in the User Interface Environment Subsystem. Using the editor, the user queries for the list of active COIs that he has created and selects the one in question. He enters information regarding the performance of this COI (i.e., that it was successful in notifying the force of the IED threat, as confirmed by the EOD detachment actions) and re-publishes it to the PISR IB Subsystem through the User Interface Environment Subsystem.

1st Squad enters the IED Report into MarineLink (e.g., as a CIDNE SIGACTS or Event report). The PISR IB Subsystem is notified of the new information through the PISR IB ETL Sub-subsystem based on its external subscriptions to this data source. IED information is published into the PISR IB Subsystem and the PISR IB Dissemination Sub-subsystem notifies any PISR components with subscriptions for IED information.

7.6.3 Step-by-step dataflow

Step A Subscribing to the IED information: SIF analytics subscribed for the IED information.

Step B Entering IED Report into MarineLink: 1st Squad enters IED Report into MarineLink via UI Subsystem.

Step C Publishing IED Report event into the PISR IB ETL Sub-subsystem: IED Report, entered into the MarineLink, causes generation of "availability of the IED Report" event from within the MarineLink database. This event is pushed to the PISR IB ETL Sub-subsystem.

Step D Publishing IED Report into the PISR IB: The PISR IB ETL Sub-subsystem captures and processes "availability of the IED Report", which is resulting in transforming and loading the metadata of the IED report into the PISR IB.

Step E Publishing IED information to the subscriber: IED subscription result set is published by the PISR IB to the SIF analytics.

7.7 Pub/Sub-driven data flow for IED Battlefield Activities 8-10

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

7.7.1 Objective

8a. Battlefield Activity: Positive Target ID/Location

8b. PISR System Activity: Human Intelligence reports are flowing in to the Command Operations Center (COC) indicating that combatants that implanted the IED have been identified and their location is known. These reports are detected by the PISR System and information from the PISR System helps support developing a mission plan to engage the combatants. Analysts and other Users who received the Alert are using the PISR System and other COC resources to plan a mission to attack the combatants. PISR System supports the fusion of real time intelligence and non-real time intelligence to support mission planning and to improve Situation Awareness.

9a. Battlefield Activity: Combatants Destroyed

9b. Battlefield Activity: PISR System is used with other tools to understand that the combatants have been identified, located, and that little collateral damage will result from an immediate attack on the combatants. Mission is executed and combatants are destroyed. Analysts and other Users connected to the PISR System update their Case Files and COIs with this new information. PISR System monitors several other data sources including MarineLink and pulls information from those sources into PISR IB for future use.

10a. Battlefield Activity: Mission continues. BN/CO S-2 continues to monitor situation & developments. GBOSS, WAAS, Ground Sensors continuously monitor signature events.

10b. PISR System Activity: PISR System continuously looks for conditions that satisfy COIs and continues to support User interface and information flow to and from the PISR System. (Reference the Steady-State Use Case)

7.7.2 Subsystem Data Flow and Processing Narrative

HUMINT reporting is another data source connecting to the PISR System through an associated SLI Sub-subsystem. The SLI Sub-subsystem extracts data from the reports to create type or schema instances to publish to the PISR IB Subsystem. The PISR IB provides subscription results to any PISR PLA Subsystems/components subscribing to information content interpreted by SLI Sub-subsystem. PISR users create information requirements or COIs to subscribe to identities and location information on the individuals who were detected performing the IED emplacement. SLI sub-subsystems, SIF sub-subsystems, and the COI Interpreter Sub-subsystem interact through pub-sub mechanisms described previously to create the situational awareness (including fusion of the real-time and non-real-time data) to provide identification and location information of sufficient certainty to enable initiation of planning to engage the combatants. Models in the SA Subsystem can provide estimates of mission success and collateral damage that could occur based on what is believed to be true about the situation and different engagement options (e.g., type and quantity of ordnance, delivery methods, etc.).

External to the PISR System, intel analysts work with operations planners to evaluate the situation and plan an attack on the combatants. As needed, the intel analysts use the PISR System User Interface Environment Subsystem to view information about the situation to assist in the mission planning. They are able to view data in a variety of presentation modes, such as geographic information system (GIS), textual reports, or other methods provided by the User Interface Environment Subsystem. The information is obtained through one-time and continuous subscriptions specified through the User Interface Environment Subsystem and processed by the PISR IB Subsystem.

A PISR user at the Battalion COC creates a collection plan and COIs to focus ISR assets on the mission area for purposes of damage assessment following the attack. Processing of the collection plan proceeds as described previously in IED Battlefield Activity 1. After the mission is executed, information from the sensors is processed by associated SLI sub-subsystems, SIF Sub-subsystem components, and the COI Interpreter resulting in the conclusion that the combatants have been destroyed at some level of certainty. Data flows are identical to earlier descriptions. See Figure 37 for further details.

No additional description is needed for Activity 10. It is similar to the ongoing use of the PISR System as described in IED Battlefield Activity 6 and in the Steady-State Use Case.

7.8 Mapping Use Cases to PISR Architecture

[NOTE: This subsection is in modification for version 1.1 of this PISR PLA specification.]

This section describes various data flow and processing activities related to two important architectural use cases: (1) PISR System steady state operation; (2) opening a new case file and entering a COI.

7.8.1 Use Case Mapping for PISR Steady State Condition

1. Activity: A Marine Intelligence Analyst is using the Command Post of the Future, MarineLink, and other information sources to perform intelligence analysis. The Analyst is not accessing the PISR System.

- UI Subsystem Functions: No Activity
- PISR IB Subsystem Functions: No Activity
- SA Subsystem Functions: No Activity
- MCL Subsystem Functions: No Activity

2. Activity: MarineLink is connected to the PISR System through the External Data Provider interface to the World Model/Information Base (PISR IB). SA is extracting information from the unstructured text within MarineLink and passing the structured data back to PISR IB.

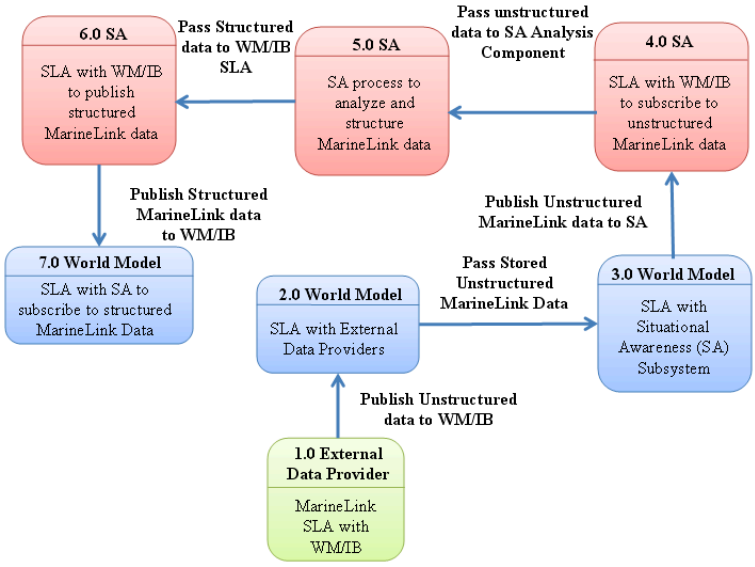

Figure 40. Data flow diagram for extracting useful information from MarineLink

- UI Subsystem Functions: No Activity
- PISR IB Subsystem Functions: External Data Provider (function 1.0 in the Data Flow Diagram (DFD) in Figure 40) is publishing unstructured MarineLink data to the PISR IB through a Service Level Agreement (SLA). PISR IB (DFD function 2.0) has a subscription to each external data provider including MarineLink. This data is unstructured strings of text, figures, and tables. Unstructured test is being stored in the PISR IB. PISR IB published unstructured text to SA through an SLA. After SA completes its work, PISR IB subscribes to SA to get structured MarineLink text. PISR IB adds additional indexing and structure and stores the information.

- SA Subsystem Functions: SA is receiving the unstructured data through a SLA subscription to the PISR IB. SA passes the unstructured data to an analytical component inside SA that parses, indexes, and annotates the unstructured text. Structured text is published back to the PISR IB through an SLA where PISR IB adds additional indexing and search ability.
- MCL Subsystem Functions: No Activity

3. Activity: The COI/IR Sub-subsystem continuously looks for information across the PISR System to determine if a Condition of Interest (COI) has been met. The "COI Editor Sub-subsystem" is also ready for a user to open, create, edit or delete a COI.

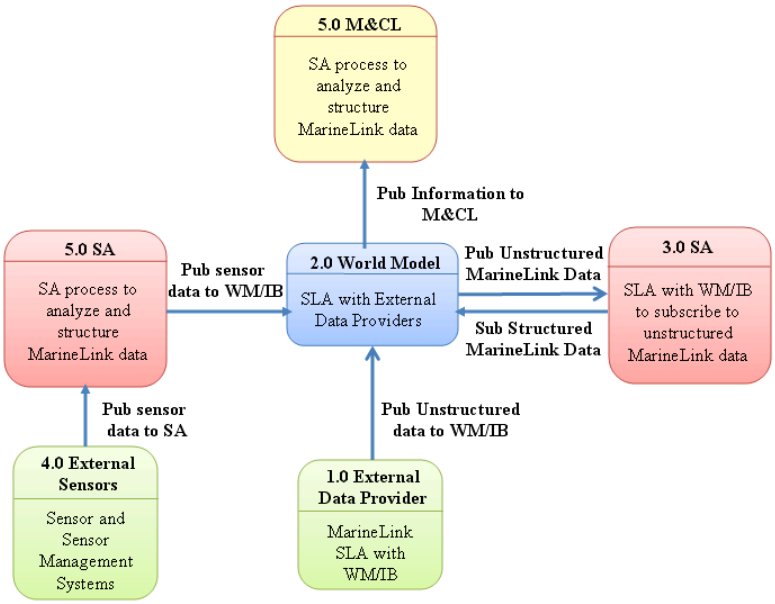

Figure 41. MCL monitors PISR System to detect COI status

- UI Subsystem Functions: No Activity
- PISR IB Subsystem Functions: PISR IB subscribes to information from SA and is taking information from external data providers.
- SA Subsystem Functions: SA is connected to external sensors and external sensor management and control systems that are feeding data into SA. The COI/IR Sub-subsystem is deployed throughout the PISR System to look for indications that a COI has been satisfied.
- MCL Subsystem Functions: No Activity.

4. Activity: The MCL manages health status monitoring of all PISR System resources. Each Subsystem collects and reports its health status. SA also collects and reports sensor health status. Real-time information may pass directly between the Subsystems and Sub-subsystems and may also be managed by the MCL.

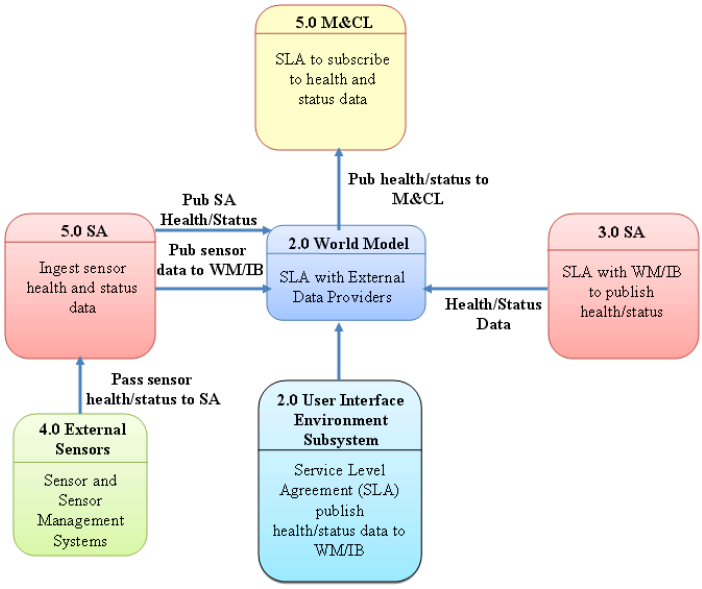

Figure 42. Health and status information about the PISR System and sensor components flows into the PISR IB

- UI Subsystem Functions: No Activity
- PISR IB Subsystem Functions: PISR IB maintains all PISR System health status as well as all health and status information about external systems passed to the PISR IB from SA. PISR IB notifies MCL when health status updates are available through established subscriptions.
- SA Subsystem Functions: SA monitors the health and status of the external sensor and sensor management systems connected to the PISR System. Information is passed to PISR IB for storage.
- MCL Subsystem Functions: MCL monitors the health status of all PISR components to perform optimizations and to detect possible anomalies with the components.

5. Activity: The "Case File Editor Sub-subsystem", located in the "UI Environment Subsystem", continuously looks for information indicating that a case file needs creating or editing. The "Case File Editor Sub-subsystem" also has access to the "World Model" for storing and retrieving case file information.

- UI Subsystem Functions: No Activity until a User decides to enter a new case file.
- PISR IB Subsystem Functions: No Activity until a User decides to enter a new case file.
- SA Subsystem Functions: No Activity
- MCL Subsystem Functions: No Activity

6. Activity: The PISR IB connects to all of the other PISR subsystems to gather and provide needed information. A mix of Redundant Array of Inexpensive Disks (RAID) technology and database processes that automatically make copies of active databases insure that a copy of all databases exist at all times.

- UI Subsystem Functions: No Activity
- PISR IB Subsystem Functions: PISR IB Subsystem Functions: PISR IB continuously receives health and status information and other information being published to the PISR IB from the PISR System

subsystems. PISR IB uses automated process to make copies of data and image level copies of complete hard drives.

- SA Subsystem Functions: No Activity
- MCL Subsystem Functions: No Activity

7. Activity: The "Situational Awareness Subsystem" (SA) connects to live sensor feeds. SA pulls information from the sensors and uses analytics within The "Sensor Level Interpretation Sub-subsystem" (SLI) to interoperate sensor data. Un-interoperate sensor data also flows through the SA. SA is responding to User and system defined COIs to use sensor data to provide situational awareness and situational understanding to the User.

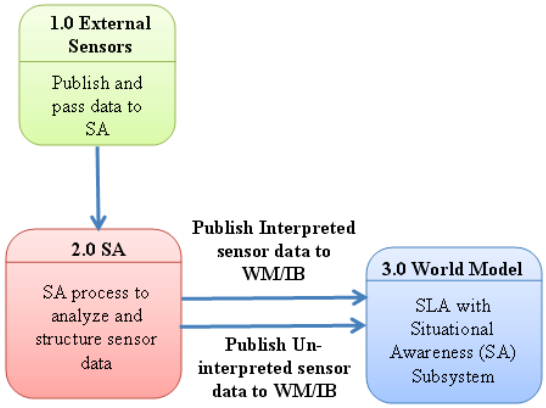

Figure 43. SA continuously publishes interpreted and uninterpreted sensor data to PISR IB

- UI Subsystem Functions: No Activity
- PISR IB Subsystem Functions: PISR IB continuously subscribes to interpreted and un-interpreted sensor data from SA.
- SA Subsystem Functions: SA interprets sensor data and publishes the interpreted sensor data to PISR IB. SA may also publish un-interpreted sensor data to PISR IB
- MCL Subsystem Functions: No Activity

7.8.2 Use Case Mapping for Opening a New Case File and Entering a Condition of Interest

1. Activity: The Intel Analyst decides to build a new Case File and signs in to the PISR System using the Single Sign On capability located in the "User Authentication Framework" that provides the Analyst access to all of the datasets he will need.

- User Authentication Framework Functions: Deferred.
- UI Subsystem Functions: Presents a log-in screen for a user to input their credential information. Shows failure messages for incorrect authentication information.
- PISR IB Subsystem Functions: No Activity.
- SA Subsystem Functions: No Activity.
- MCL Subsystem Functions: No Activity.

2. Activity: Analyst's User Name and Password triggers the "User Interface Environment" to request a "User Profile" for this User from the "World Model", pass that "User Profile" through the "Dissemination Subsystem" to the "User Interface Environment Subsystem".

    - UI Subsystem Functions: User accesses the Security and Accreditation Framework through the UI. User Profile is presented to the User through the UI.
    - PISR IB Subsystem Functions: Publishes the User Profile to UI.
    - SA Subsystem Functions: No Activity.
    - MCL Subsystem Functions: No Activity.

3. Activity: User enters the basic information about his Use Case. During data entry, the "Use Case Editor Sub-subsystem" continuously queries the "PISR IB Subsystem" to identify other Use Cases or COIs that may have information relative to the Use Case being built. If a link is found, the "Use Case Editor Sub-subsystem" alerts the User during Use Case development.

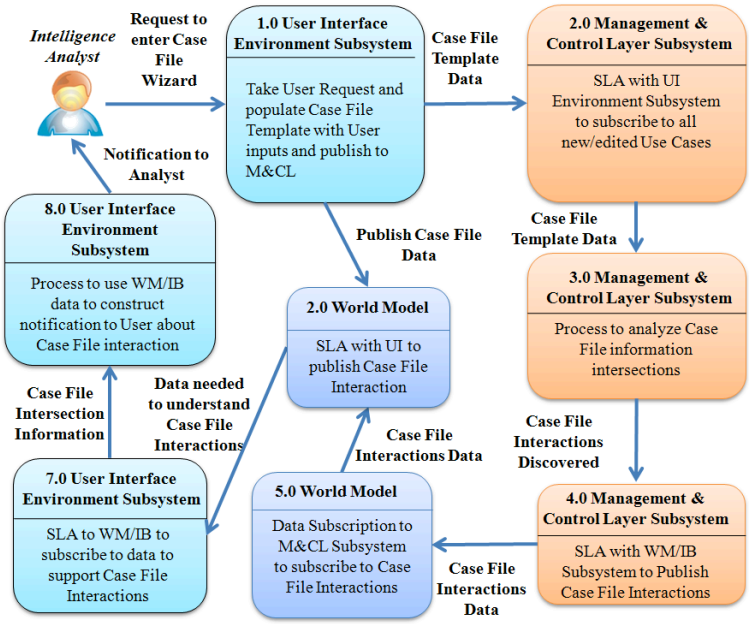

Figure 44. Analyst builds new case file and is notified by MCL of possible interactions between the new case file and existing case files

- UI Subsystem Functions: User accesses all of the case file tools through the UI. The UI publishes data to the PISR IB and subscribes to important information to support case file development.
- PISR IB Subsystem Functions: Publishes all important case file information to the Analyst through the UI. PISR IB subscribes to data during case file development. PISR IB publishes case file information to MCL and then subscribes to MCL notifications if a notification is given. PISR IB publishes the notification to UI.
- SA Subsystem Functions: Monitors case file development process through subscriptions with PISR IB to identify possible interactions with this new case file and other case files or COIs currently active. If an interaction is identified, MCL develops a notification and publishes that notification to the PISR IB which publishes to the UI.
- MCL Subsystem Functions: No Activity.

4. Activity: User decides that some additional sensors will be needed to support intelligence gathering for this case file. User accesses the "Sensor Plan" Wizard to discover sensors about a target AOI. User selects sensors needed and saves them as a "Sensor Plan". User reviews the plan, edits the plan and then saves the plan. The plan is saved in the "World Model"
    - UI Subsystem Functions: User accesses all of the Sensor Plan tools through the UI. The UI publishes data to the PISR IB and subscribes to important information to support Sensor Plan development.
    - PISR IB Subsystem Functions: Publishes all important Sensor Plan information to the Analyst through the UI. PISR IB subscribes to data during Sensor Plan development. PISR IB publishes Sensor Plan information to MCL and then subscribes to MCL notifications if a notification is given. PISR IB publishes the notification to UI.
    - SA Subsystem Functions: Keeps PISR sensor status and registration up to date so that MCL can have the proper information about various sensors within the PISR System.
    - MCL Subsystem Functions: Monitors Sensor Plan development process through subscriptions with PISR IB to identify possible interactions between this Sensor Plan and other Sensor Plans that are currently active. If an interaction is identified, M&CL develops a notification and publishes that notification to the PISR IB which publishes to the UI.

5. Activity: User pulls up a list of Battalion points-of-contact and selects the Battalion Command. User than selects "Email" from a list of ways to communicate with the Battalion Commander and posts the "Sensor Plan" to the Battalion Commander requesting the resources necessary to monitor the road intersection of interest. The "Sensor Plan" is stored in the "World Model" and transmitted to BN HQ through the "Dissemination Subsystem" using the Email Binding Component in the "Dissemination Subsystem".
    - UI Subsystem Functions: User accesses all of the distribution tools through the UI. The UI publishes data to the PISR IB and subscribes to important information to support dissemination.
    - PISR IB Subsystem Functions: Publishes all information needed by the Analyst through the UI. PISR IB subscribes to data during dissemination planning. PISR IB publishes dissemination data to data recipient and notifies Analyst that message has been sent.
    - SA Subsystem Functions: No Activity
    - MCL Subsystem Functions: No Activity

6. Activity: The Sensor Plan is reviewed and vetted at Battalion and approved by the Battalion Commander. The Sensor Plan is sent to the Company Commander who releases to the Combat Operations Center and the User.
    - UI Subsystem Functions: User is notified through the UI that his Sensor Plan has been approved and that Sensor Plan is ready for implementation.
    - PISR IB Subsystem Functions: Subscribes to Sensor Plan from higher HQ. PISR IB publishes Sensor Plan to MCL and UI.
    - SA Subsystem Functions: No Activity
    - MCL Subsystem Functions: MCL subscribes to PISR IB to get approved Sensor Plan for sensor re-tasking. This sensor re-tasking drives the creation of new Collection Plans and results in the MCL notifying the SA of sensor configuration changes.

7. Activity: The user signs on to the PISR System and selects to add a Condition of Interest (COI).
    - User Authentication Framework Functions: Deferred.
    - UI Subsystem Functions: User obtains all COI tools through the UI. Leverages the COI/IR Ingest Sub-subsystem for COI development.
    - PISR IB Subsystem Functions: PISR IB is queried for various COI templates for enhancement or use.
    - SA Subsystem Functions: COI/IR Ingest Sub-subsystem queries PISR IB for COI templates.
    - MCL Subsystem Functions: No Activity.

8. Activity: User enters the "COI Editor Sub-subsystem" and selects the "Condition of Interest Wizard" to help him build the COI.

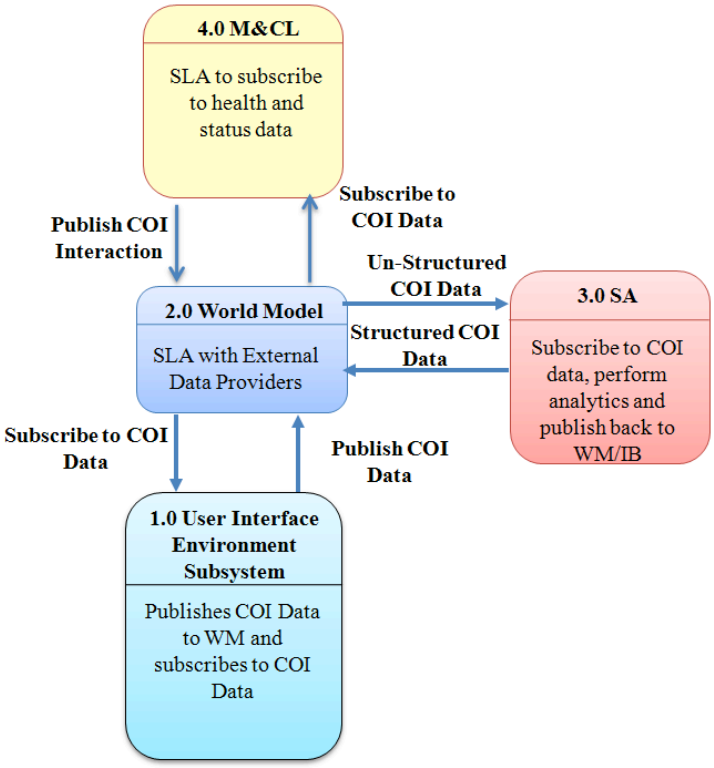

Figure 45. Analyst creates a COI

- UI Subsystem Functions: User obtains all COI tools to support COI development through the UI. User enters COI information, edits information and posts COI to PISR System through the UI. UI generates the Wizard based on COI templates received from COI/IR Ingest Sub-subsystem.
- PISR IB Subsystem Functions: PISR IB stores to COI states during COI development. PISR IB publishes progress on COI development back to Analyst through the UI.
- SA Subsystem Functions: COI/IR Ingest Sub-subsystem helps UI in its COI Wizard through querying the PISR IB.
- MCL Subsystem Functions: Produces any health information about the PISR that might be useful for COI development.

9. Activity: The COI Wizard asks a series of questions. These questions and answers are shown in **Error! Reference source not found.**.
    - UI Subsystem Functions: No Activity
    - PISR IB Subsystem Functions: PISR IB stores to COI states during COI development. PISR IB publishes progress on COI development back to Analyst through the UI PISR IB notifies other subsystems COIs in development so that the COI/IR Ingest Sub-subsystem can validate the COI as well as let the MCL validate a COI against collection, process, and dissemination plans.
    - SA Subsystem Functions: COI/IR Ingest Sub-subsystem helps UI in its COI Wizard through querying the PISR IB. Validates COIs being developed.

- MCL Subsystem Functions: MCL subscribes to PISR IB to get information about COI details in development for possible interaction or conflict with collection or process plans. If MCL detects a possible conflict, an alert is generated for the user developing the COI.

10. Activity: User now reviews his COI, decides that COI is correct and then confirms his new COI for inclusion in the PISR System.
    - UI Subsystem Functions: User obtains all COI tools through the UI. User enters COI information, edits information and posts COI to PISR System through the UI.
    - PISR IB Subsystem Functions: PISR IB publishes the COI through the Dissemination Sub-Subsystem to targeted SA Subsystem instances.
    - SA Subsystem Functions: PISR IB publishes COI information to SA so SA can interpret the COI and perform the necessary analytics to extract actionable intelligence from the COI. SA subscribes to PISR IB to get this information and then publishes the analyzed COI and associated files and indices back to PISR IB.
    - MCL Subsystem Functions: MCL subscribes to PISR IB to get information about COI details to develop new process plans and collection allocation plans focused on the highest valued information to satisfy that COI taking into account other information needs.

11. Activity: The PISR System automatically sends a notification that a case file and a COI are active to the following; a. Any user with an interest in this same map grid; b. Any user with an interest in this same intersection; c. Any user with a collection plan collecting data from this same map grid; d. Any user using sensors on GBOSS tower 22 Bravo.
    - UI Subsystem Functions: User obtains all COI tools through the UI. User enters COI information, edits information and posts COI to PISR System through the UI. Verification that COI has been transmitted is sent to Analyst through the UI.
    - PISR IB Subsystem Functions: PISR IB subscribes to COI during COI development. PISR IB publishes progress on COI development back to Analyst through the UI. PISR IB is also publishing details of COI to the MCL for possible identification of interaction with other COI or Case File. PISR IB publishes the final to Analyst through UI for final approval. PISR IB then publishes the COI through the Dissemination Sub-Subsystem to those identified by the Analyst and those identified by MCL.
    - SA Subsystem Functions: No Activity
    - MCL Subsystem Functions: MCL subscribes to PISR IB to get information about COI. MCL maintains special distribution rules and conditions and activates these rules to send alerts to people not selected by Analyst. MCL is responding to standing operating procedures (SOPs) and other needs.

12. Activity: The PISR System automatically develops a summary of activity related to this Case File and COI and presents a report to the User. The report contains; a. Users who have active tasking for other sensors on GBOSS tower 22B; b. Users who have active Case Files within map Grid 453224; c. Users who have active case files or COIs relative to crossroad of road 22C and 34A; d. List of patrols, convoys, and other Blue Force activities planned for this crossroad during the period of this COI; e. List of known Red Force activities planned for this crossroad during the period of the COI; f. List of know humanitarian, news stories and other civilian activities planned for this crossroad during the period of the COI; g. Short weather report for the Area of Interest including weather for the last 30 days and weather predictions by day for the period of the COI.

    - UI Subsystem Functions: No Activity
    - PISR IB Subsystem Functions: No Activity
    - SA Subsystem Functions: No Activity
    - MCL Subsystem Functions: No Activity

13. Activity: User reviews the COI, reviews the activities planned for this COI for the period of the COI and decides that the COI if finished and posts the COI to the PISR System.

    - UI Subsystem Functions: No Activity
    - PISR IB Subsystem Functions: No Activity

- SA Subsystem Functions: No Activity
- MCL Subsystem Functions: Health status is continuously monitored about PISR resources affecting this COI. As a COI is revisited, this information is made available to the User for analysis.

8 Rapid Prototyping Process

8.1 Overview

As introduced in Section 1, the foundational rapid prototyping strategy is to improve warfighter effectiveness by quickly fielding advanced PISR systems using the construct of product line architecture (PLA) and judicious application of off-the-shelf technologies. Defining this PLA and applying it to develop a family of PISR systems requires a process that is stakeholder-driven, repeatable, and agile enough to respond to new opportunities, user feedback, and unexpected events. This section summarizes software engineering best practices for that process, recognizing the particular challenges of constructing software-intensive systems based on off-the-shelf components. Several reference publications are liberally quoted – personnel actively engaged in the development process are encouraged to peruse them directly, in particular, *Evolutionary Process for Integrating COTS-Based Systems (EPIC)*[25] by the Software Engineering Institute (SEI).

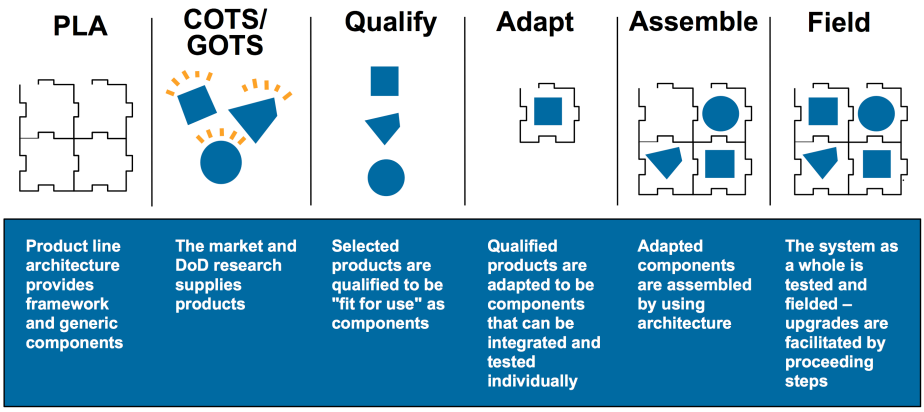

Figure 46: Simplified view of the rapid prototyping process[26]

A more comprehensive view of this prototyping process, shown in

Figure 47, depicts the stakeholders, policies, and existing operational environment (the "brownfield") that shape the PISR product line architecture. Once defined, the PISR PLA acts as a stable foundation for a family of PISR system products. Each system is designed for a particular environment, scale, timeframe, cost, and set of "A"-priority quality attributes, specifying a set of components that meet these constraints. These include the sensors collecting raw data, the historical/contextual databases that will be consulted, and the analytic processes that will be configured. In order to deliver a system rapidly and minimize risk, predominantly mature, off-the-shelf products are chosen from commercial and government sources, though not to the exclusion of highly valuable, but less mature, components.

Following Product Design, Development & Integration constructs the adapters, glue code, and infrastructure necessary to assemble the components into a unified system. Lab-based testing of individual components and the partially integrated system occurs on a frequent basis, reflecting the iterative and incremental approach of the PISR

[25] Albert, C., Brownsword, L., *Evolutionary Process for Integrating COTS-Based Systems (EPIC)*, 2002, http://www.sei.cmu.edu/reports/02tr009.pdf

[26] Adapted from Sureesong, K., *COTS-based System*, http://userpages.umbc.edu/~cseaman/ifsm698/spr01/COTS.ppt

rapid prototyping process. Field-based alpha and beta tests executed in a relevant environment provide an opportunity for a broad spectrum of end-user feedback and solidify trust in the assembled system before it is certified and accredited for a formal Field User Evaluation (FUE).

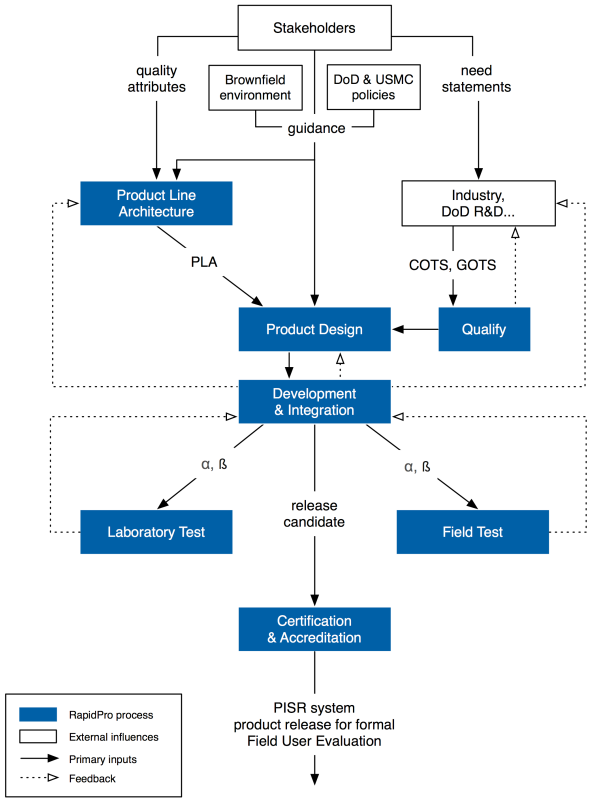

Figure 47: PISR rapid prototyping process

Lastly, this production process is intended to scale to multiple parallel iterations, as shown in Figure 48, targeting three fielded systems per year. A product line architecture re-evaluation step is shown at the start of each PISR system release train, however, it is expected that major PLA refinements will only occur once a year.

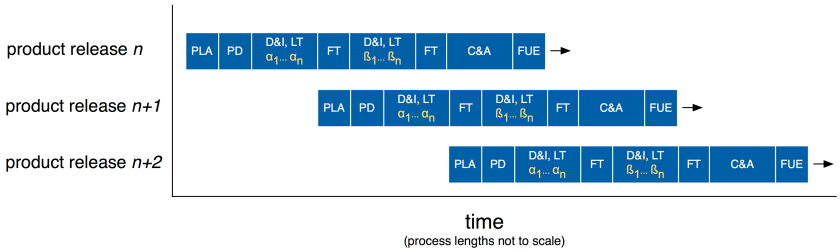

Figure 48: Pipelined, evolutionary product releases

8.2 Development methodology

While almost any methodology can be made to produce software, light processes are more often successful.[27] For rapid prototyping, we advocate a methodology that implements the principles and practices advocated by the agile software development community:

Principles[28]	• Iterative and Incremental development via sprints (time-boxes) that deliver small amounts of tested, ship-ready code • Regular adaptation to changing circumstances • Working software is the primary measure of progress • Simplicity – the art of maximizing the amount of work not done – is essential • Sustainable pace
Practices	• Unit testing, fuzz testing • Continuous integration • Extensive code coverage • Code standards • Effective, minimal-overhead metrics to evaluate progress

One suitable methodology is the Agile Unified Process (AUP)[29]. AUP strikes a balance between a heavyweight approach like the Rational Unified Process (RUP)[30] and the relatively documentation-scarce style of Extreme Programming (XP)[31]. As with most methodologies derived from the Unified Process (UP)[32], it divides a project into four phases:

1. *Inception*: identify the initial scope of the project, its architecture, and obtain stakeholder acceptance

[27] Cockburn, A., *Characterizing people as non-linear, first order components in software development*, HaT Technical Report 1999.03, Oct 21, 1999, http://alistair.cockburn.us/Characterizing+people+as+non-linear,+first-order+components+in+software+development
[28] Agile Manifesto, 2001, http://agilemanifesto.org/principles.html
[29] Agile Unified Process, http://www.ambysoft.com/unifiedprocess/agileUP.html
[30] IBM Rational Unified Process, http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process
[31] Extreme Programming, http://www.extremeprogramming.org
[32] Unified Process, http://en.wikipedia.org/wiki/Unified_Process

2. *Elaboration*: prove the architecture of the system
3. *Construction*: build working software on a regular, incremental basis which meets the highest-priority needs of the stakeholders
4. *Transition*: certify, accredit, and field the system into the operational environment

Each phase consists of one or more time-boxes (sprints) that act as regular checkpoints to assess progress, handle business process "exceptions", re-prioritize tasks, obtain new stakeholder feedback, and motivate implementers with goals that are within sight.

The Unified Process also defines several activities (disciplines) that are executed throughout the project phases: modeling, requirements gathering, analysis & design, implementation, test, and deployment. Figure 49, below, visualizes several characteristics of this approach:

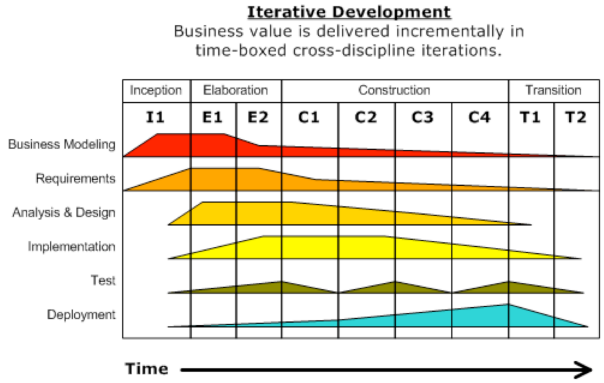

Figure 49: Relative emphasis of different disciplines over the course of a project[32]

Note that:

- Implementation begins before requirements have fully stabilized; nascent, partial solutions allow early feedback from stakeholders and encourage early, rather than late, changes
- Testing occurs throughout the process, resulting in regularly produced chunks of stable, tested functionality
- Deployment activities begin early – this does not refer to fielded systems, but to lab-based staging areas that mimic the operational environment; components are deployed and tested end-to-end through automated *continuous integration* processes

From the general framework provided by the Agile Unified Process, specific activities have been identified and emphasized for USMC PISR rapid prototyping – they are the focus of the rest of this sub-section.

8.2.1 Prioritized quality attributes

In the Inception phase, the rapid prototyping process employs a facilitated method called a Quality Attribute Workshop (QAW)[33] to engage stakeholders early in the system development life cycle and discover the driving quality attributes of the desired architecture and eventual PISR systems. The QAW provides an opportunity to gather stakeholders together to provide input about their needs and expectations with respect to key quality attributes that are of particular concern to them. Participants are individuals on whom the resulting PLA and systems will have significant impact, such as end users, administrators, trainers, architects, acquirers, engineers, and others. In general, the workshop

[33] Barbacci, M., et al., *Quality Attribute Workshops, Third Edition*, 2003, http://www.sei.cmu.edu/library/abstracts/reports/03tr016.cfm

should have at least 5 participants and not more than 30. In preparation for the workshop, stakeholders receive a "participants handbook" providing initial quality attribute taxonomies, questions, and scenarios.

The contribution of each stakeholder is essential during a QAW; all participants are expected to be fully engaged and present throughout the workshop. Participants are encouraged to comment and ask questions at any time during the workshop. However, it is important to recognize that facilitators may occasionally have to cut discussions short in the interest of time or when it is clear that the discussion is not focused on the required QAW outcomes.

After overviews of the known high-level goals, drivers, and requirements of the architecture/systems, participants review and discuss the initial quality attributes and scenarios and brainstorm new ones. Scenarios typically fall into three general categories – each type should be represented in the QAW:

- use case scenarios - involving anticipated uses of the system
- growth scenarios - involving anticipated changes to the system
- exploratory scenarios - involving unanticipated stresses to the system that can include uses and/or changes

Next, similar quality attributes/scenarios are consolidated to ensure that during the subsequent voting process, stakeholders do not split their votes, possibly relegating important quality attributes below a threshold of consideration. Finally, each stakeholder is allotted a number of votes equal to 30% of the total number of quality attributes/scenarios generated after consolidation. Stakeholders can allocate any number of their votes to any scenario or combination of scenarios. The votes are counted, and the quality attributes/scenarios are prioritized accordingly.

This prioritized list of quality attributes with associated scenarios, but primarily the top 30%[34], are used to:

- update the architectural vision
- refine system and software requirements
- guide initial implementation
- influence the order in which the architecture is developed
- describe the operation of a system

The PISR rapid prototyping process will conduct Quality Attribute Workshops once a year, as part of major Product Line Architecture re-assessment.

8.3 Component qualifications

8.3.1 Overview

A critical part of a PISR system product design is the qualification and selection of commercial and government components that are likely to support the system's targeted environment, scale, timeframe, cost, and quality attributes. Off-the-shelf packages will impose additional constraints and requirements, and depending on component maturity, introduce risk that must be mitigated through additional testing.

The *Evolutionary Process for Integrating COTS-Based Systems (EPIC)*[25], referenced earlier in the parent section, describes a comprehensive approach for screening off-the-shelf-components[35]. While it describes a documentation-heavy approach via the creation of very detailed *Component Dossiers*, we encourage practitioners to adapt it as appropriate, focusing on capturing the highest-value data. Information typically captured in the Component Dossier includes characteristics of the vendor, component architecture and functional capabilities, standards supported, required hardware and software configurations, non-functional characteristics like usability, supportability, reliability, interoperability, portability, and scalability, and quality of documentation, costs, and licenses.

Component criteria specific to the PISR product line architecture are discussed in the following sub-sections; please refer to *EPIC*[25] for general guidance on software-intensive off-the-shelf based systems.

[34] Referred to as "A" priorities elsewhere in this document
[35] Throughout the Inception, Elaboration, Construction, and Transition phases, but in detail in chapters 8 and 9

8.3.2 Capabilities

The primary qualifying characteristic of a candidate PISR component is the set of capabilities it brings that provide measurable value for the USMC warfighter. During the product design for a particular PISR system instance, priority use cases will be identified – for example, IED emplacement detection or convoy protection. In order to objectively assess a candidates' contribution to these priority use cases, several pieces of data are beneficial. In the case of a candidate sensor, these would include:

- rate of detection/classification (i.e., ROC curve)
- coverage area
- temporal coverage (e.g., day or night)
- terrain factors
- weather/seasonal factors
- communication range
- detection/classification latency

These characteristics should be evaluated against historical and anticipated enemy tactics, techniques, and procedures (TTPs) and the targeted physical environment in the context of the primary use cases.

Components extending or replacing functionality in other sub-systems of the PISR PLA will be evaluated against other performance metrics. For example, an optimization engine for the Management and Control (MCL) subsystem might be evaluated in the context of its:

- model scope
- number and type of variables
- number and type of objectives
- speed of calculation
- quality of results

Lastly, it is useful to analyze the capabilities of a candidate component in the context of other candidates and fielded systems – the sum of the whole may be greater than the parts. Conversely, packaged off-the-shelf components can often be decomposed – sub-components may deliver value in their own right and may be more readily integrated and deployed. For example, a software-based video analytic that has been traditionally optimized for and deployed with low-mounted, fixed field-of-view high-definition cameras could be paired with an existing tower-mounted, pan-zoom-tilt camera.

8.3.3 Dependencies & Requirements

The target physical environment of a given PISR system will impose several constraints on component selection, in particular, the available:

- physical volume
- computational power
- electric power
- network capacity, latency, and connectivity
- volatile and persistent storage capacity
- local and network services (e.g., high-res. base maps, weather forecasts, etc.)
- human operators (limited in quantity as well as training)

A candidate component for a PISR system should have accompanying artifacts that document its dependencies and the expected resource consumption. In addition, in order to evaluate if multiple software components can be hosted

on shared hardware platforms, it is important to understand other low-level runtime characteristics. For example, if the process is compute or disk bound, if the resource consumption is relatively constant or occurs in bursts, or if it typically executes many small disk reads/writes, or fewer larger reads/writes.

8.3.4 Interfaces

The PISR product line architecture identifies several categories of interfaces that promote extensibility, manageability, and robustness of the resulting systems. Excellent candidates for integration will have an architecture that matches well with their design. Specific interfaces are described in detail in their corresponding sections (ref. Situational Awareness, Management and Control, PISR Information Base) – general interface qualities sought in components are described here.

Open data

Both the required input and expected output of a component must be documented, well-defined, and freely sharable with DoD civilians and contractors performing PISR system integration.

Testable

To support a laboratory test harness, components should have interfaces that permit replay of recorded or generated input. Testable components will enable continuous integration and C&A processes that precede larger-scale, field-based testing. Products that were developed with a significant focus on testing and have existing test suites are preferred.

Event-based

Ideally, components should support event/push-based mechanisms for data delivery, contributing to the architecture goal of near-real time processing.

Diagnosable

Components should actively, or upon request, be able to report their status. The status of a component might include meta-data such as data processing rate, resource consumption, errors in sub-components, or availability of services. Basic self-diagnostics should be executable upon request.

Controllable

Primary component functions should be controllable, in near real-time, by external sub-systems of the PISR architecture. For a camera, this might include focus, zoom, pan, tilt, or color calibration. For an analytic, it might include sensitivity settings that affect false positives or negatives, or where to publish output.

8.3.5 Information Assurance

As discussed in detail Section <9>, the PISR architecture promotes privilege separation through use of virtualization – isolating components and exposing only the operating system and network services required for it to accomplish its work. Software components should be capable of running in a virtualized environment. Further, candidate components should be designed with the principle of least privilege – for example, software processes should execute under a limited user account, and not execute as a superuser (e.g., root or Administrator).

Components should also support encrypted and authenticated communication channels or be easily adapted through a tunneling proxy (e.g., stunnel[36]). It should expose a minimal network attack surface and operate network services through registered ports, configurable at deployment time. Other IA qualification considerations include:

- Ability to delegate access control decisions to the centrally managed PISR IA policy decision point
- Secure logging of resource access to permit audits
- Constructed using programming languages that by design reduce sources of security vulnerabilities, like buffer overflows
- Developed under a process that incorporates code audits, static analysis, or formal methods

In conclusion, candidate components that already employ standardized security mechanisms and communication protocols such as Transport Layer Security (TLS), Kerberos, the Security Assertion Markup Language (SAML), the Xtensible Access Control Markup Language (XACML), and operating-system-enforced mandatory access controls will more readily integrate into the PISR architecture.

[36] stunnel, a multiplatform SSL tunneling proxy, http://stunnel.mirt.net

9 Test, Evaluation & Certification (Test/Cert) Framework

9.1 Introduction

Rapid Prototyping (RapidPro) delivers incremental PISR products to the U.S. Marines through the use of the PISR PLA. Delivered PISR components share a common, managed set of capabilities that comprise the core of the PLA. Additional hardware and software (HW/SW) components are added to the core to meet critical Marine needs. The Test, Evaluation, and Certification (Test/Cert) Framework tests and obtains certifications and authorizations for the core components and for any HW/SW added to the PLA to support Marine needs.

"Framework" is used to describe the Test/Certification approach because the approach is much more than instrumentation and software. Test/Cert includes policies, procedures, interfaces, and working relationships with Government and commercial agencies. The Test/Cert Framework contains specifications, standards, and DoD guidance for PISR systems. Test/Cert Framework is a physical and conceptual structure designed to make testing easier, automated, and repeatable.

The Test/Cert Framework tests and validates technical and functional capabilities of PISR System components. The Framework provides the data and reports necessary to obtain critical certifications to assure that PISR equipment being deployed to the warfighter meets current DoD guidance to be net-centric and interoperable. The Framework also provides data necessary to obtain authorizations to connect (ATC) and authorizations to operate (ATO) so the warfighter is assured that the new PISR components are secure, able to operate on classified networks, and cannot be exploited by the enemy.

Portions of the test framework are delivered with each PISR System to provide a streamlined, intuitive interface for the user to understand and maintain system readiness by identifying, troubleshooting, and resolving system problems.

9.2 Test and Evaluation Methodology

Three components are needed for the Test/Cert Framework: (1) something to test; (2) something to test with; and (3) something to test against. The Test/Cert Framework is used to test components of PISR Systems. Components are tested using a modified version of the Test/Cert Framework used by the Joint Interoperability Test Command (JITC) at Fort Huachuca, Arizona. Testing is accomplished against functional and technical PISR system specifications and DoD standards and specifications. The High-Level Test concept is shown in Figure 50.

JITC has a Test/Cert environment called the Open-Source Test Framework (OSTF). The OSTF is used to test new DoD systems for interoperability and network readiness, and to certify that IA requirements are met. RapidPro's Test/Cert Framework is a version of this COTS/GOTS system, tuned to the functional and technical requirements of the MCISR-E environment. JITC tests and certifications must satisfy a large number of DoD and commercial specifications. The PISR product line adopts some of these specifications and modifies others to meet the specific requirements of USMC PISR. Test tools, specifications, standards, and other components used to create the test harness for each PISR component may come from NPS, the system under test, JITC, or other commercial or Government sources. Test Tools at NPS are under configuration management through the RapidPro Lifecycle Management system (see Section 11).

The RapidPro Test/Cert Framework is closely integrated with JITC systems. This supports early engagement with JITC for certification of core PLA capabilities and new PISR capabilities that are fielded. New PISR products are developed using the core capabilities certified by JITC. Each new product starts with 85% to 90% of their system level certification requirements already met. Close integration with JITC supports obtaining the additional certifications quickly so new capability can be deployed rapidly to the USMC.

The Test/Cert Framework supports all components of the PISR PLA. The Test/Cert Framework includes hardware and software, sets of test and certification metrics, test instrumentation, and data gathering and reporting tools for implementation at the system integration laboratory at NPS. The Test/Cert Framework includes testing processes that use instrumentation and metrics to grade the ability of system components to meet Marine operational needs.

The PISR PLA is a foundation for producing fielded systems that can adapt to a variety of operational problems, including inadequate or insufficient resources and component failures. Each system iteration necessarily carries risks that need to be assessed in terms of their ability to adapt appropriately to the operational environment. The Test/Cert Framework measures performance and quantifies risks so appropriate officials can judge the value of the PISR components relative to mission needs and outcomes. There is always inherent uncertainty and lack of predictability in decisions made at the tactical edge under dynamic conditions. The Test/Cert Framework uses tools and processes to test and certify systems with the understanding that risk is managed in a manner that optimizes mission outcome. All PISR components enter the PLA using the process shown in Figure 50.

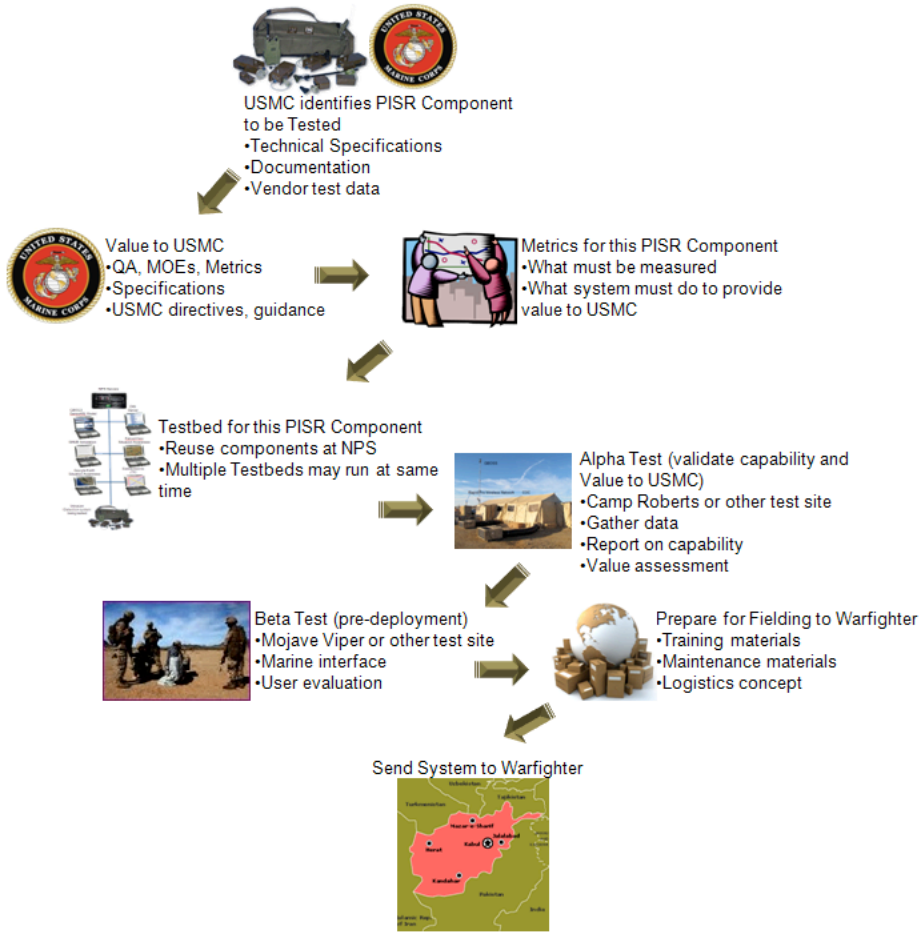

Figure 50. T&E Methodology: PISR components move through a series of structured activities to become ready to be used by the warfighter

The Test/Cert process starts with the USMC identifying a candidate PISR component that is expected to meet specific critical USMC needs. The USMC provides that PISR component to the RapidPro Team. The Team has an extensive list of Quality Attributes (QAs) that are vetted with USMC and form the baseline of understanding and assessing what capabilities provide value to the warfighter. The RapidPro Team takes functional and technical

information for the PISR components and uses the QAs to develop Measures of Effectiveness (MOEs) for this PISR component. MOEs tell the Team how the PISR component must act to be effective. Metrics are developed to address the MOEs so analysis can be performed to assure that PISR component is effective in providing expected value to the USMC. Metrics are used to identify what is must be measured and how it will be measured. Additional hardware and software components are combined to form a testbed at NPS for this PISR component. Multiple PISR component testbeds may be active at the same time.

The NPS Testbed is used to validate the capability of PISR components and to establish interfaces between the PISR component and other systems that are needed to provide a realistic operational environment. Live and constructive elements are pulled from the NPS T&E repository to build the testbed. When the PISR component is determined to be ready, the component and most of the testbed are moved to an Alpha test site, such as Camp Roberts or some other test site, to conduct field testing of the operational capabilities of the PISR Component. After review of the Alpha Test Report and approval to proceed by USMC, a Beta (pre-deployment) test site is chosen. Results of the Beta test indicate the PISR component's readiness to move to operational deployment. After additional work to develop training materials and make necessary refinements based on results of the Beta test, the PISR component is ready for transition to the warfighter.

9.3 Test/Cert Framework Functions

Testing is accomplished in three functional areas:

1. Verification and Validation (V&V): Does the system operate as advertised?
2. Technical Specifications and DoD guidance: Does the system meet the technical interface requirements of the PISR PLA?
3. Operational Effectiveness: Does the system provide value to USMC?

Certification is accomplished in two functional areas:

1. Interoperability: Can the system interoperate with existing DoD HW/SW?
2. Accreditation and Authorization: Can the system be certified and accredited to operate in a classified Marine Corps environment?

The Test/Cert Framework validates that when PISR system operators configure sensors, analytics, and people to support decisions, appropriate triggering, cueing, and alerting occur. MOEs are used to determine if PISR components are providing measurable improvement in support to Marine missions. Test plans identify specific instrumentation requirements and data collection requirements to quantify success, partial success, or failure.

9.4 Behaviors

The Test/Cert Framework exhibits the following behaviors:

- Operates in open standards, open-source environments that are agile and adaptive
- Emphasizes services rather than point-to-point connections
- Operates in a publish and subscribe environment
- Provides test and certification services as an integral part of design, development, and fielding of the PISR System
- Complies with DoD mandates and guidance as modified by and approved for the USMC Rapid Prototyping process
- Uses concurrent engineering and agile development processes to develop and deploy incremental capability
- Exploits modeling and simulation technologies
  - DoD certified models
  - Direct connection to government and contractor models

- Uses a continuous improvement process, resulting in a robust Test/Cert capability hosted at NPS and producing test components that can be fielded with PISR Systems
- Takes maximum advantage of off-the-shelf HW and SW
- Connects and federates with existing DoD Test/Cert capability at: JITC, Fort Huachuca; Integrated Team Solutions Facility (ITSFAC), Stafford, VA (Quantico); and the planned Information Assurance (IA) Test Range
- Develops test metrics that are mission-driven and support the smart push model embraced by the PISR PLA
- Develops certification metrics that are tuned to the specific environment where the system under test will be used
- Develops test tools to obtain data and develop reports
- Supports the collection of data from field exercises or appropriate simulators suitable for testing and verifying the PISR System's performance on identified key use cases
- Provides methods for PISR System developers to utilize test data in laboratory experiments routinely to support their testing, debugging, and evaluation of system configurations

9.5 Quality Attributes Derived for Test/Cert Framework

Quality Attributes (QAs) are developed for the Test/Cert Framework. MOEs are developed from the QAs and metrics are developed from the MOEs. These QAs, MOEs, and metrics apply to the core RapidPro components and to all PISR components that become part of the PLA. QAs, MOEs, and metrics identify what needs to be measured to provide value to the USMC. Using the same quality baseline for all PISR products supports rapid integration of new components into the core capability and assures that measurable value is delivered to the USMC.

Three tiers of MOEs are developed for RapidPro. Tier 1 addresses the effectiveness of RapidPro to perform testing of any PISR component. Tier 2 address the effectiveness of testing a specific PISR component. Tier 3, the most important, addresses mission effectiveness for expeditionary U.S. Marine forces. Satisfaction of Tier 3 MOEs provides great value to USMC and must track back to QAs and MOEs for the Test/Cert Framework. Tier 1 MOEs are identified in Appendix B, Tier 2 MOEs are provided in Appendix C, and Tier 3 MOEs are located in Appendix D.

The following Quality Attributes (QAs) are derived for the Test/Cert Framework:

- **Composability:** Capability to compose the elements of the desired test environment seamlessly by selecting and configuring live, virtual, and constructive components into a meaningful test environment.
- **Reusability and Persistence:** The test infrastructure persists over time and includes organized repositories to support the reuse of models and analytics.
- **Extensibility:** The test infrastructure can be efficiently extended through the use of common architecture, interfaces, processes, and tools.
- **Agility:** Ability to automatically and adaptively monitor and manage selective functioning of the test infrastructures, test scenarios, networks, and systems and services under test.
- **Automation:** Ability to continually enhance the degree of automation of all the processes involved in defining, implementing, managing, reusing, and executing test events. This includes automated self-organizing recognition, initialization, and control of plug-and-play test environment components.
- **Usability:** Ability to instrument test environments in a manner that is principally non-intrusive and highly embedded, which provides real-time measures at the system and system-of-system levels. Measurements are consistent and repeatable.
    - Capability to reproduce the test environment and play back segments of the test event that facilitates assessing the effects of modifying the experimental conditions with plug-and-play components.
    - Capabilities to measure, compare, and evaluate experimentally-specified architectural and parametric configurations of the system under test.
    - Capability to collect and segregate operational data
        - Red/Blue Data
        - White Data or Truth data from the Test/Cert Environment

- Capability to seamlessly switch between real-time and after-test analysis of collected data.
- Capability to perform asses overall net-readiness of components under test.

9.6 Test/Cert Framework Reference Implementation

An initial reference implementation serves as an example for the testing and certification of all PISR components that will join the RapidPro PLA. The sensor technology selected for this reference implementation is real although it is not currently being evaluated by the National Reconnaissance Office (NRO). The instrumentation and analysis tools referenced are real and are currently being used by NPS and JITC. The Use Case for this reference implementation is modified from the one developed for the Marine Corps TRSS test at Camp Roberts from 8 to 19 November, 2010 (refer to Appendix E for further information). Data connections between NPS and JITC do not currently exist but are under development. The schedules for Camp Roberts Tactical Network Topology (TNT) and Mojave Viper are fictional as are the results of the tests. The process to identify a critical PISR technology, mature and test that technology, and then deploy that technology to the warfighter involves the following fourteen (14) steps:

Step 1: PISR component is approved by USMC for rapid prototyping and fielding.
Step 2: Initial technical assessment of PISR component
Step 3: Use Case development
Step 4: Test Concept developed
Step 5: Metrics developed for this PISR component from the Rapid Pro value baseline
Step 6: Test instrumentation and test procedures identified
Step 7: Detailed Test Plan completed and approved by USMC
Step 8: PISR component testing at NPS
Step 9: PISR component Alpha testing at Camp Roberts (example Alpha test site)
Step 10: PISR component Alpha testing at Mojave Viper (example Beta test site)
Step 11: Interoperability Certification is obtained
Step 12: Authorization to Connect is obtained
Step 13: Documentation, training materials and logistics concept completed
Step 14: PISR component is ready for deployment to USMC selected site

A sample timeline for the Test/Cert process is shown in Figure 51.

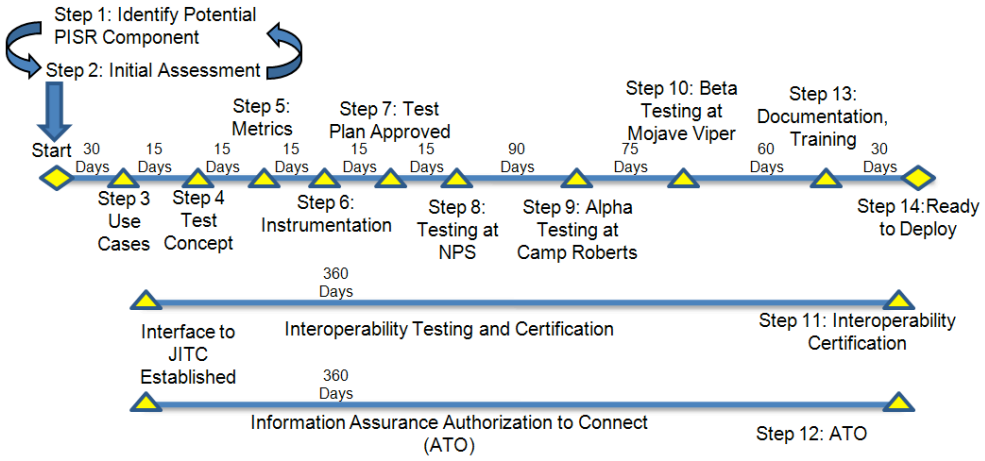

Figure 51. Sample timeline for fielding important PISR technology

The following discussion walks through each of the above steps in the Test/Cert process and provides an example of the work accomplished at each step.

Step 1: Process to field an important technology begins with a technology identified for potential fielding from Marine Corps Systems Command (MARCORSYSCOM), Naval Research Office (NRO), Los Alamos National Lab (LANL), or other Government source.

Activity: An unattended ground sensor using Radio Frequency Identification (RFID) has been evaluated by NRO. Intrusion detection sensors contain no batteries. They receive power from a Radio Frequency (RF) source that is located as far as 2km from the sensor. Sensors are activated at preprogrammed periods or when requested by a user. NRO indicates that the sensors have the following characteristics:

1. Do not require the sensor a battery at the sensor
2. A central RF power source radiates all sensors
3. RFID approach promises lower initial cost, quicker deployment, lower incandesce of false positives, and lower detection and destruction of sensors by enemy (sensors are only 5cm x 5cm x 1.5cm).
4. Sensors integrate seamlessly with field camera with slew and zoom capability.
5. System provides real time data analysis.

Step 2: Initial assessment of system using documentation provided by the Government Agency offering the technology and from the Vendor is performed to determine:

1. If the component will be able join the PISR PLA?
2. If it realistic to expect to obtain an interoperability certification and Authorization to Connect within 4 to 6 months?
3. Is the system mature enough to be able to field within 4 to 6 months?
4. Is there sufficient Government interest, documentation and vendor support to move into an Alpha – Beta testing?

Activity: Research of documents, phone calls, and physical inspection of the component indicates:

1. Component uses open published standards for interface to external systems.
    a. Cursor on Target (COT)
    b. Distributed Interactive Simulation (DIS)
    c. Published XML schema
    d. MySQL database with schema
2. Radiating device is human safe at highest power, providing intrusion detection at 2 km line-of-sight from transmitter.
3. Assessment indicates that system is a possible candidate for inclusion in the Rapid Pro PLA.
4. Sufficient interest and need exist to expedite this PISR component to the field for U.S. Marine use.

A report of this initial assessment is provided to MARCORSYSCOM with recommendations from the RapidPro Team. MARCORSYSCOM may approve this PISR component to move forward to be tested and fielded, they may ask for additional information, or they may terminate investigating this technology. The following assumes that this PISR component if approved for testing and deployment to the field.

> **Note:** Several potential PISR components may be going through Steps 1 and 2 at the same time. Test programs are designed to share resources. Objective capability is for 3 to 4 potential PISR components to enter the process each year and 3-4 PISR products to graduate the process and proceed to fielding with the warfighter each year.

Step 3: Use Cases are developed and vetted with U.S. Marine stakeholders. Use Cases identify activities necessary to test the system. An important part of Use Case methodology is agreement from Stakeholders on assumptions, preconditions and postpositions. Working closely with stakeholders to create Use Cases helps assure that the Rapid Pro Team understands important USMC needs and the critical Marine need being addressed by the PISR component being tested. Use Cases do not include implementation-specific activities or details regarding interfaces between this component and other components and users. Please see Appendix E for a sample Use Case.

Activity: Use Cases are developed.

Note: Step 4 begins a series of activities that occur in parallel leading to the Alpha test of the PISR component at Camp Roberts. From Step 4 until the fielding of the PISR component, NPS, JITC, NSA, NRO and MARCORSYSCOM work closely together to mature and field the PISR component. The following steps will be discussed individually, but they occur in parallel:

Step 4. Test concept developed
Step 5. Develop metrics for PISR component test
Step 6. Test Instrumentation, standards, procedures identified.
Step 7. Test Plan completed and approved
Step 8. System tested in NPS lab
Step 9. System tested at Camp Roberts (Alpha Test)

Step 4: Test concept is developed by working closely with USMC to identify and document operational scenarios that can be served by the PISR component being tested. Rapid Pro has 3 USMC approved scenarios as our baseline. These are; (1) Sneak attack, ambush, (2) Improvised Explosive Device (IED) detection, and (3) Identification of High Value Individual (HVI). The following discussion provides an example of how steps 4, 5, 6 and 7 are accomplished.

Sneak attack/ambush scenario: Two or more people are approaching the FOB from the Northwest. There path will cross a field populated with the intrusion detection PISR components being tested. PISR components are deployed 1.5 km from Combat Operations Center (COC). The FOB has a Ground Based Operational Surveillance System (GBOSS) tower located at the COC. GBOSS and the PISR component are reporting through their own ground stations and through the Geospatial Hub (GHub). PISR system has a standing request to slew GBOSS sensors toward any remote sensor that indicates an intrusion has occurred.

Possible tactical significance:

1. RPG-7 family of rocket propelled grenades:
    a. 500 meters max effective range for stationary target
    b. 400 meters max effective range for moving targets
    c. 920 meters max range for nuisance attack
2. Sniper
    a. 1.5 km max effective range
    b. 650 m normal engagement distance

Step 5: Metrics are developed for the PISR Component from the Rapid Pro Measures of Effectiveness (MOEs). Rapid Pro has three tiers of MOEs. Tier 1 (Appendix B) was used to develop metrics to assess the readiness of the Rapid Pro project to perform Alpha and Beta testing of PISR components. Tier 2 MOEs (Appendix C) address all PISR components that can be part of the PLA. Tier 3 MOEs (Appendix D) address USMC mission needs. Metrics are used to identify what needs to be measured and how it will be measured. Single MOEs are shown for this example although all MOEs are assessed for each PISR component. Additional MOEs and metrics are developed to verify and validate that the PISR component being tested is performing as designed and advertised.

Tier 2 MOE: Automatically collected test data is sufficient to perform validation and verification of the operational capability of PISR products being tested.

a. % of data collected automatically
b. Accuracy of independent measure of intrusion event is greater than PISR component measurement
c. % of intrusion activities captured

Tier 3 MOE: Increase Situational Awareness through expanded Common Operating Picture.
Metrics

a. % increase in surveillance area
b. % increase in combat reach
c. extent to which tactical-level coordination is improved
d. number of essential situational awareness activities being performed by the PISR System

Intrusion detection PISR component reports anytime an object passes within 10 meters of a device. The devices survey themselves based on the known location of a single device. Devices automatically report their ID#, the time of the detection (GPS time) and their location.

Step 6: Test instrumentation and test procedures are determined from the scenarios and metrics. Additional analysis of the scenario is necessary to quantify activities and select instrumentation. Metrics for both Tier 2 and Tier 3 MOEs require measures of the location of the intrusion detection devices, measure of the track of individuals of vehicles performing the intrusion and identification and timing of decision made in the COC.

Additional analysis includes:

1. RPG-7 family of rocket propelled grenades
    a. 500 meter effective range
    b. Detected at 1500 meters from COC
    c. Assume 4 km per hour movement of enemy (fast walk) as max speed
    d. Assume 2.5 km per hour movement of enemy (slow walk) as minimum speed
    e. Assume 3 minutes to take position and fire RPG
    f. Must identify enemy activity and take action within 20 minutes (Key Performance Parameter (KPP) threshold)
        i. User has 1.1 km of travel at 4 km/minute plus 3 minutes
    g. Must identify enemy activity and take action within 30 minutes (KPP objective)
        i. User has 1 km of travel at 2.5 km per hour plus 3 minutes

1. Sniper
    a. 650 meters effective range
    b. Detected at 1500 meters from COC
    c. Assume 4 km per hour movement of enemy (fast walk) as max speed
    d. Assume 2.5 km per hour movement of enemy (slow walk) as minimum speed
    e. Assume 5 minutes to take position and fire sniper rifle
    f. Must identify enemy activity and take action within 25 minutes (KPP threshold)
    g. Must identify enemy activity and take action within 16 minutes (KPP objective)

Data Collection:

1. Independent measurement of time enemy crosses the 1.5 km line
2. Independent measurement of location of all personnel or vehicles performing the intrusion
3. Time sensor system indicates enemy crosses the 1.5 km line
4. Time user notices that an alert has been given by the PISR component
5. Time enemy stops at 500 meter line
6. Time for enemy to get ready to fire the RPG
7. Time GBOSS sensor slews to intrusion detection alert position
8. Time user determines that intrusion is a threat
9. Time user initiates action to counter the threat
10. Time when response is available
11. Time when threat is eliminated

Instrumentation:

1. Observer in field to follow enemy activity with video cameras positioned to do the same
2. Observer in the COC to monitor the PISR component ground station for alerts
3. Observer in the COC to monitor GBOSS activity
4. Observer in COC to monitor and record activity and decisions made by COC personnel
5. GPS equipped red personnel and vehicles that will perform the intrusion
6. GPS equipped blue forces that will eliminate the threat
7. Interface to GHub to record all activity.
8. Interface to PISR component ground station to record all activity.

Value Assessment:

a. Damage to the FOB if enemy is engaged before they begin their attack versus damage to the FOB if engagement is started after the attack begins.
b. Time to engage enemy with the PISR component alert versus time to engage without the alert.

Step 7: All information is now available to complete a detailed Test Plan and get that Test Plan approved by USMC.

Step 8: Component testing and software development has been underway at NPS since evaluation of this PISR component began. Figure 52 presents a block diagram of a typical testbed setup at NPS. Models and Simulations are mixed with real hardware to provide a realistic test environment. Live connects to JITC support performing interoperability analysis during the Alpha test phase. NPS is now ready to complete the hardware, software and simulation environments to perform testing at NPS. Additional work includes:

a. PISR component is modeled using SensorML. SensorML is an open source tool, supported by the Open Geospatial Consortium (OGC) used to model sensors. Numerous sensor models are available from OGC, including intrusion detection sensors.
b. Sensor Web Enablement (SWE) is used to manage the connection of the PISR component and GBOSS.
c. The following components are borrowed from the NPS Test Tool Repository (Figure 40)
    i. GBOSS optical sensors SensorML models
    ii. GHub simulation
    iii. Cursor on Target (COT) server
    iv. Distributed Interactive Simulation (DIS) service interface
    v. Falcon View Situation Awareness (SA) components
    vi. Google Earth SA component
    vii. Service interface to JITC's OSTF
    viii. Service interface to USMC Information Assurance personnel

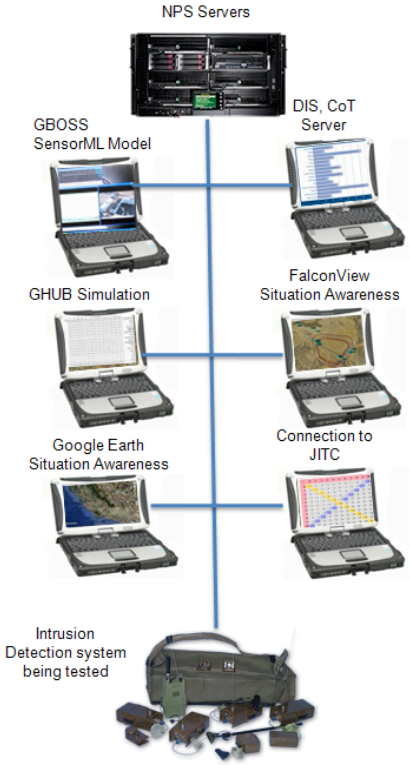

Figure 52. Reuse of components within the Test/Cert Framework reduces cost and risk

Step 9: PISR components are now ready to go through an Alpha Test at Camp Roberts or other test site. Test/Cert components integrated together at NPS are moved to Camp Roberts. Baseline Test/Cert capability remains active at NPS and is connected to test site. NPS has all of the hardware and software necessary to test in the field. Core RapidPro capability is augmented with live hardware and constructive simulations to provide a realistic environment for testing. Combat Operations Center (COC) is established to control all activities during the test. Figure 53 presents a typical setup with the COC and elements of the Rapid Pro wireless network being used to support testing. Figure 54 presents a view inside the COC where NPS and other personnel conduct the test and manage rest resources. All these resources are reusable from test to test. Figure 55 presents FalconView being used for situational awareness (SA). Figure 56 shows Google Earth being used for SA. FalconView was chosen as the primary SA display because it is used throughout the USMC and in various programs in DoD. FalconView also brings with it the Cursor on Target (COT) message protocol. COT is used by numerous DoD programs including TRSS, C2PC, and CPOF.

Figure 53. RapidPro test resources at Camp Roberts

Figure 54. Activities inside the COC

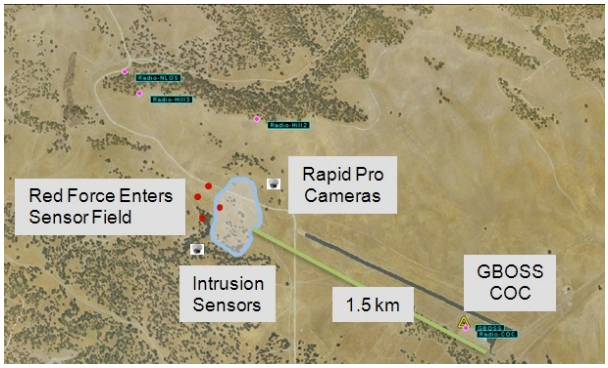

Figure 55. FalconView used for situational awareness

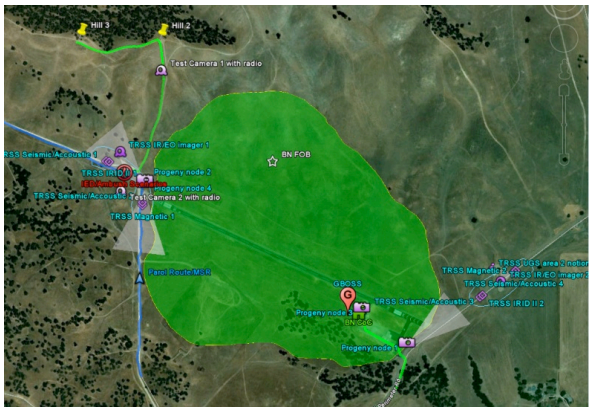

Figure 56. Google Earth used for SA

Step 10: Beta testing or pre-deployment testing is now possible. Shortfalls in system performance and user interface identified during the Alpha tests are addressed and the PISR component is made ready to move to Mohave Viper or some other pre-deployment site. Beta testing is the first time that a broad spectrum of Marine users is exposed to the PISR component. Important data about usability and supportability are gathered to prepare the PISR component for transition to the Warfighter.

Step 11: Interoperability Certification is obtained from JITC. At this point in the process, JITC has been working with the Rapid Pro Team for 6 months to gather the data necessary to assess the risk of allowing this PISR component to interoperate with other components within the Warfighter's environment. Interoperability certification is accompanied by a risk analysis detailing what the User can expect after this component is added to the fielded ISR components.

Step 12: Authorization to Connect is obtained with additional risk assessments. A PISR component that moves from COTS to a fielded capability in 6 months will not meet all of the Information Assurance and net-readiness requirements of the DoD. The PISR component will however, meet those critical requirements to add value to the Warfighter without adding additional risk.

Step 13: Documentation, training materials and logistics support concepts are completed so that the PISR component can be integrated smoothly into the operational environment of its destination.

Step 14: PISR component is ready for deployment to USMC selected site. Receiving command can be sure that the PISR component operates as expected, that it is supportable, and that it will add value to USMC intelligence operations.

9.7 PISR Subsystem Support to the Test/Cert Framework

The User Interface Environment Subsystem supports automated tests with and without user input being required. An automated test framework will replace the PISR thin client to exercise the PISR System with predefined tests. This will have extra displays that indicate what data is flowing within the PISR subsystems. Only the relatively simple thin client code within the web browsers are not tested with this approach. That code will be tested by recorded scripts of mouse movements and button clicks carried out on the web browser itself. Each of the UI services will provide test options in their configuration to allow outputting in verbose mode all their inputs and outputs into log files.

The SA Subsystem supports test and evaluation by using standardized interfaces connecting loosely coupled components at its boundaries. Internally, since most components connect through the PISR IB blackboard and the generation of hypotheses is monotonically increasing, the system already records much of the output needed for testing. Sample hypotheses feeds can also be injected into the SA subsystem via a test sensor integration interface that replays example test data.

The MCL Subsystem provides automated tests and extensive logging of information relating to the various control and optimization features provided. Optimization computations are fully supported by data and logic trace-back to allow cross-check of results and system decisions/adjustments resulting from those computations. By its nature, the MCL Subsystem is tightly interwoven with the Test/Cert Framework as means to monitor and assess system processing.

The PISR IB subsystem provides data storage and access to stored data for the Test/Cert Framework. Connection points are provided for all data flows into and out of the PISR IB, both for internal and external data sources, to allow the Test/Cert Framework to monitor all information flows into and out of the PISR System. This is also true of the Dissemination Subsystem, enabling the Test/Cert Framework to monitor and track the application of dissemination policies to recipients of data from the system.

This page intentionally left blank.

10 Information Assurance (IA) Framework

10.1 Background

The purpose of this section is to an RapidPro Security Architecture consistent with the objectives of the PISR PLA. This includes identifying all of the security requirements for information assurance at the architecture level. Because the intention of identifying architecture level requirements is to achieve a level of abstraction which does not constrain implementation, no specific implementation details, including specific products, are either recommended or required in this discussion. However, in some cases, particularly because of their current certification and accreditation (C&A) status, some specific operating systems or type of operating system are identified. The intent in doing this is not to limit a developer, but to highlight where C&A costs have already been incurred, and to recognize when developers will bear more of the burden of the C&A costs from using another operating system or even developing their own. In other instances, specific products may be identified as examples of implementation to facilitate understanding; for example, employing an Apache Tomcat plug-in. It is not, however, the intention of this architecture document to insist on any specific implementation details.

This section of the PISR PLA document is concerned with the IA requirements and technologies as related primarily to the Application Platform, where the Application Platform provisions the local computing power for the system. Therefore, the Application Platform includes the computing infrastructure hardware and hardware abstraction (processors, memory, and interconnection), the computing infrastructure, the partitioning, the operating system services, a multilevel security operating environment (when required for Cross Domain Solutions), and middleware and services supporting the RapidPro general applications.

The IA requirements for the Platform External Environment are not included in the scope of this document. The Platform External Environment includes, for example, the Human Computer Interface (HCI), the individual sensors, and the external information services and communications associated with existing networks, such as SIPRNET and JWICS.

10.2 General Requirements

The best known requirement for security and assurance is for separation. Separation is typically based on domains, where domains are identified by access security levels. There are three security levels: TOP SECRET, SECRET, and UNCLASSIFIED. While there can be, and usually are, many caveats in each of these domains, and some additional levels of Discretionary Access Control (DAC) are required, for example between SECRET and SECRET (5 eyes), the levels of robustness required for this level of separation are generally not as robust as for the primary levels of separation. These three have a single level of separation between them, so for a system which spans from TOP SECRET to UNCLASSIFIED, two levels of separation are required. The target environment for the RapidPro project will require two levels of separation. In the RapidPro project, additional controls for separation, especially among coalition users, will also be required.

Additionally, as identified in the National Security Agency (NSA) Global Information Grid (GIG) IA Roadmap, and other NSA documents, there is an access property called Risk Adaptive Access Control (RAdAC). Traditional systems, which are based on Mandatory Access Controls (MAC) and traditionally run at System High, typically are very brittle and do not support the dynamic capabilities required for RAdAC in which unexpected changes in the operational environment may require changes in permissions or access, especially in order to save lives. A simple example of a use case is where enemy preparation for a coalition convoy ambush is detected using NTM, and this information must be made available to the convoy in time for action to reduce the risk of loss of life. Such a system must be designed and architected in such a manner as to facilitate this requirement. Because each PISR System will be used in theater, often with other coalition users, it has a requirement for such a capability, particularly across the previously discussed enclave domains.

10.3 Information Assurance Concerns

IA and/or security concerns are broadly and often not consistently identified. Although some specific concerns must be addressed in specific testing and certification, for example those in CJCSI 6212.01E, all concerns have to be addressed eventually. One general way to identify all of them is to take the security requirements addressed by the

NCES Security Architecture, which includes five primary tenets of Information Assurance: Confidentiality, Integrity, Authentication, Non-repudiation, and Availability. In addition, the NCES document identifies the following additional security requirements that need to be addressed:

- **Manageability** – The security architecture should provide management capabilities for the security functions. These may include, but are not limited to, credential management, user management, and access control policy management.
- **Accountability** – This includes secure logging and auditing which is also required to support non-repudiation claims.
- **Security Across Trust Domains** – The architecture must provide a trust model under which Web Service invocations across different trust domains can be secured, just like those within a single trust domain. All basic security requirements apply to cross-trust domain service invocations. Additionally, such invocations must be controlled by the local security policies of participating domains.
- **Interoperability** – Interoperability is the cornerstone of Service Oriented Architectures (SOAs), and the security architecture must preserve this to the maximum extent possible. Major security integration points in the architecture—such as those between service consumers and service providers, between service providers and the security infrastructure, and between security infrastructures in different trust domains—must have stable, consistent interfaces based on widely adopted industry and government standards. These interfaces enable each domain or organization to implement its own market-driven solution while maintaining effective interoperability.
- **Modeling tailored constraints in security policies** – In a traditional security domain, resources and services are often protected by a uniform set of security rules that are not granular enough to meet specific application needs. Under a SOA, service provider requirements may vary in terms of how they need to be protected. For example, one service may require X.509 certificate-based authentication, whereas another service may only need username/password authentication. Furthermore, because clients that access a resource may or may not be from the local domain, different "strengths" of authentication and access control may be required. Consequently, security policies must be expressive and flexible enough to be tailored according to Quality of Protection (QoP) parameters and user attributes.
- **Allowing Integration with existing IA solutions, products, and policies –** The SOA-based security architecture does not intend to replace an existing investment in security infrastructure. On the contrary, a flexible IA solution should be designed to leverage existing IT investments without causing any redundant development efforts. Seamless integration with existing security tools and applications also increases the overall stability of the enterprise.
- **Securing other infrastructure services within the SOA**, such as discovery, messaging, mediation, and service management.
- **Unobtrusiveness –** The architecture should be unobtrusive to other service implementations.

10.4 C&A and Operational Requirements

Several policies pertain to C&A to meet operational requirements; including:

- RapidPro Wide Area Surveillance CONOPS
- DoD Instruction (DODI) 8500.01E (DoD Information Assurance Certification and Accreditation Process [DIACAP])
- ICD 503 (Committee on National Security Systems Instruction [CNSSI] 1253) replaces Director of Central Intelligence Directive (DCID) 6/3
- Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 6212.01E
- Unified Cross-Domain Management Office (UCDMO) approval for CDS solutions
- Common Criteria
- DO-178B Level A

These requirements, except for the CONOPS, focus primarily on the C&A needs for the system to achieve an Authority to Operate (ATO) or an Interim Authority to Operate (IATO). In addition, the CJSCI focuses primarily on interoperability requirements, which are identified by the Net Ready Key Performance Parameters (NR-KPPs). For IA

interoperability, these include (i) authentication; (ii) integrity; (iii) confidentiality; (iv) non-repudiation; and (v) availability. The RapidPro project must meet all of these requirements and, to the extent possible, it is the intention of the architecture that these capabilities be "baked-in." In other words, they are identified, designed, and delivered from the very first builds, rather than through some effort to add them on at the end. It is virtually impossible to add adequate security into the system at some later date if it is not provisioned from the beginning. Without this early provisioning, the RapidPro project will not be able to achieve an adequate level of security. Therefore, the details of what is required for a successful C&A and interoperability will be among the first requirements identified, with particular concern to mitigating any risks assumed. For example, if source code is required as an artifact for a completed ATO, and a commercial product is selected for which no source is available, then the risk mitigation strategy will be identified and managed from the beginning of the selection of the product.

JITC also tests and certifies the IA implementation for interoperability. For IA controls, this was introduced to the CJSCI document 6212.01E on 17 Dec 2008. Of special concern to JITC is the interoperability of the IA capability, so that the end result is data and resource exchanges which are not only interoperable, but also trustworthy. As a somewhat unique requirement for the RapidPro project, it is sometimes the case that the level of trustworthiness for the transaction may have to be identified. Thus it may be possible that a transaction may be completed, but not trusted. Or, it may be trusted a small amount, but not completely.

The UCDMO was established in July 2006. All DoD and Intelligence Community (IC) cross domain efforts now fall under the jurisdiction of the UCDMO. The UCDML establishes and maintains a baseline of approved and recommended CDS. The UCDMO Baseline is a re-use list of technologies and solutions that are available to Agencies in the DoD and IC. It is a list of cross domain technologies that are already in place somewhere, have a government sponsor, and at least a three-year lifecycle support agreement. Ideally, the solution required for CDS by the RapidPro project would be already on the baseline list so that acquisition, integration, certification, and accreditation would be the only steps remaining to field the solution. Unfortunately, however, this is not the case. None of the current products on the UCDMO baseline list meet RapidPro PISR System requirements. There are two primary reasons. As noted previously, the RapidPro requirement is for both access and transfer across two levels of domain separation, meaning, for example, from TOP SECRET to UNCLASSIFIED. Current approved solutions allow for only a single level of separation. Thus, in order to meet the two levels of separation requirement using existing solutions, two separate architectures would have to be fielded, one for each level of separation. Typically, this could mean the equivalent of somewhere between six and ten TSOL 8 boxes, or equivalent. The Space Weight and Power (SWAP), as well as the maintenance requirements for such an approach, exceed the capabilities of the target environment of the PISR System (battalion and below). In addition, because these solutions rely on traditional guard-like approaches, meaning that they generally have components such as a communication handler, a filter orchestrator, the actual filters, and the cross domain component, their security attack surface is extensive. Additionally, because of the many millions of lines of code involved, their flexibility is much diminished. If there is a need to change access control privileges because of some operational or environmental need, such as a new coalition partner joining the enclave, changes and recertification of the system can take many days. A PISR System Quality Attribute is for it to take only minutes to accomplish this task.

As a result, RapidPro has taken a much lighter weight approach using Separation Architectures which block access and a policy engine which permits changes in privileges based on attributes. This approach is based on the NSA High Assurance Platform (HASP) architecture, which in turn is based on a separation architecture using virtualization. This solution will have to be briefed to the UCDMO and an endorsement of the architecture, or recommendations for a better approach, will have to be obtained so that in the end, prior to the final IATO, the solution will be on, or recommended for, the UCDMO baseline.

Other methods used to achieve certification include the Common Criteria and DO-178B. The Common Criteria are used to achieve an Evaluation Assurance Level (EAL) against some Protection Profile (PP). This evaluation is done by an independent lab, called a NIAP lab, which is run under the auspices of the NSA. While EAL evaluations for 3 and less have been done for years by NIAP labs, the criteria for 3 and less, and most often 4, is determined by NIST. EAL 3 evaluations are inadequate for a single level of separation CDS, while in many cases EAL 4 solutions can be used for a single level. For higher levels, NSA establishes the criteria and at this time, there is some turbulence at NSA as to exactly the value of the evaluation. In general, the main problem with the approach is that while the EAL criteria will yield a good result and analysis of the specific component identified by the PP, interactions between that component and the rest of the system are not addressed and hence effects are largely unknown using this method. Therefore, risks can be inadequately identified and risk mitigations incorrect. From an architecture point of view, this suggests that investment in further EAL may be misplaced if other ways to identify the risk and mitigations are not

concurrently completed. However, if some component or capability which can be consumed by RapidPro for IA is available, by all means, the EAL arguments can be included as well.

In a similar manner, albeit with more success, DO-178B is used for safety (of flight mostly) software. It is generally thought that there are many similarities in the assurance arguments and proofs for EAL 6+ and DO-178B Level A. There are also, of course, differences, one of the most notable being that by and large safety processes are transitive whereas security processes are not. As with EAL, while no direct investment in achieving this level of certification for RapidPro is justified, if components which can be integrated into the RapidPro solution already have such a certification, then by all means, we should expect to inherit the assurances.

10.5 PISR IA Architecture Objectives and Goals

The RapidPro Application Platform will constructed with the following high level objectives and goals. Although to meet individual Certification and Accreditation the DCID 6/3 or DITSCAP may be used, the DCID has actually been canceled and replaced by the ICD 503. The ICD indicates that the security controls can be determined using the CNSSI 1253, where the general set of controls are as identified in NIST SP 800-53 (Recommended Security Controls for Federal Information Systems and Organizations). It is anticipated that the objective IA levels for the RapidPro system, when fully developed, integrated, deployed, and certified and accredited will be:

- confidentiality high
- integrity high
- availability moderate
- mission criticality high

The final determination of the IA category levels may be revised by the RapidPro Architecture in accordance with the policies of the CNSSI 1199 and as required by the system owner. Although the DoD is the primary customer for the RapidPro, the system may be required to support other non-DoD national interests, to include their specific C&A and interoperability requirements.

The primary goals for the RapidPro IA Architecture are as follows:

1. The RapidPro IA Architecture should facilitate an integrated solution across heterogeneous PISR systems and sensors.
2. The RapidPro IA Architecture shall promote interoperability across security domains, sensors, and joint service needs, both as a resource provider and as a resource consumer.
3. The RapidPro IA Architecture shall facilitate incorporation of changes due to advancements in technology and revisions of policies and instructions and support of the PLA.
4. The RapidPro IA Architecture shall facilitate the development of emergency CONOPS, sensors, and networks within the integrated IA Architecture using the PLA.
5. The RapidPro IA Architecture shall identify information flows using DODAF for analyzing security requirements and developing security policies and to support identification of where in the architecture security services are instantiated.
6. The RPB IA Architecture must also be adequately described so that it can be effectively understood, supported, and implemented using the PLA by the RapidPro stakeholders.
7. The RapidPro IA Architecture shall support an adaptive and dynamic modular certification and accreditation model with reciprocity for both security and interoperability accreditation.
8. The RapidPro IA Architecture shall provide the ability to process, access, and transfer data across multiple security domains at different classification and releasability levels from TS SCI to UNCLASSIFIED to support interoperability.
9. The RapidPro IA Architecture shall use standards and standardization for the implementation of data and security tagging across sensors and sensor systems and other RapidPro resources so that data can be easily shared among warfighters.

These IA goals may be revised as required by the RapidPro system and the PLA. The minimum IA properties that the Application Platform shall meet for interoperability are as specified by the NR-KPPs for the system and in 6212.01E: Availability, Integrity, Confidentiality, Authentication and Non-repudiation.

10.5.1 PISR IA Architecture

In order to meet the requirements for the RapidPro IA interoperability and CDS, while at the same time being tactical in footprint and dynamic, the RapidPro approach is logical separation. To keep security domains separate and IA functions separated, there are three separate approaches: temporal separation, physical separation, and logical separation. All currently certified solutions use one or more of these methods. Temporal and spatial, or physical separation, will not meet the RapidPro requirements for a number of reasons. Although either one or both can be used to meet the required two levels of separation, neither can be used to meet the CDS requirement. Additionally, with temporal, often called "periods processing," the identified resources are used for one period, or domain. When that mission is completed, then the components are scrubbed or physically switched out and the system is brought up in another period. Since the periods are not concurrent, no communication among them is possible. It addition, for a typical 25 workstation environment, such a method of change can take many hours for each change in period. With physical separation, a complete and separate environment is brought for each period. Typically, this results in each participant being required to have several workstations, one for each domain, or a selection of key variable management (KVM) switches. Unlike the periods processing solution, it is possible to access any or all of the domains at the same time, but the SWAP considerations are considerable. Additionally, in order to provide for transfer, a UCDMO approved solution is required between each separate domain, including, for example between SECRET and SECRET (Rel). Thus, in addition to the already discussed CDS hardware and software, this solution would also require a separate hardware suite for each domain required. For example, between TOP SECRET and UNCLASSIFIED, it would require three times as much hardware as any one of the domains. Further, for the reasons previously discussed, these solutions are not dynamic.

The logical solution is based on virtualization. Simply put, it depends on separating the supervisor mode from the user mode at the operating system level. This has been accomplished a number of ways. In the NetTop® project, the NSA succeeded in achieving it using Linux with some Security Extensions (SE) and additional host policies with a hypervisor, made using VMware. Additional methods involve the use of a separation kernel and both Type 1 and Type 2 hypervisors. This method is used both to meet the needs of the IA interoperability as well as the CDS service. To keep the security domains separated from each other, as well as to isolate the additional IA services which may require higher level of assurance, logical separation will be used. In general, the sense and detect segments of the RapidPro architecture are typically implemented in a mobile tactical environment at a location called a FOB. The system, and its sensors, can be deployed as a single operational node; or can be distributed across multiple nodes through a network, which are individual to the Application Platform, are based on external communications, or are both. The Application Platform therefore, provisions both the sensor services and any local IA services.

The technical basis for logical separation is based on Multiple Independent Levels of Safety/Security (MILS). MILS is an implementation architecture for high assurance software based on abstracting the privilege mode from the user mode in the operating system. Current implementations of this are accomplished through use of a separation kernel. The overarching goal of MILS is to increase capabilities for the warfighter by dramatically decreasing the time and cost of developing, evaluating, certifying, and accrediting multilevel secure systems throughout the multi-decade life cycle. The overarching goal of MILS is to increase capabilities for the warfighter by decreasing the time and cost of developing, evaluating, certifying and accrediting multilevel security systems. It decomposes complex systems into components, each meeting well-defined security requirements.

- MILS restricts highly security-critical code to very small components (kernels) that can be proven to meet well-defined security requirements at very high levels of assurance.
- MILS composes the system in a methodical way so the system can also be shown to meet system-level security requirements.
- MILS distributes security policy for an entire system among the components, each component responsible for only its security policy.
- MILS aligns the system architecture and assurance case to reduce security evaluation cost throughout the life cycle.

10.5.2 General Virtual Architecture, the Open System Environment (OSE) Reference Model

One approach permits logical partitioning such as that provided by the IEEE POSIX® Open System Environment (OSE) Reference Model (OSE/RM), as shown in Figure 57. Logical partitioning can offer affordable solutions and enable application to be collapsed into a single computer environment and network, including safety-critical and security-critical applications.

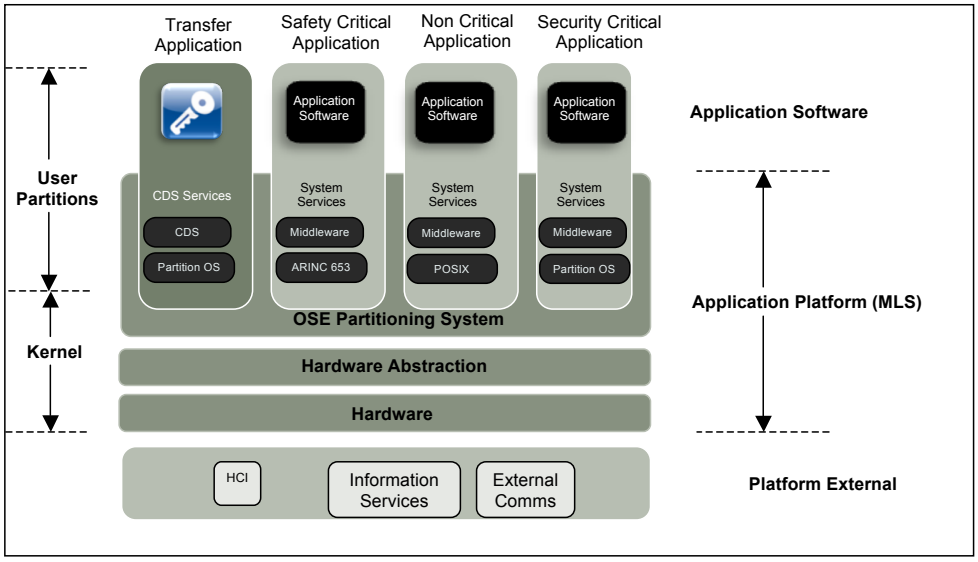

Figure 57. OSE profile for virtual separation with a Cross Domain Solutions (CDS)

A partitioning architecture permits one or more applications to execute on a single target computer. An Operating System controls the Application Platform hardware (through the hardware abstraction layer) and apportions resources between Applications in accordance with configuration tables. The Operating System uses memory management capabilities of the underlying hardware to map the physical memory into partitions. The memory maps are decoded and translated dynamically to offer each partition a virtual address space into which Applications may be loaded and run. Each Application operates in its own virtual memory space which is active while the Application is running. The time each Application is allocated to run is controlled and apportioned by the Operating System in accordance with schedule tables. External events such as interrupts and exceptions are translated by the Operating System and delivered to the Application only while the Application is running, and these are delivered as pseudo interrupts. Input and output of data are controlled, and receipt and delivery of information to/from an Application are only permitted while the Application is running. In effect, a partitioned system provides an execution environment for an Application that is equivalent to a target computer, implemented as a virtual machine that only gets a configured share of the resources of the physical hardware.

10.5.3 ARINC 653, another virtualization

ARINC 653 (see Figure 58) provides a specification that is used commonly on avionics platforms and on some UAS systems. It describes a partitioning system as well as an Operating System that schedules processes within a partition. The implementation choices are not prescribed, but the presence of two Operating System interfaces are: the Module Operating System (MOS) performs scheduling of the partitions and the Partition Operating System (POS) performs the scheduling of processes within a partition. It is possible to implement this with one or two actual schedulers; the specification does not prescribe which is more appropriate.

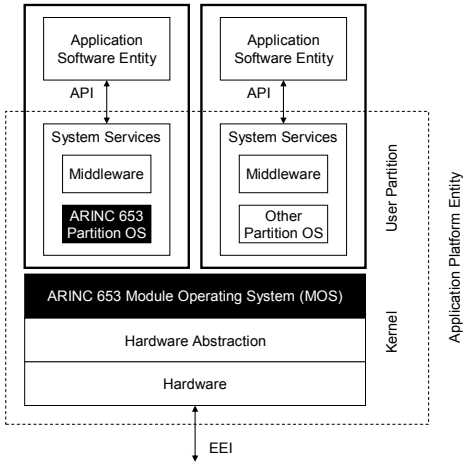

Figure 58. ARINC 653 Module Operating System

The MOS schedule consists of a list of partitions; their duration times may include other parameters which may be useful to synchronize the execution of applications. The list is repeated continuously. It may be possible to switch schedules by switching to a new list. This mechanism provides the capability to apportion different partition durations while the system is initializing itself, or if a different schedule is selected to disable some partitions and enable others or change timing, perhaps in response to a mode change in support of "battle mode" vs. "civilian mode", damage control, or emergency conditions. Information sent to a partition or received by a partition must be transferred using the designated data ports that are interconnected to form a data channel. The data ports are configured and connected using configuration tables and information will be transferred such that one channel will not affect any other data transfers.

10.5.4 MILS Separation Kernels

As introduced earlier, MILS is a security architecture concept for information processing systems that may be implemented in hardware through separate computers, through physical means on a single computer such as field-Programmable Gate Arrays (FPGAs), through software state machines, through a *separation kernel* on a single computer, through multi-core processors, or through other means. The overarching goal of MILS is to decrease the time and cost of developing, evaluating, certifying, and accrediting multilevel secure systems throughout the system life cycle.

The core concepts of MILS stated briefly are:

- **Divide and conquer:**
  - Decomposition of complex systems into components, each meeting well-defined security requirements (which may range from none to very high);
  - An emphasis on restricting highly security-critical code to very small components that can be shown to meet well-defined security requirements to very high levels of assurance ("evaluation");

- Composition of a system from components in a methodical way so that the system can be shown to meet system-level security requirements[37];

- **Distribution of security policy** for an entire system among the components, with each component responsible for only its security policy;

- **Alignment of system architecture and the assurance case** in order to reduce security evaluation cost throughout the life cycle of a system (i.e. decades for UAS systems).

The most active area of development for MILS-based systems is through software implementation of *separation kernels* based on the U.S. Government "Protection Profile for Separation Kernels in Environments Requiring High Robustness", Version 1.03 (see Figure 59). A *protection profile* is an implementation-independent set of information technology security requirements for a category of devices that meet specific consumer needs. The concept is central to the security evaluation scheme used for such real time operating systems that of the Common Criteria for Information Technology Security Evaluation, an international standard to evaluate the security of any IT product. This PP can, and has been used, by vendors to achieve EAL 6+ for commercial separation kernels.

The Separation Kernel Protection Profile (SKPP) provides specifications for partitioning in a system based on the MILS architecture that are in principle very similar to an ARINC 653 system, but in practice their implementations are different because of the need to both ensure security in the SKPP-based system; and yet also provide mechanisms for secure inter-partition communication and for development of high assurance user applications. In both cases, the kernel is responsible for time and space partitioning (scheduling and memory management of partitions). The hardware abstraction layer (device drivers) will tend to be *virtualized* and operate in polled mode rather than be interrupt driven (because in most cases it would be impossible to maintain strict control over schedule in the presence of interrupt-driven drivers). Drivers in most existent ARINC 653 systems, however, reside in kernel space (that is, they operate in supervisor mode), while in an SKPP-based system drivers will generally reside in user space (operate in user mode) to dramatically decrease the amount of code that must be evaluated to high assurance.

Other features of commercial separation kernel offerings include:

- *Trusted initialization* to ensure that the system begins operation of Applications in a secure initial state;

- *Trusted recovery* to ensure that a secure state is restored upon recovery from a fault;

- *Trusted delivery* to ensure that the code of the separation kernel delivered to a customer is the code that was evaluated and certified;

- Some mechanism for secure inter-partition communication, perhaps by changing page tables for performance ("zero copy");

- Provisions for running a variety of Partition Operating Systems to serve as a base for Applications, including traditional RTOS, Linux (in various flavors) and Windows;

- Provision for a "bare metal" or "minimal runtime" Partition OS that can itself be evaluated to high assurance to serve as a base for high assurance Applications;

- A set of drivers suitable for operation in partitions and suitable for evaluation to high assurance (e.g., RS-232, Ethernet) or to lower assurance (e.g., 1553, OpenGL, USB).

A MILS-based platform will undergo special scrutiny to check that there are no covert channels which could be exploited to pass information through unauthorized means.

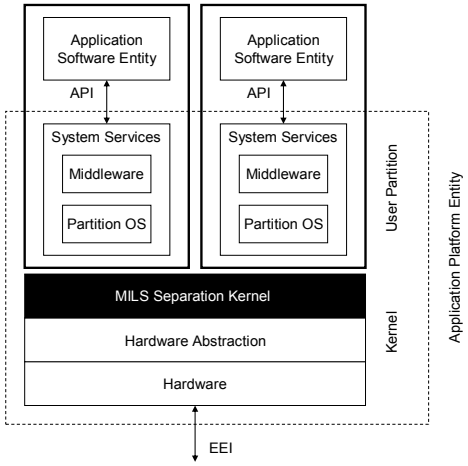

Figure 59. MILS Separation Kernel

10.5.5 Cross Domain Solution/Multilevel Security (CDS/MLS) Operational Environment

With the preceding paragraphs serving as background information, we now present the architectural element that appears to best satisfy the needs for partitioning within the PISR System Security Architecture. The expanded OSE Profile for a CDS/MLS operating environment is provided in Figure 60. This profile provides the capability to securely transfer data among or between different Security Domains and the ability to access information with different sensitivity or classification levels through one computer system.

The profile includes a special user partition that hosts a type of Application Software entity called a Transfer Application. The Transfer Application has access to a type of System Services entity called CDS Services. In this profile, the Application Software entities remain security unaware (except the Transfer Application). Virtually anything which connects to more than a single security level or domain is a CDS/MLS must be certified as such in order to achieve an ATO. In some of the current NSA literature, a high assurance device which connects to multiple security enclaves is shown and it has been called an Assured Sharing Manager (ASM). It is illustrated as different from the typical CDS/MLS device, the certified versions of which are on the UCDMO list, because these devices connect to only two levels of security, such as Secret and TOP SECRET. The ASM connects to all levels of security, from UNCLASSIFIED to TOP SECRET.

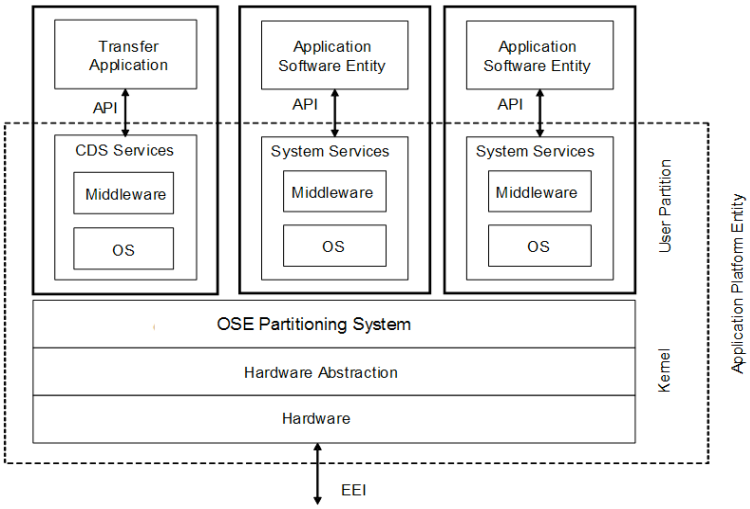

Figure 60. OSE Profile for CDS/MLS

Figure 61 shows a MLS CD security architecture with logical partitioning that has been tailored to UAS requirements. Objects with different security attributes are logically partitioned into separate domains or Communities of Interest. Such domains may include communities that are UNCLASSIFIED, SECRET, TS, SCI, or safety-critical. Each of these domains resides on top of a Security Separation Kernel to isolate the application software from the processer hardware and provide high assurance that the software domains are logically partitioned. Also hosted on the separation Kernel is a cross domain transfer application that includes an Authentication Manager, Trusted Capability Registry, and a Dynamic Policy Manager. This application allows the logically partitioned domains while providing high assurance that all security policies are being enforced. The authentication manager authenticates the source and destination of the information to be exchanges. The Trusted Capability Registry validates the security attributes associated with the domains and information to be exchanges. The Dynamic Policy Manager enforces all defined security policies and provides the capability to change policies as the tactical situation changes.

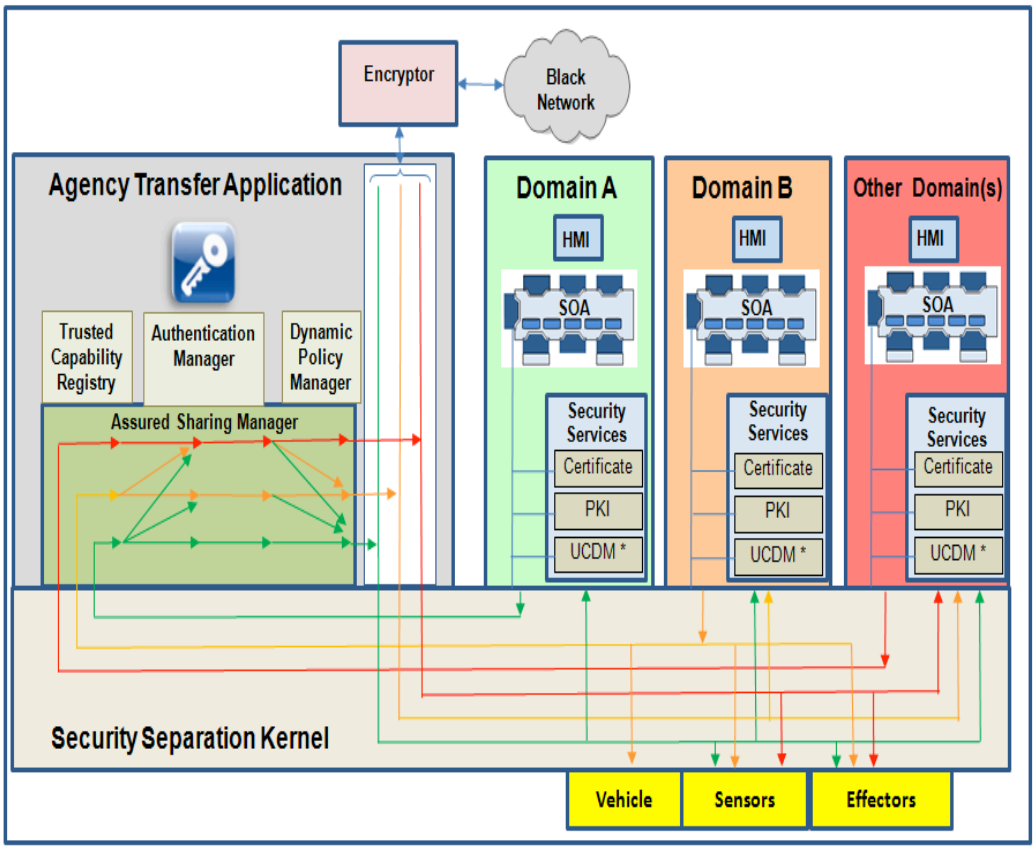

Figure 61. Platform CDS architecture with logical partitioning

10.5.5.1 Authorization: Distributed Policy Based Access Control

The RapidPro security architecture has a requirement for adaptive policy based access control. This is derived from the NSA GIG IA Roadmap for a capability called Risk Adaptive Access Control (RAdAC). RAdAC can be viewed as a new or evolving enterprise requirement which was not possible under the older stovepipe architectures. It is a requirement which enables changes in access privileges based on environmental or operational needs. Because these changes are driven, in many cases, by emerging and developing operational needs, traditional access controls are inadequate. In order to achieve a higher degree of flexibility as well as a greater dynamic range, RapidPro access controls will be attribute-based. Whereas Enterprise-level attributes, because of the many stakeholders at the enterprise level, may be used for backbone access, at the tactical edge, more granularity will be required. In these cases, rather than simple Role Based Access Control (RBAC), Authorization Based Access Control (ZBAC) can be used.

Based on a study previously done with ONR, the RapidPro project anticipates using Soutei, a dialect of Binder, as a policy language. This is open source GFE code and is available at http://soutei.sourceforge.net/. We've nominated

this implementation because of its adaption to DoD 8500 controls, and for disconnected operations. However, in no case will a specific dependency on a specific vendor product be acceptable.

10.5.5.2 Integrity

In addition to the integrity controls inherited through the Identity Management architecture, where there is a critical need and resources in terms of bandwidth and connectivity are available, additional Integrity controls can be implemented using NSA Crypto-Binding. In no case will a specific dependency on a specific vendor product be acceptable.

10.5.5.3 Identity Management

Where available, particularly at the enterprise level, DoD soft certificates using a PKI architecture will be used. Where not available, for example at the tactical edge, a user name and password mechanism, which complies with the DODI 8500 series instruction, such as Open Single Sign On (SSO), will be used. In no case will a specific dependency on a specific vendor product be acceptable.

10.5.5.4 Confidentiality

In some specific cases, NSA Type 1 encryption devices may be required and where this is true, they will be used. Otherwise, for network solutions, other means may be used. In particular, Suite B with SSL, or equivalent, may be used.

10.6 PISR Subsystem Support to the IA Framework

The UI Environment Subsystem needs to define operational views that describe tasks, activities, operational elements, and information exchanges required to conduct operations to meet information assurance guidelines. The architecture must show that it is compliant with DoD Net-Centric Data and Services Strategy. System views need to describe the systems and interconnections. Connections to approved architectures must be shown. Any Web Services Description Language (WSDL) definitions and Extensible Markup Language (XML) instances need to be registered in the DoD Metadata Registry (MDR). Data assets must have associated security metadata that identifies an authoritative source and the UI must insure that the user that accesses that data has the authorization and permission to view it. A compliance test description must be created to specify how the UI will be tested for compliance with the Net NR-KPPs.

IA concerns with respect to the User Interface Environment Subsystem include:

- User authentication – Verify user's identity and the privilege level of data access that they have and insure that all the data that is available to a user falls within that privilege level. If critical data becomes available that is not at the privilege level of the user, the system must provide an option to the user to send the data to a user with the sufficient privilege level.
- Interface authentication – When subsystems communicate with other subsystems, verification that the correct subsystem is really the one communications are going to must be performed along with protection of the data along the communication path.

Since the SA Subsystem generates information for users at the tactical edge including support of combat operations and includes analyzed intelligence data, all data from the SA Subsystem is classified SECRET//NOFORN. Although some data might be suitable for sharing with allies such as "five eyes" nations, the automated information system lacks clear policy guidance as to when this is acceptable. Single-level components are also simpler to field.

Although the SA Subsystem operates at a single security level, sensor integration includes information sources from multiple security domains. For example, SIGINT tips and cues originate from a Top Secret/Special Compartmented Information (TS//SCI) network and are based on more sensitive sources and methods. The tips and cues themselves, however, have already been downgraded to SECRET//NOFORN.

Other sources of sensor data include UNCLASSIFIED video feeds. While the video data stream is unclassified, once an intelligence analyst comments on the feed, the result is SECRET//NOFORN. The SA Subsystem's approach is similar; the hypotheses generated by the sensor level interpreters are SECRET//NOFORN for lack of better classification guidance.

Integrity is also a concern for the SA Subsystem. If users depend on the SA Subsystem for timely intelligence data matching their Conditions of Interest, malicious deletions of COIs could blind users to adversarial actions. Falsified sensor feeds could also either blind the PISR system by replaying old data or lead to ambushes. Forensic data for justifying the actions of US forces also must be preserved with integrity. Human-generated hypotheses should include non-repudiation to help establish reputations. Audit logs must be immune to tampering as well.

As the manager and controller of primary policies involved in coordinating processing and communications across the PISR System, the MCL Subsystem works closely with the Information Assurance Framework to register and manage authentications, authorizations, privileges, and other mechanisms for ensuring information, processing, and dissemination are properly protected. MCL is responsible for the optimization of available resources, including optimization of the dissemination resources and pathways. The properties of dissemination resources are under the control of IA policies. MCL dissemination planning needs to have control of the following resources:

a. PISR IB System configuration, which includes PISR IB subsystems and sub-subsystems.
b. User interfaces, data sources, and external systems (e.g., MarineLink, GHub, Distributed Knowledge and Knowledge Needs (DKKN), DIB).

Subsection 6.2.2.2 provided an introduction to PISR IB Subsystem approaches for persistency of IA policies and User Access Control (UAC). As discussed there, the PISR IB Intelligent Distribution Sub-subsystem cannot deliver valuable information to the devices without adherence to the IA policies defined at the enterprise level. UAC imposes restrictions on the distribution of data to the devices used by the warfighters. For UAC, the PISR IB needs to support schemas for at least the following artifacts:

a. Organization Hierarchies
b. Users with user profiles, which should include organization, user role, user device, etc.
c. Network topology, including communications networks with their profiles
d. Security enclaves and security guards

Considering that all thing/event schemas generate location-indexed views, UAC will be capable of providing read/write access to different kinds of UAC views corresponding to any chosen UAC-related IA policy. For instance, one IA policy may define access to particular information which is role-based. Another IA policy could further restrict access to the information based on the role and the location of the warfighter device. A third policy might restrict access based on user roles and organization echelons for particular platoons and squads for a group of companies, which, in turn, belong to a group of battalions.

This page intentionally left blank.

11 Life-Cycle Management (LCM) Framework

11.4 Introduction

For PISR PLA to be managed in a cost-effective way, it must include Life-Cycle Management (LCM) as part of the overall collaborative development capabilities (document tracking, release tracking, etc.). LCM is an integrated approach to addressing configuration management and software product development from application creation to demise. Without LCM it will be much more difficult, expensive, and time-consuming to develop and maintain a coherent and compatible product line for PISR. The RapidPro project intends that the PISR product system will have full LCM to enable effective systems management and evolution, as well as to provide information for future integration with related systems. Development of the initial PISR application is an engineering challenge because of the project scope and the rigorous testing/validation requirements, as well as the focus on a product line approach. An effective LCM toolset will support this effort with improved understanding of existing systems and effective documentation of software products.

A product line is a family of applications that share a common architecture (or product line architecture – PLA). There are multiple approaches to their development and use. However PLAs generally focus on 1) finding the functionality that is required for all family members, 2) implementing the functionality, and 3) supporting the use of the common functionality for current and future applications. While the first two steps can make development of the first family members quite costly, the payoff comes when continued use of the PLA makes development of additional family members cheaper, faster, and less error-prone. Product lines also facilitate re-use of non-software elements associated with the line, such as documentation and test plans.

The PISR PLA described in this document is intended to provide interfaces to existing and future data sources, based on the individual developer needs and rights. It supports a world view consisting of both real-time, networked data, and databases. The core of the application, four interacting subsystems (as shown in Figure 1 in Section 1.4), provides the functionality required for all members of the product line. The core itself is component-based and configurable to support the requirements of the current product. Current and future components that specialize a PISR product, such as new sensor capabilities or data sources, are intended to plug-and-play via the external interfaces defined for the subsystems.

LCM tools used to support the PISR PLA also must support its information assurance (IA) and interoperability test and certification processes. The goal of RapidPro's rapid prototyping process and deployment cycles is to produce products that provide needed capabilities and are ready to be fielded. To accomplish this goal, IA and interoperability testing and certification processes must be followed and carefully documented throughout the product life cycle, from development to deployment and use. To achieve the shortest IA cycle it is essential to document that each new cycle of development represents an incremental change over the previous cycle. LCM software tools support this requirement by providing the capability to document PISR processes and provide traceability of certification artifacts throughout the product lifecycle. IA certification requires specific configuration of components to be documented and tracked so that certified configurations can be maintained and deviations from these can be readily identified. Configuration management is a key functionality of LCM.

11.5 How a PISR System is Modeled and Configured

Our approach to LCM is intended to support all phases in the development cycle. As seen in Figure 62, a central element in this support is a database that captures information relevant to all of these phases and is updated whenever new information associated with the PISR product line is available.

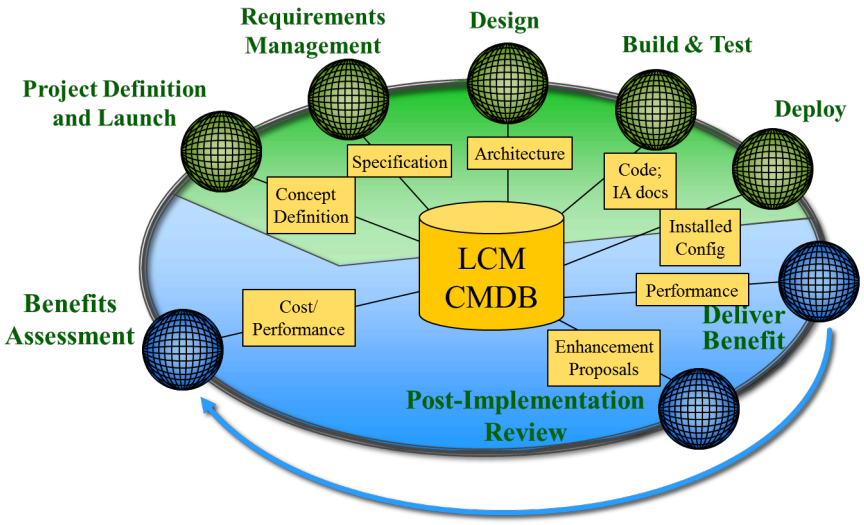

Figure 62. LCM software support

Every member of the PISR product line has at its core the PISR PLA, which consists of four interacting component-based subsystems. These subsystems are configurable and we believe that the subsystem components are assets that will also eventually be maintained within LCM. For example, the MCL Subsystem is responsible for overall resource management in the resulting system. It is likely that different product line members will have different resource constraints and will require a different set of control components. Under this assumption, the first step in creating a new element of the product line is to choose how to specialize the core architecture of the new PISR application.

Most of the specialization of the PISR product line member comes from the components that are intended to plug-and-play at the external interfaces of the PISR PLA. Three of the subsystems explicitly describe external interfaces for these components: SA, PISR IB and UI Environment. For each of the subsystems, there may be many possible components that provide a needed functionality. It is the job of the LCM to assist the product line developers in locating the best component for the task and in assessing how this component will fit with other chosen components and with the PLA. Various kinds of information, such as component inter-relationships, interface information and resource constraints, will be important in this assessment. Therefore, the critical function of LCM is to maintain an adequate and current collection of information about the system and to make that information easy for developers to locate.

Traditional Integrated Development Environments (IDEs) provide support for the development of individual applications and provide a significant part of any LCM approach. However, they do not adequately support creation of new product line members, where the emphasis is on component classification, retrieval, and integration with a core PLA. To manage component-level information, we use an approach derived from configuration management (CM). The central element of a configuration management system is the Configuration Management Database (CMDB), as shown in Figure 63. This is a repository containing Configuration Items (CIs) and the relationships between the CIs. In traditional CM systems, CIs can be hardware elements (such as routers and servers), software elements (such as applications, drivers), or business elements (such as licenses or contracts). The CI relationships also may be of many different types; for a large system, there may be thousands or even millions of items and relationships. Our LCM approach maintains a similar database of elements of interest. The CMDB provides a picture of the environment that allows personnel to query and understand the CIs and the CI relationships in order to find an appropriate plug and play component for each new PISR application.

Most industrial CMDBs use some subset of the industry-standard Common Information Model (CIM) as their schema for information representation. The standard CIM schema is hierarchical and object-oriented. It is designed to

be extended, allowing a CMDB to be specialized to a particular environment and to the type of information required for CI of interest.

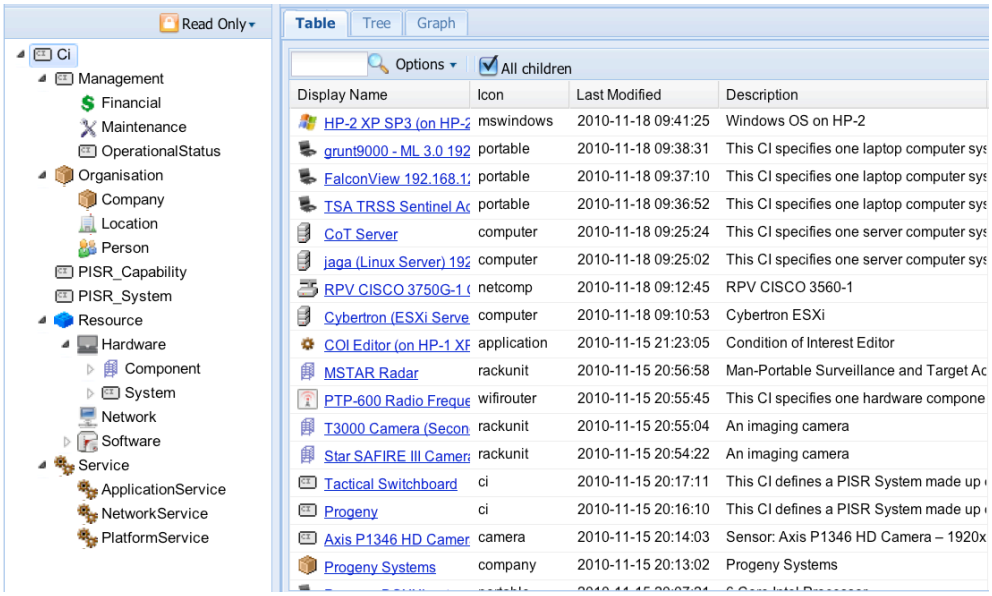

Figure 63. The CMDB maintains information about configuration items

For example, suppose the new PISR system needs a particular kind of sensor capability that is provided by several different systems. The first step in meeting this need would be for the developer to determine what components are capable of this type of sensing. Information about this set of candidates would be provided by querying the LCM system. Once this list has been provided, the developer can examine these components more in-depth to narrow down the choices based on the constraints of the components themselves and of the emerging PISR system. Components that do not match can be detected easily and eliminated from consideration. Where a component requires a particular resource or the use of other specific components, this information is readily available. Interface and integration information from the LCM system can help the developer determine how much effort will be required to plug the component into this PISR system. In cases where a component has been used in a previous PLA instance, this information should be very complete and will allow easy integration. Information about the needed testing/certification requirements for the component and its integration will be available within the LCM as well, to facilitate that part of the development. For example, where a component requires authentication for use, the CI would document how to integrate the component's authentication scheme in a certified enterprise authentication system. The central role of the CMDB is shown in Figure 64.

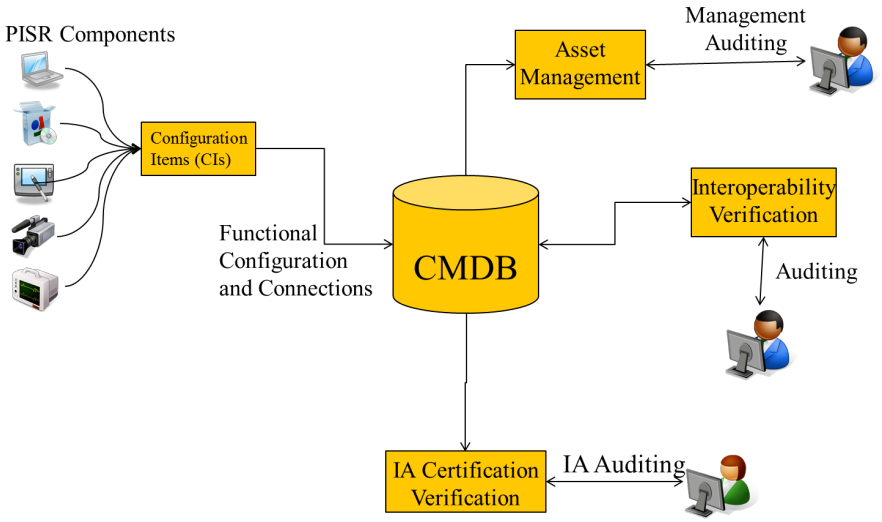

Figure 64. The CMDB is central to all phases in the application lifecycle

Much of the PLA's realization is facilitated by information regarding components and their characteristics obtained from the LCM system. However, information flows into the LCM system as well. If this is the first time a particular component has been used in a PISR application, information about the component can be expanded to assist developers with integration/testing of this component in future family members. Although this requires additional work on the part of developers, keeping the LCM database up to date is essential to make creation of these future members of the product line easier. For initial PLA development, where an existing CMDB is not available, a "brownfield" survey must be completed to bootstrap the CMDB. The result of such a survey is a catalog of the initially available resources, as well as their inter-relationships.

11.6 Description of Asset Management

11.6.1 Processes

Figure 62 showed the various states in the LCM process. Although the RapidPro project intends to cycle around these states more rapidly than most development efforts, the same states apply. The development process leaves a state and goes to the next one when the artifact(s) produced in that state is approved and checked into the LCM system.

11.6.2 Components

Documenting components such as sensors, analytics, communications connectors, planners, and external data sources such as MarineLink is a critical role of LCM for PISR. These components can be used to create a particular product from the core product line functionality. Components intended to configure the core itself and components intended to plug-and-play into the external interfaces are important. These components may come from different sources: the research community, commercial organizations (COTS), or other government agencies (GOTS). The source is likely to influence the amount and quality of information initially available for LCM. If the information for a particular component is not adequate, the "brownfield" documentation process must be repeated to obtain adequate information.

One part of the PLA where this appears to be particularly important is the SA Subsystem. This subsystem is the heart of the valued-information philosophy that drives RapidPro products. External interfaces are defined for components including sensors, sensor integrators, user interfaces that aid SA, and analytics that operate on and interpret the many data sources. One of the tasks in the development of the SA Subsystem is survey, analysis, and classification

of sensor capabilities. To be useful, this classification information will be managed under the LCM system so that the best available components for the current needs can be chosen. To ensure that these components can be integrated quickly into the PLA, information about interfaces, controls, and outputs must also be maintained in a usable fashion.

As an example, Figure 65 and Figure 66 show information associated with the GBOSS Heavy component. The first figure provides characteristics in a hierarchical fashion, including information about relationships to other hardware and software CIs, operational status, and a PISR classification. Much of this type of information can be shown graphically as well, as shown in the second figure.

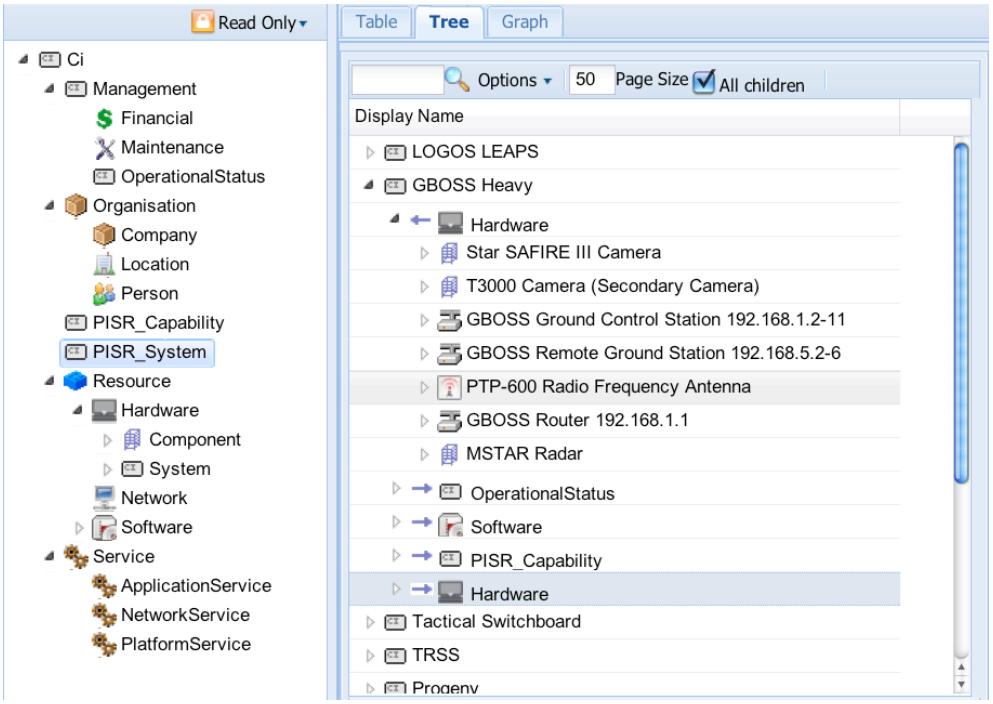

Figure 65. CI information in textual form

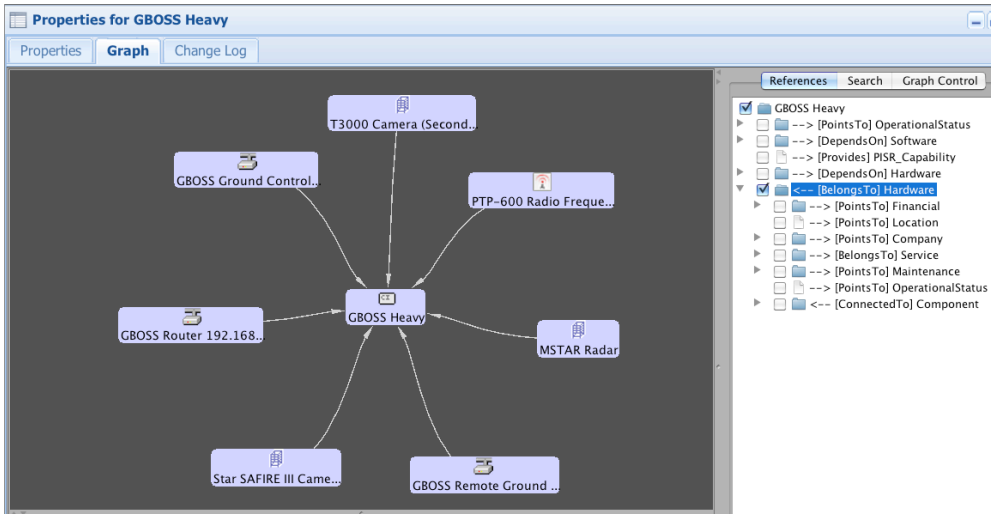

Figure 66. CI elements can be connected by various relationships and shown graphically

.

11.6.3 Components of the Test/Cert Tool Kit/Repository

The PISR Testing and Certification Framework requires management of testing hardware and software, sets of test and certification metrics, test instrumentation, and data gathering and reporting tools. Documentation is required both for the PLA and for the components. Relationships between all of these different components must be maintained.

11.6.3.1 LCM and IA: Complementary Processes

The LCM and IA processes are highly complementary in that both involve careful documentation of system components and processes along with description of their internal and external interfaces. The IA processes are driven by the Defense Information Systems Agency (DISA), Application Security and Development, Security Technical Implementation Guide (STIG), and Application Security Checklist. The checklist includes a comprehensive list of procedures that are performed to conduct an Application Security Readiness Review (SRR) that assesses compliance with the STIG requirements. LCM can be used to demonstrate compliance and best practices in an SRR.

The LCM system provides a versioned artifact repository to store security documents such as System Security Plans, Application Configuration Guides, Coding Standards, Test Procedures and Results, Threat Models, Checklist reviews, and other security relevant documentation. LCM can demonstrate that IA processes were followed throughout the development process by supporting a versioned history of these documents that can be linked to the relevant software releases or code.

Depending on the application, the SRR may require assessment of other dependent systems (for example: web and database servers, directory and authentication systems, firewalls, operating system platforms). The LCM CMDB system will simplify identification of these systems and can be linked to security relevant configuration for each CI stored in the artifact repository.

11.7 How the Tools, Models, and Repositories of LCM Evolve Over Time

The goal of RapidPro LCM development is to provide a set of tools that will add value to the existing LCM process. Our focus is on open-source and GOTS tools that are available off-the-shelf and require minimal tailoring for

PISR PLA. An effective integration of such tools is not available today; the current emphasis will be on achieving that integration.

11.7.1 Initial RapidPro LCM Capability

The initial capability consists of a simple integration of two different types of tools: (1) a CMDB to provide simple configuration management; and (2) an interface for the CMDB to a software development/collaboration tool suite to support traditional distributed software development. The first phase tool uses OneCMDB for the configuration management and CollabNet TeamForge as the development tool suite. The described LCM approach does not depend on these particular tools and we will be continuously addressing their suitability.

OneCMDB was chosen for the CMDB implementation because it is light-weight, open-source, and includes an acceptable client interface, as can be seen in the earlier figures. Its data model is simple but customizable.

CollabNet TeamForge is an integrated suite of web-based development and collaboration tools for distributed groups of software developers. This toolset has many features including:

- Tracker (issues/bugs/features requests): Track issues and artifacts with integrated change monitoring and management. For example, track bug reports or feature requests that are referenced to specific code commits or software releases.
- Documents/Artifacts (file management): A central document repository where documents and files can be referenced throughout the system. Documents and files are tracked and versioned.
- Tasks (project management): Create and manage tasks that need to be accomplished in the project. Reference tasks to specific people or artifacts.
- Code Repositories (via the popular open-source Subversion system): Commit source code, software releases, and related material in a central repository where they are versioned and tracked making it easy to identify what was modified and who was responsible for the modification.
- Discussion Forums (including mailing lists): Facilitate communication among stakeholders and maintain a record of communications.
- Reporting (visualizing data in TeamForge): Project plans and deliverables that provide unified view into development status. Project-wide metrics for insight into real-time activities.
- File Releases (installation packages): Provide a central location for posting and managing complete file release packages including documents, notes, and links. Statistics to track downloads and access control to limit availability to individuals or by role.
- Wiki: Shared web space that can be collaboratively edited among project members. Links can be made to specific artifacts.
- Hudson continuous integration engine: Used to provide automated and scheduled builds of committed code. Provides an easy-to-use system for developers to integrate changes to the project and making it easier for users to obtain a fresh build of code. Monitors executions of internally or externally-run jobs that can run test cases against builds. Hudson keeps those outputs and makes it easy for you to notice when something is wrong.

Most of TeamForge is not open-source; however, it can be integrated with other tools through a Simple Object Access Protocol (SOAP) Application Program Interface (API). It is through this API that we intend to both extract information regarding the state of development and provide information to the TeamForge development environment. Other tools such as the Hudson engine have been integrated in this way.

11.7.2 Integrating OneCMDB and TeamForge

In the initial LCM system, TeamForge and OneCMDB will be used independently by the RapidPro team; however, as development progresses, the information in the two systems will be maintained consistent with respect to each other. The initial integration includes:

- Access to the OneCMDB via a link within TeamForge environment and modification of OneCMDB to support a single sign-on capability so the two systems can share authentication information and access control.

- Customization of OneCMDB data model to PISR systems. This data model will be refined over time as our experience with the architecture and with the development methodologies increases.

The RapidPro project currently imports artifact information using spreadsheets; however, OneCMDB has the capability to import from other types of interfaces well. We may be able to build a custom interface to allow us to extract information from TeamForge into the database. In addition, we will be able to define associations between artifacts in OneCMDB and TeamForge. We intend to avoid replicating information whenever possible, instead providing links using TeamForge's URL/artifact ID system. We intend to provide the ability to reference artifact IDs within CIs in the CMDB that link back to those artifacts existing within TeamForge.

11.7.3 Replacing TeamForge with Other Tools

Open source and GOTS tools will be located and integrated to replace the functions of TeamForge such that most or all tools will have a no-cost license. The capabilities of the initial commercial LCM, TeamForge, may be found in an open-source or low cost software package or some combination of packages. We are considering the software systems below for this purpose.

1. Trac: Web based enhanced wiki and issue tracking system for software development projects. Supports traceability and linking of wiki entries and issue tickets throughout the system, similar to TeamForge. Integrates with underlying version control systems including Subversion and Mercurial. Trac is Python-based and has a completely open-source license.
2. Redmine: Project management web application that supports multiple projects, flexible role-based access control, issue tracking system, Gantt chart and calendar, news, documents and files management, feeds and email notifications, project wiki, project forums, time tracking, and versioning system integration (SVN, CVS, Git, Mercurial, Bazaar and Darcs). Redmine is Ruby-based and has a completely open-source license.
3. GForge: A free software fork of the web-based project-management and collaboration software originally created for SourceForge. Provides project hosting, version control (Subversion and CVS), issue-tracking, discussion forums, document repository, shared wiki, and reporting capabilities. GForge has both open-source and commercial versions with slightly different feature sets.
4. Codendi: Codendi is another web-based application LCM tool developed by Xerox. It includes capabilities for version management (CVS and SVN), universal tracking system (bugs, issues, features), test management tool, document manager, file releases, collaborative editing wiki, code snippet library, survey tools, discussion forums, mailing lists, instant messaging, and RSS tracking. Codendi is provided as a community edition in open-source or professionally supported version.

11.7.4 Automated LCM Interfaces for PISR

An automated interface will be implemented for PISR LCM. Each component will register its latest configuration in the CMDB at startup. This will require that a reporting capability be installed by the developer of each component.

11.8 Use Cases Employing LCM

Scenario #1: The developer has a set of requirements for a new product in the product line.

1. The developer queries the DB to find existing available components that fit some requirement(s) based on the PISR classifications.
2. A set of components that fit the given requirements is returned. For each component, the developer can further query the database to get relevant information for the decision process. Information regarding each potential component includes
   - interface information (syntactic and potentially semantic) that will allow the developer to address integration of this component into the PLA
   - constraints associated with this software component such as hardware requirements, dependences on other system components, conflicts with other system components
   - business information such as licensing.

This information allows the developer to assess the suitability of using each component in the larger software product.

3. For the chosen components, developer can determine how to integrate (software, other artifacts) based on DB information and templates. Some of these integration methods may be automatically generatable. The developer updates the DB information for each component.
4. Product documentation (and related artifacts) is generated for the software product.

Scenario #2: An enhancement to an existing product is needed.

1. The developer queries the DB to determine what parts of the product are affected by the given enhancement.
2. The information returned from the query allows the developer to reason about what new functionality is required and what existing components are affected (directly or indirectly) by the enhancement.
3. If new functionality is required, the developer queries the DB to find existing available components that fit some requirement(s). Just as in initial development, the information (interfaces, constraints) from the query is used to evaluate the suitability of using this component.
4. If new components are to be integrated into the product for the enhancement, integration methods are generated and the DB is updated.
5. The query results also provide information about existing components that must be changed due to the enhancement and other components indirectly affected by the enhancement. New integration methods are generated if needed and information about the component(s) is updated in the DB.
6. Product documentation is updated to reflect the enhanced system.

Scenario #3: An updated version of a component is scheduled to be installed.

1. Even if this component is not used directly in some existing software product, some component of the existing software product may have a dependency on this updated component. The developer queries the DB to determine what other components might be affected by changes to this component.
2. The result of the query should inform the developer whether the updated component will affect the software product. If an issue does arise with a component, the developer have to determine what changes will be needed after the update.
3. Information regarding changes is put into the DB for use in later products and any new/changed integration methods that are needed can be generated.
4. Product documentation is updated to reflect the modified system.

Scenario #4: The IA process for a given software system needs to be documented

1. As noted in earlier, LCM and IA are highly complementary processes. The steps of a typical IA process define a use case.
2. As individual components undergo the IA process, the appropriate security documents are updated and linked to the earlier versions of the document. The component information is updated to link to these security documents. As the component evolves, so do the documents, providing the needed documentation of the history of the component.
3. To address the IA requirements for the software as a whole, it is necessary to determine all relevant components used in a given system and to provide the relevant documents as support. The LCM can provide this complete list and could be used to provide the system level IA history needed.
4. Here is a example of an IA related process:
    a. A System Security Plan (SSP) must be documented and approved for an information system.
    b. The SSP is developed and the system is configured accordingly.
    c. The system configuration is captured as a configuration item in the configuration management database.

d. When preparing for an IA audit, the configuration can be confirmed to match the SSP by comparing it to the recorded configuration.

12 Networking for PISR

12.1 Introduction

This chapter describes a set of requirements for integrating and managing the PISR network. It is the latest chapter in PLA document, based on results of most recent joint studies with Tactical Network Topology (TNT) team during the Fall quarter of 2010. Correspondingly, the current version is limited to most well understood battalion and below networking architecture requirements as well as fundamentals of 8th Layer based network management technique[38] to be designed in accordance with Management Control Layer architecture.

12.2 Battalion and below

12.2.1 Probable ways to deliver bits at this level

At the battalion level and below, robust, ubiquitous, ad hoc mobile mesh networking clusters constitute the core for PISR bits delivery. Within the clusters, operators, unattended sensors, aerial and ground manned/unmanned surveillance nodes (towers, UAVs, UGVs, surveillance aircraft, ground vehicles, ground stations, etc) maintain self-forming networks by controlling their location on-the-move as well as the application load, subject to current terrain and node availability constraints , and COI based information delivery requirements.

Within the cluster (1-3 mile radius footprint) most of the layer 1/2 wireless links are the Line-of-Sight (LOS) types. We define cluster as small scale **squad level** network of operators, vehicles, unmanned nodes, and unattended sensors. However, the mesh character of the node-to-node connectivity allows to overcome most of the LOS obstacles by extending the peer-to-peer mesh around terrain obstacles, or alternating the links through the high elevation (towers in the area) or aerial relay nodes The result is is highly dynamic short-haul architecture, which employs light portable radios, hand-held PISR devices, and wearable relays.

Additionally, within the cluster, several single short-haul obstacle penetration or/and n-LOS links could be employed to augment the self-forming end-to-end mesh by through through-the-wall or n-LOS of capability.

The mesh enabled, sensor-unmanned systems-USMC operator PISR clusters could be interconnected by:

- Broadband wireless point-to-point links via the ground (towers), aerial (UAVs, tactical blimps, or air balloons), and sometimes limited orbital (Ku-band GEOS) nodes. This is a small scale solution with 3-4 PISR clusters, more suitable for the force protection type scenarios, in which the area of surveillance is fixed and doesn't change for several days or even weeks;
- Broadband wireless self-forming mesh links among the cluster gateway nodes via the ground (towers, reconnaissance vehicles, UGVs), aerial (UAVs, tactical blimps, or air balloons), and emerging orbital nodes. This is a more scaled solution for 6-12 PISR clusters, more suitable for highly dynamic ISR scenarios, in which the area of surveillance is changing hourly and might include surveillance areas distributed geographically beyond 200 mi area. Directional steerable antennas are highly desirable for maintaining inter-cluster broadband wireless mess architecture.

Figure 67 illustrates a small-scale PISR cluster example as assembled for the November 15-18, 2010 RPV-TNT Trial.

Mark Pullen 2/4/11 1:09 PM
Comment [1]: This chapter is very hard to follow. It needs some scaffolding – comes across as a bunch of disjoint ideas. Perhaps an introduction could describe the principles involved. Also the chapter does not seem to fit in the larger Architecture document; it needs to explain how the network architecture presented meets the needs of the over PISR-PLA.

Mark Pullen 2/4/11 1:09 PM
Comment [2]: The "8th Layer" concept needs to be described and justified

Mark Pullen 2/4/11 1:09 PM
Comment [3]: Sentence is too long – should break it into two

Mark Pullen 2/4/11 1:09 PM
Comment [4]: ditto

Mark Pullen 2/4/11 1:09 PM
Comment [5]: to you mean 1-3 mile radius?

Mark Pullen 2/4/11 1:09 PM
Comment [6]: can overcome?

[38] Bordetsky, A. and F. Hayes-Roth (2007). "Extending the OSI model for wireless battlefield networks: a design approach to the 8th Layer for tactical hyper-nodes." *International Journal of Mobile Network Design and Innovation* **2**(2): 81-91

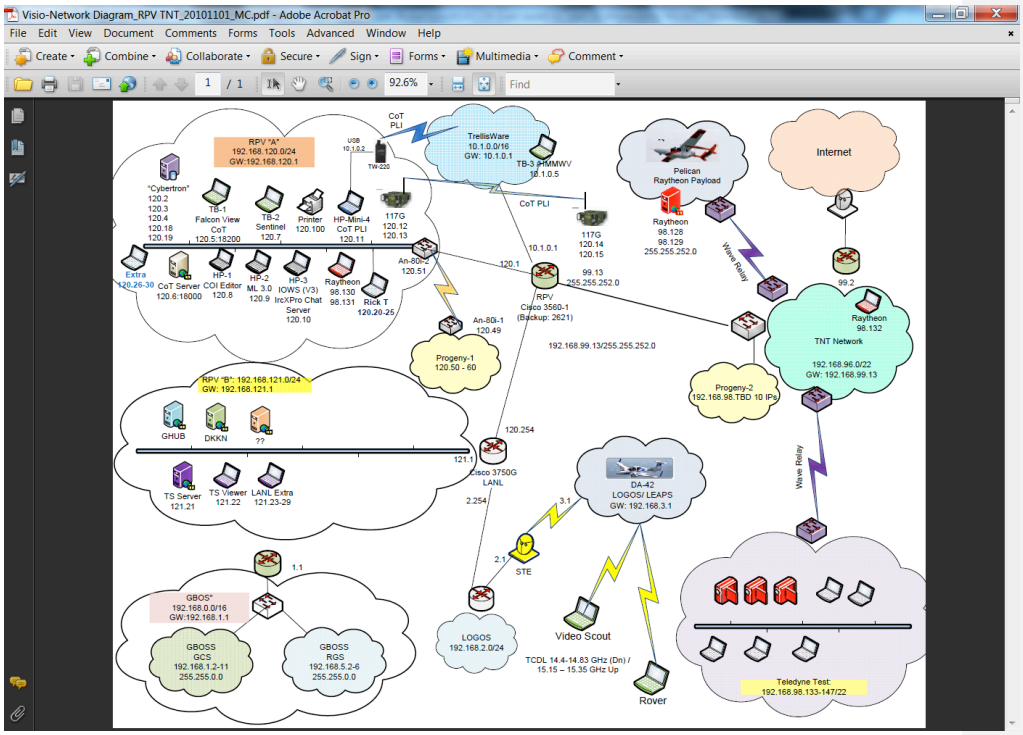

Figure 67. RPV-TNT tactical network diagram

- Support Mechanisms IP Space Routing Architecture
- Wireless Mesh Platforms
- Wireless platform Layer 2 bits-CoT message adapters/parsers: interoperability enablers

12.2.1.1 IP Space Routing Architecture

The routed network design for the PISR architecture was driven by two primary objectives: segmenting portions of the network to reduce the traffic load across bandwidth-constrained network links, and enabling multiple parallel data paths through the network.

During previous tests, it has been observed that with a moderate number of computing devices connected into a common Local Area Network (LAN), the level of "ambient" (background broadcast and multicast) traffic can exceed 1 Mbps. This ambient traffic is largely comprised of ARP requests, NetBIOS announcements, switch management protocols such as Spanning-Tree, and other similar discovery and management protocols. Although these protocols are necessary to support certain application functionality, an excess of traffic on bandwidth-constrained links drastically reduces the "useful" throughput of that link. Most wireless portions of the network, such as the Trellisware data-enabled radios, have a much lower maximum throughput than the wired portions of the network. Overhead traffic that is not noticed on a 100 Mbps or Gigabit wired network can significantly impact application traffic on a wireless link.

To prevent overloading constrained links, routing boundaries were implemented between the primary wired segments and any major and bandwidth-constrained wireless segments. As can be seen in Figure 1, the Track-A segment was separated from the Trellisware segment by a routed boundary. Likewise, Track-A and the TNT segments

were separated, since each network, though wired, contained many computing devices generating ambient load on the network.

Routing also allowed the use of multiple parallel pathways without introducing configuration pathologies. If multiple paths exist from one point on a LAN to another, a pathology called a "bridging loop" can occur, where packets will continue to traverse in a loop between the two points. Most modern switches prevent this behavior by selecting one path and disabling the others. However, it may be useful in some cases to allow certain traffic over one path versus another, or to share the load across multiple paths. Routers are able to implement these rules. For instance, there were multiple connections between the Trellisware segment and the Track-A segment; one supported all end-to-end application traffic, the other was used exclusively for management traffic (node position and performance monitoring).

12.2.1.2 Layer 2 bits-CoT Adapter : An example of Trellis Ware PLI-to-CoT parser

TrellisWare (TW) radio provides Position Location Information (PLI) in two data formats: KML (formerly Keyhole Markup Language) and JSON (JavaScript Object Notation, which is a lightweight data-interchange format) . Due to limited TW radio bandwidth, JavaScript Object Notation (JSON) data wrapping format was selected to be used, since to compare to KML format, the JSON generates more compact data messages. As shown in Figure 68, TW radios are forming mesh network of mobile units TW-1 – TW-n. Each unit provides PLI via mesh network (CheetahNet) by reporting its location to TW-master unit, specifically configure for that purposes. In TrellisWare terms, this unit is also known as command node or CMD.

Mark Pullen 2/4/11 1:09 PM
Comment [7]: Define acronym

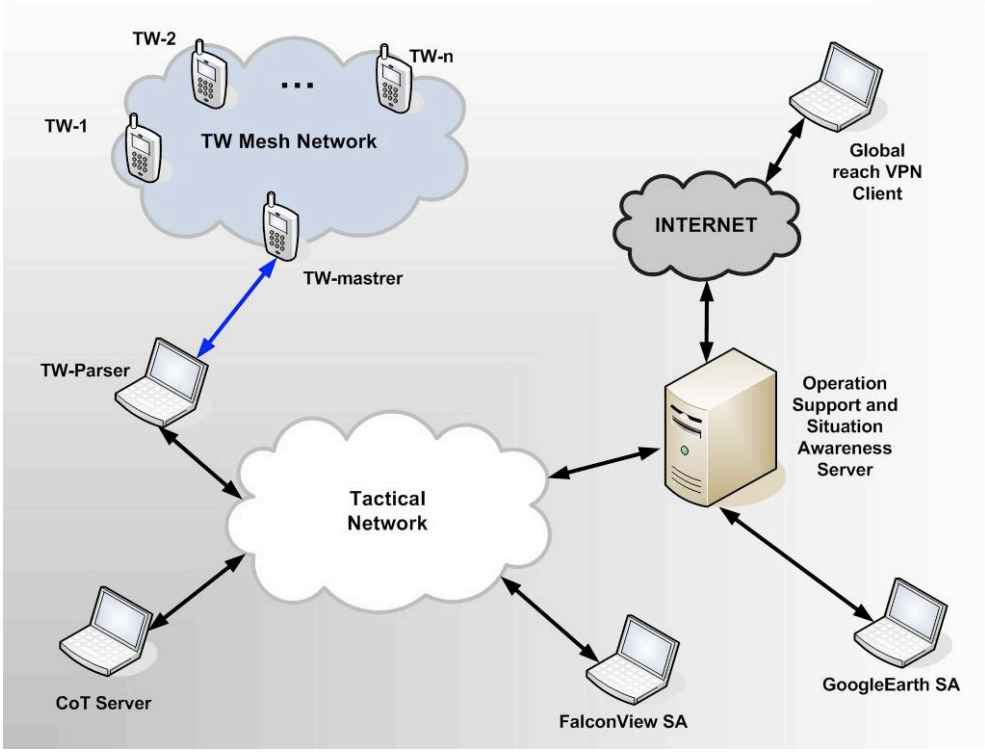

Figure 68. Network topology and TW mesh integration

TW-master radio via USB or Ethernet cable connected to computer running TW-Parser software. TW-Parser software was designed for current RPV experiment to provide the following:

- Polling TW-master data in JSON format
- Parse JSON data
- Generate CoT messages based on parsed data
- Send CoT messages to CoT Server to update FalconView and GoogleEarth SA

Operation Support and Situation Awareness Server generates GoogleEarth SA view based on CoT messages flow. The CENETIX SA Server located in NPS was playing this role in RPV experiment. Another important role of CENETIX SA Server is to provide global reach functionality to the remote VPN clients. Each PLI postings was time-stamped and stored in SA Server database for later analysis and replay. An example of database query of single TW unit tracking on GoogleEarth SA presented in Figure 69. Live tracking as it appears on GoogleEarth SA is presented in Figure 70.

Figure 69. Example of TW unit tracking on GoogleEarth situational awareness

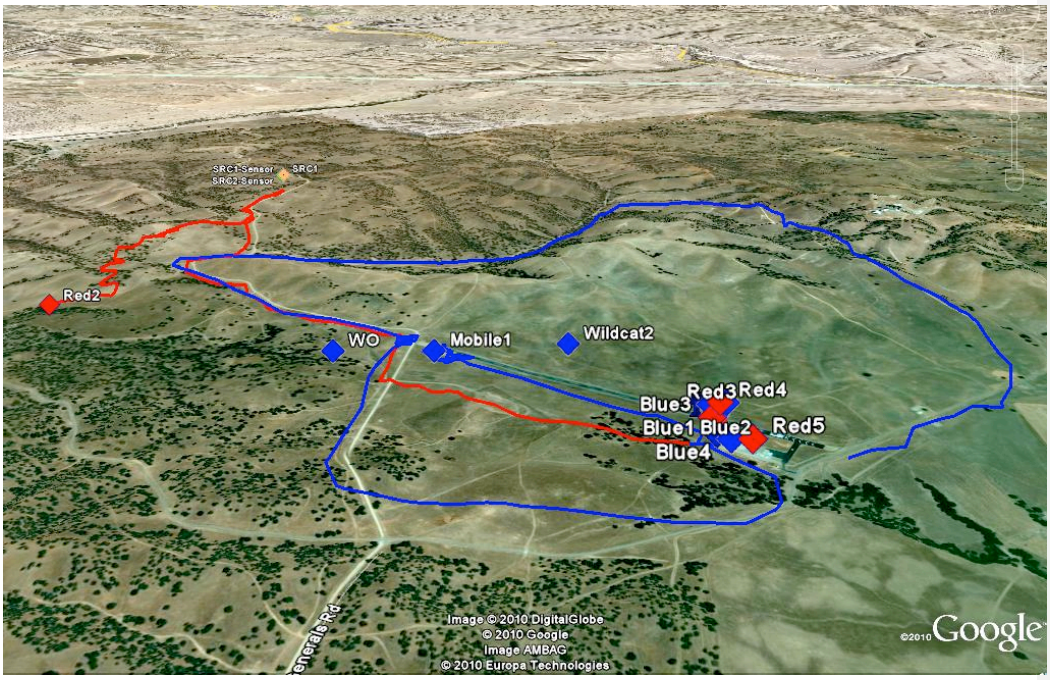

Figure 70. Example of live tracking on GoogleEarth situational awareness

The TW Parser GUI presented on Fig. 3. GUI allows user to assign IP address to the master node (CMD unit) in which PLI from all available via CheetahNet TW units will be collected until polled out by TW Parser within assigned polling interval. The CoT Server IP and its core configurable parameters are also available via TW Parser GUI.

TW Parser GUI provides JSON parsed data from each TW node currently registered with CheetahNet. Only nodes covered by CheetahNet mesh network and providing adequate security key might be successfully registered with CheetahNet. The PLI set of data consists of Latitude, Longitude, Heading, Speed, and Altitude. TW unit registered with network but failed on its GPS fix, will be represented by record with yellow background in GUI table grid as shown on Figure 71. Poor GPS reception or malfunctioning (disconnected) GPS antenna should be considered as the most likely reason for that. The CoT format allows to map the TW radio location and movement into the common operational picture GUI tracks (as shown in the GoogleEarth figures), while the CheetahNet data elements allow to track the health status of each radio node. The association of such two types of GUI, is an important requirement for integrating tactical radio nodes in the battalion level situational awareness environment.

Mark Pullen 2/4/11 1:09 PM

Comment [8]: This seems to be one detail out of many you could have addressed. If it's important, explain why.

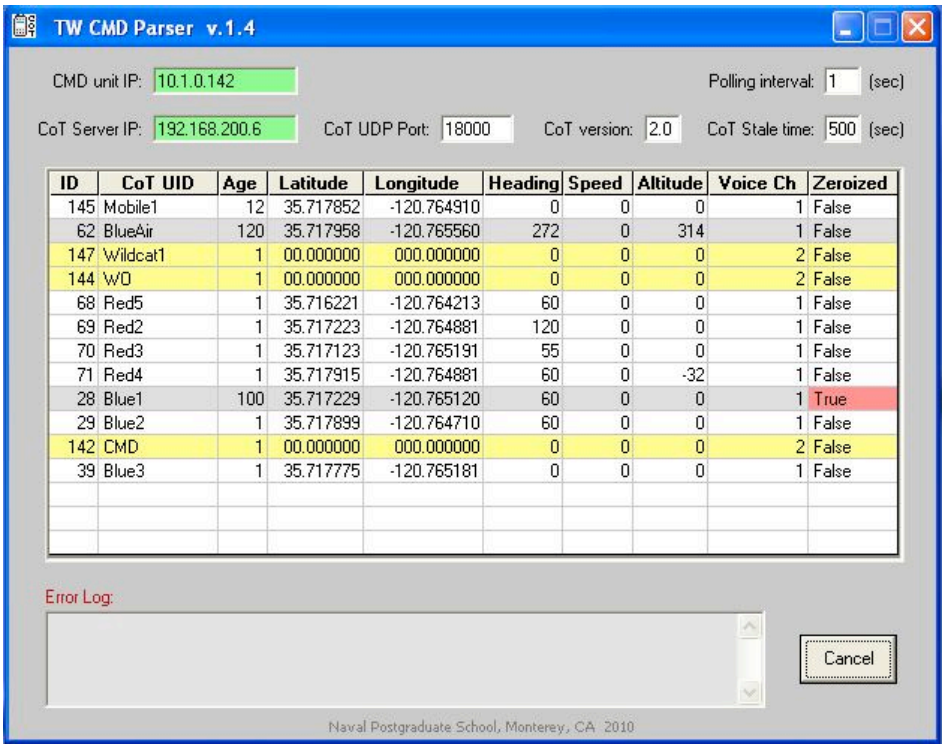

Figure 71. TrellisWare JSON-CoT parser GUI

If no data are received after 20 polling intervals, the TW node is considering as disconnected and will be marked with a gray color background. Some possible disconnection reasons are: out of mesh network coverage, battery failure, zeroized unit. Zeroized unit also marked with a red color background field. The Age field is a counter of polling intervals since the last successful update. If Age is more than 20, then Latitude and Longitude are representing the last known PLI before GPS lost. TW Parser generates CoT message in accordance with unit's status. As a result, the shape and color of TW unit icon is visually representing current unit's status on FalconView SA. The example set of unit icons representing current unit's status are shown in Figure 72.

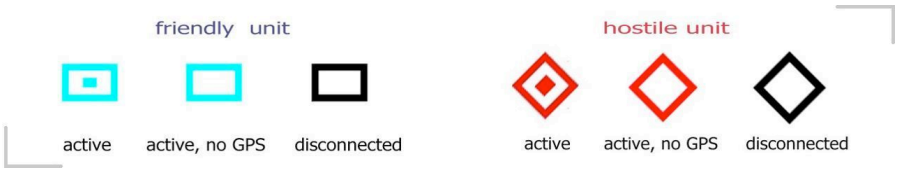

Figure 72. Example of icons representing unit status

The current version of TW Parser works with up to 15 TW units, but can be easily modified to manage more units if needed.

12.2.1.3 Standards for PISR Cluster Mesh

Based on the last 5 years of NPS-USSOCOM-DHS field experimentation with different mesh networking solutions for ISR, HVT (High Value Target tracking) and MIO (Maritime Interdiction Operation) missions, we recommend the following standards for PISR cluster mesh networking:

- PISR Self-forming mesh broadband wireless mesh: OFDM 802.11
- Mesh enabled software programmable radios
- Short-Haul obstacle penetration: UWB (Ultra-Wide Band), MIMO (Multiple Input-Multiple Output),
- Mesh Routing Standard: MANET (Mobile Ad Hoc Networking-DARPA)
- Mesh Routing with Feedback Control: CBMANET (Control-Based MANET-DARPA)

12.2.1.4 Standards for Inter-Cluster Links

Similarly, the extensive field experimentation studies of different inter-cluster links, conducted at NPS for the last 5 years39 show most promising performance of the following platforms:

- Point-to-Point fixed: OFDM 802.16
- Tactical Cellular (GSM, GPRS)
- Mesh mobile, with directional steerable antennas: OFDM 802.11
- Orbital fixed: Ku-Band GEOS
- Orbital routing: IRIS LEOS

Mark Pullen 2/4/11 1:09 PM
Comment [9]: Why were these standards selected? What is important about them?

12.2.1.5 What's off the shelf to support developers/integrators in rapidly reapplying this in the next system

- PISR Self-forming mesh broadband wireless mesh: OFDM 802.11: Persistent Systems Wave Relay, fixed and wearable systems, MANET standard
- Mesh enabled software programmable radios: Trellis Ware radios, Harris 117G
- Point-to-Point fixed: OFDM 802.16: Redline Corporation A 80i system

12.3 8th Layer

12.3.1 How we make this system controllable so that we can optimize the value of bits delivered

In accordance with 8th Layer concept, the PISR network could be made controllable through the coordinated work of PISR node monitors, which associate network status at Layer 1-3 with the health and services constraints at the higher levels of node functionality:

- SNMP events Monitor (OSI layers 1-3),
- SA constraints Monitor (MCL Registration Service),
- Service constraints Monitor (MCL Health Service Monitor,,).

In such an architecture the SNMP event-constraints monitor is simply a commonly used SNMP agent manager, relocated from the Network Management System suite at the NOC to the PISR node 8th layer suite. Unlike it, the

[39] Bordetsky, A. and Netzer, D. (2010), TNT Testbed for Self-Organizing Tactical Networking and Collaboration, *International Journal of Command and Control Research,* Special Issue on Interagency Experimentation, v4, No. 3.

monitors for SA constraints and SLA requirements negotiation do not have a common standard, and these need to be developed.

Given the fact that in the current PISR architecture, Management and Control Layer (MCL) subsystem by Coogar is responsible for monitoring configuration (SA), health, and policy constraints associated with PISR nodes, the 8th control of most valuable bits delivery could be accomplished through the integration of SNMP MIB Agents with MCL monitors. This would allow to put under control such variables as application switching, node physical mobility initiation, receiver context and requirements modeling, sender dynamic information context and transmission requirements modeling, recipient context determination, SLA generation, SLA negotiation, QoS monitoring and SLA assurance, etc.

Mark Pullen 2/4/11 1:09 PM
Comment [10]: 8th layer?

We envision that coordination of different monitoring processes within the 8th Layer would be driven by the network productivity SLA requirements. Each hyper-node would evaluate its own 8th Layer controllable variables. Each hyper-node would attempt to optimize its own sub-network by making changes in the application load, or by moving the node physically to a better postion (Node mobility control) as depicted in Figure 73. The "dualilty" of 8th layer adaptive management technique is that the SNMP-type performance monitor observes an instantaneous network behavior at Layer 2 and Layer 3 levels, however the SLA controls could only be applied via the MCL agent

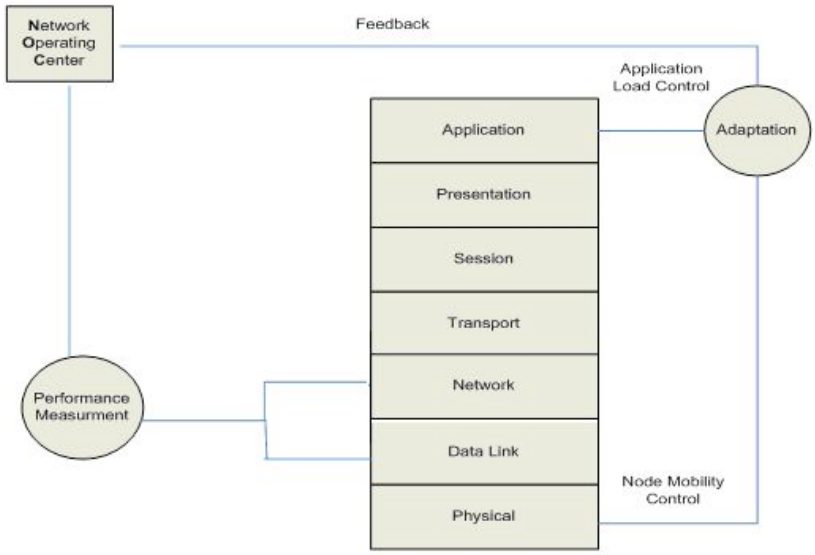

Figure 73. Intelligent adaptation required to maximize network productivity

platform at Layer 7 and Layer 1 respectively (Figure 73). Translation of of SNMP alerts into the load change controls (Layer 7 control), or/and node mobility control (Layer 1) should be done via the MCL Health Status Knowledge Reasoner (performance measures translation into the MCL knowledge base) and MCL Policy Knowledge Reasoner (Layer 7 and Layer 1 controls). Correspondingly, the following 8th Layer Adapters might be needed:

-SNMP (read)-----→MCL Health Status Knowledge Reasoner Adapter: Translation of SNMP based performance measures (RFC 1213 and related SNMP MIB standards);
-MCL Policy Knowledge Reasoner------→SNMP (write) Adapter: Translations of MCL Policy Knowledge Reasoner rules into the SNMP (write) applications load changes and mobility related node controls;
-SNMP MIB extensions might be needed to maintain translation to Health Knowledge Reasoner Alerts and provide for the lower level SNMP (write) controls.

12.3.2 The 8th Layer Memory

In addition to key monitoring processes, the 8th Layer protocol, which enables adaptive network management by the hyper-node itself, should also include a memory mechanism. Such memory mechanism would record and apply a small-scale knowledge base reflecting configuration, performance, security, and application management experiences of NOC crews. The MCL Health Status Knowledge Reasoner could be the main building block for memory component

12.3.3 The 8th Layer Solvers

If we were to define the 8th layer ontology, the most straightforward way would be to represent it through a concatenation of quantitative and context-based constraints reflecting the NEML, NML, SML, and SLA requirements, with SLA constraints defining the goal-seeking intelligence of the 8th Layer. Adapting different resources of physical, link, network, transport, and application layers of hyper-nodes functionality would require a multiple criteria solver, which would enable the hyper-node to perform feasibility analysis and then compromise on a large number of heterogeneous constraints.

Appendix A. A-Level Stakeholder Quality Attributes

QA #	QA	Use-Case Scenario
51	Users should be able to access the PISR System using only a computer browser (Microsoft Explorer preferred) without the need for large application software modules on the user's computer.	A friendly force intelligence specialist asks to have access to the PISR system. He plugs his laptop into the system. After being issued a username, Password and PISR system web site location from the PISR system administrator, the friendly force intelligence specialist connects with the system and begins his work
6	System should make it easy for users to collaborate on any Case File they share an interest in.	User Alpha is presented with a list of Case Files. List indicates clearly which Case Files are open or being worked on at this moment. User selects a Case File that is open and being modified by user Bravo. Bravo receives a notification that Alpha has also opened the Case File. A collaborative chat and edit process is enabled.
2	A Case Files may be linked to another Case File to indicate a relationship between the two.	User Bravo opens a Case File. Bravo inputs a Situation Report that reports that several village leaders meet each Wednesday morning. Report states that Mullah Mohammad Rabbani was one of the meeting participants. PISR system places a notification in Alpha's Case File indicating Bravo's interest in Rabbani. System notifies Alpha that Bravo has an interest in Rabbani which appears in the Case File Bravo worked on.
12	The system provides a high-fidelity criteria-based filtering system so users can decide which Case Files they wish to review.	User Alpha opens a Case File and requests information about other case files that refer to map grid 192961. 100s of other Case Files refer to the same map grid. The user filters these Case Files to those opened or modified within the last 24 hours. This results in 35 Case Files. User selects an additional filter to restrict the area of interest to the northwest corner of 192961. Alpha is presented with five relevant Case Files that satisfy this set of filters.
13	PISR System allows users to define conditions of interest (COIs) that prompt the system to generate notifications against vehicles, people, cargo, and certain specific infrastructure. The user who creates a COI also specifies who should be informed when an event is detected that matches the COI.	User selects vehicle-23 from a list of vehicles maintained by the PISR system. User selects the condition "Enters a specific geographic area" from a list of possible "Conditions". When requested by the system, user enters map grid 192961. User selects "Alert Me with an Email" from a list of ways the PISR system can communicate with the user. User reviews the new COI and posts it to the PISR system and waits for the alert.
1	PISR System supports the creation of Case Files about entities of interest. Case Files contain all information input from the user who opens the Case File including audio files and video clips.	User selects "Case File Creation and Maintenance". User Alpha creates a Case File and identifies Mullah Mohammad Rabbani as a Person of Interest (POI). User enters important information about this person, including who, when, where information was collected. User inputs an audio recording of an interview with Rabbani.
65	The system allows users to retrieve all information about a selected Area of Interest and a date range, then keep or deselect various data elements, and then select a format for the data product.	User Alpha selects map grid 192961 from a PISR system onscreen map. User is presented with a list of information abut the map grid. User selects IED reports for the last 10 days. "IED Detected" icons appear on map with date of detection. User selects a specific IED and is presented with the full IED Detection Report. User selects report of IED Detections from a list of possible reports. User selects PowerPoint from a list of report formats. Alpha reviews the PowerPoint and asks PISR system to Email the report to him.

34	PISR System makes it easy to incorporate behavior patterns such as enemy TTPs, described in the same language used for COIs or embedded in a "black box" algorithm, that have been discovered by analysts or machine learning. The system detects occurrences of such enemy patterns of activity and presents these occurrences to users for assessment. Patterns can be rated as valid or invalid by users based on their performance.	Analysts have created a library of enemy patterns in the PISR System labeled as aggregate COI "enemy behavior patterns". System monitors for instances of these patterns. One pattern is labeled "individual loitering on roof top". PISR System detects this activity over a three day period. PISR System send an alert to users who have subscribed to ""enemy behavior patterns". User Alpha is authorized to validate suspected enemy patterns and validates that pattern correctly identifies "individual loitering on roof top" activity. Alpha uses this pattern to construct a COI that he places into the aggregate COI "Ambush enemy TTP". Users who have subscribed to aggregate COI "enemy TTPs" which includes "Ambush" are notified.
79	Clicking on an item of interest allows the user to drill down to the live information, stored information, and archived information, including history of when additions or changes have been made.	User Alpha asks for a list of Vehicles of Interest (VOI). User selects VOI 22Bravo. User is then presented with a list of the information about the VOI. User selects a live track file and watches VOI stop at roadside.
114	Higher Headquarters (CO, BN, and above) network must be capable of transmitting and receiving tactical edge information essential to intelligence collection, processing, and dissemination.	A fire team spots a possible High Value Individual in a remote area. They take a picture of the HVI and transmit the picture back to BN. They mark the transmission as "Urgent". The wireless network adjusts the bandwidth of each of the links between the fire team and BN to obtain the network resources necessary to deliver the picture. Picture is delivered and action is taken. BN uses facial recognition software and validates the HVI. BN notifies the squad leader to retain the HVI for questioning.
14	Users can specify COIs by choosing one from a list of COI templates and supplying actual values for each choice field in the template. The specific chosen value is drawn from a list of possible values or typed in by the user.	User selects "Create or Edit a Condition of Interest". User is given the option to select predetermined COIs or create a new COI. User selects "COI Templates" and is presented a list. User selects "Vehicle of Interest (VOI) entering an Area of Interest (AOI)". User selects a specific vehicle to watch for from a list of known vehicles. User selects an AOI from a list. User reviews and posts the COI (makes the COI active).
58	PISR System provides information to different users based on their level of access.	User Bravo attempts to access a Case File created by user Alpha. Bravo is notified that access to that Case File is restricted and Bravo does not have the appropriate clearance so he may not access the Case File. Bravo is provided a point of contact to request access to the information he needs.
20	The system provides means for each user to be alerted whenever any COIs are triggered in a space and time close to the point where their own active COI has recently been triggered or is still active. This makes it easy for users to be cued to investigate possible opportunistic coincidences.	User Alpha has entered a COI that alerts when Vehicle of Interest (VOI) 29 enters map grid 192961. User Bravo has entered a COI that alerts when Person of Interest (POI) Mullah Mohammad Rabbani enters map grid 192961. Neither user has requested alerts from other users. VOI 29 enters map grid 19269614, Alpha and Bravo receive an automatic alert indicating activity in map grid 19269614 with a network link to live video.
77	The system allows users to select an Entity of Interest and a date range of interest and then select what data product they would like for this information search.	User selects "Area of Interest" from a list of Entities of Interest. User enters grid square 192961 when requested. User selects a 10 day period. User reviews the information on screen using a map provided by the PISR System. User selects "PowerPoint" for his report and prints his PowerPoint report.

93	User is able to prioritize the notifications he receives from the PISR System.	User enters PISR menu to prioritize notifications. User is informed that "warnings" are automatic and the Highest priority and "threat alerts" are automatic and 2nd priority. User prioritizes the remaining PISR system notifications. User prioritizes his notifications and his prioritization is presented on screen each time the user logs on.
10	User can see a concise summary of each Case File and can drill down for more details as desired. The contents of a summary sheet are standardized by users with appropriate administrative privileges.	User Alpha has a Case File open, whose summary sheet shows the latest values for the entity's identifying information, motivation for the case file, last reported whereabouts, and contact information. Alpha asks to see changes made by user Bravo during the last 10 days and these are presented.
17	System notifies every concerned user when an event occurs that satisfies the COI using the user's preferred communication means. If requested, the system continues to try to contact user until user acknowledges receipt or employs a work around to guarantee the message is accepted by an appropriate alternate.	User Alpha has posted a COI requesting he be alerted if unknown dismounts are observed at night on a nearby road that has been a frequent site for IEDs. He has asked for pop-up screen alarms with confirmation within 2 minutes, followed by text messages with confirmation within 2 minutes. Failing that, he has asked for an alarm to be sent to the Battalion TOC as well as to the watch officers at each company in the Battalion.
57	System operates with Marines onboard ship (Landing Force Operations Center when on ship).	User Alpha is in the Landing Force Operations Center using the PISR System. System comprises 4 laptop computers using data stored on the laptops and additional data from DVDs as needed. System communicates only intermittently with shore through bandwidth-limited satellite communications.
68	User employs familiar tools to view and edit PISR data including case files, MarineLink, Microsoft Word, PowerPoint, Excel. Data are maintained within the PISR System in a common database with formats capable of supporting interoperability among different tools. Data products may be exported to a wide variety of Commercial-Off-The Shelf and Government-Off-The-Shelf tools with different format requirements.	User Alpha opens a Case File that he entered. Alpha asks for a list of documents associated with the Case File. Alpha then asks to see only the Situation Reports. Alpha selects one Situation Report to review. The Situation Report is opened in Microsoft Word. User reviews the report and decides to email the report to user Bravo using Microsoft Outlook.
109	PISR System alerts and warnings that are important to mobile warfighters are pushed to platoon leaders.	A platoon leader receives a short text message that enemy vehicles are approaching his position.
115	The system (to include appropriate network nodes) can simultaneously disseminate data to multiple nodes and users.	User Alpha has a COI asking for notification of all IED activities. Fire team on patrol spots a possible IED implantation. Team leader transmits this important information to his COC. At the same time, the wireless network using prearranged policies automatically transmits the information simultaneously to several key individuals at BN and CO.
31	PISR System helps manage Priority Information Requirements (PIR) by showing how each PIR is linked to various COIs and Case Files that address that PIR.	User Alpha is creating a PIR. He is presented with a list of current PIRs and a diagram showing how current COIs are linked to PIRs. Alpha uses this information to create his PIR and associate it to appropriate COIs and Case Files. Once a week, Alpha reviews the entire set of PIRs.

81	System keeps a record of the history of all data to include information about where the data originated and what operations (in the PISR System) have occurred to change it.	User requests all activity for last 10 days in map grid 192961 and is presented with a list of reports, notes, alerts, warnings and notifications. User is also presented with a list of audio, video, picture and map products from grid 192961. User selects several products to review and receives the products along with information about who, when and what initiated and modified the products, including automated inputs from the PISR system.
97	System should provide a High level of availability through the use of redundancy, distributed databases, graceful degradation and other techniques.	Power cable to COC is accidentally cut during construction around the COC. The primary PISR System servers stop working. PISR System users continue their job using batteries in the laptops, battery backup for their desktop computers and PISR System data stored on the laptops and desktops.
112	Wireless network has the ability to self-form, segment, and reconnect to radios that leave the network and then come back into range.	Dismounted patrol is in a mountainous area. Two team members enter a cave and radio contact is lost. One team member moves back toward the mouth of the cave while maintaining line-of-sight with the other team member. Radio contact is reestablished with the two Marines in the cave and with the rest of the team.
127	System prevents denial of service attacks through the use of Host-based security services and other elements of DoD Computer Network Operations (CNO) that are part of Information Operations (IO).	A denial of service attack is launched against the PISR system. The system identifies the risk, alerts the system Administrator and begins logging attack attributes. The system implements predetermined procedures for mitigating the risks.
4	Case File is able to contain references to documents, audio files, and video files stored in databases external to the PISR System.	User is reviewing a Case File about a mission that used the PISR system to find and deactivate an Improvised Explosive Device (IED). User determines that the IED is a type developed by Al-Qaeda personnel trained in Kenyan Africa. User places a link in the Case File to a CIA-maintained database of reports on the activity in Kenya so other users can find the reports.
9	The system supports the use of pre-determined categories such as “Warning”, “Threat” and “Watch” and corresponding lists such as “Warning List”, “Threat List” and “Watch List” The system maintains these lists automatically as users change category labels associated with entities. The system provides user-tailorable procedures for responding to changes in these lists.	User Alpha has a Case File open and determines that a new threat has developed within map grid 192961. User selects “Enter a new Threat” option from the PISR system. User enters Threat and publishes the Threat to the PISR system. Immediately PISR system maintained list of Threats is updated. Threat alert system takes over the computer resources necessary to send all users specified in the “Threat” response procedure a short Alert message detailing the new Threat with links to information about the Alert.
54	Users may query the system’s information base using ranges of attribute values and typical logical connectives such as AND, NOT, and OR.	User is developing a training mission for local police in the town of R’ayat Godale. User queries PISR system for reports of enemy attacks in and around R’ayat Godale and observations of movement of enemy vehicles toward R’ayat Godale in the last 24 hours. User is provided a list, selects a report format, and asks for a printed report.
60	User authorized to provide access to the PISR System can assign permissions to an individual, a billet, a role and other policies. Permissions can be granted at different levels to different users. For example unclassified read and write permission or unclassified read only permission.	PISR System access is provided to a foreign national partner. Partner is granted access to Situation Reports with read access only. The partner has no other access to system resources.

70	PISR System provides notification to user when cross-cueing of sensors would benefit user.	User is developing a COI within a Case File. COI will alert when any vehicle stops within Map Grid 192961. User begins developing a Collection Plan, is presented with a list of PISR assets with information about Map Grid 192961. User selects a wide area sensor mounted on a GBOSS tower. System. System automatically notifies user that a high-resolution Pan-Tilt-Zoom (PTZ) electro-optical sensor is on the same tower that can track targets identified by wide-area sensor. User adds PTZ and cross-cueing requirements to Collection Plan.
72	System provides the coverage area of sensors selected for use. System also provides the areas that are not covered by the sensors, highlighting known or suspected enemy locations.	User is developing a COI looking for truck traffic through a check point. System shows the coverage area of sensors with coverage of the check point. PISR System also shows potential ambush positions that are not covered by any sensor.
126	PISR System integrates with multiple existing displays (C2PC, CPOF, FalconView, GCCS, etc.)	A vehicle of interest (VOI) has entered an area of interest, satisfying a Condition of Interest. An alert is sent to the COI author, and the C2PC map display is updated showing the VOI.
24	Users can define COIs for High Value Entities, especially High Value Individuals (HVIs). Users can specify who should be alerted when HVIs are detected, as well as personal weightings for false alarms and false positives. The user can modify an associated list of information to be provided with an alert for the HVI.	User Alpha opens a Case File and identifies Mullah Mohammad Rabbani as a High Value Individual (HVI) and asks for any sighting with a 75% confidence or higher. Two days later, sensor 22Bravo, using facial recognition alerts Alpha that there is an 85% probability that the image connected to the alert is Rabbani. Alpha validates that the image is Rabbani, selects two other users from a list and send alert, picture and a short bio of Rabbani.
35	Activity Patterns associated with Enemy tactics, techniques, and procedures (TTPs) that indicate risk are consistently applied across the PISR and related information sources, in near real-time, so that users relying on PISR alerts and warnings do not “miss” instances of those patterns.	Several large enemy trucks are gathering in map grid 192961. PISR system identifies this activity pattern as possible troop buildup before an attack. PISR system sends an alert to all users who have subscribed to information from this map grid. System also sends alerts to a list of personnel maintained in the PISR System who require alerts about enemy activity. List was provided by Battalion Commander referring to Standing Operating Procedures. List was entered by system administrator.
38	PISR System can recommend actions to gain valued information from operations that it expects to encounter entities of interest, such as entities identified in open case files, entities that trigger COIs, or entities that might trigger a COI if additional facts were true. The System generates an RFI and routes it to an appropriate liaison for consideration.	Alpha is on patrol. Alpha is sending photos of vehicles that he sees on patrol to the COC. PISR System is receiving the photos. PISR System identifies one of the vehicles as a vehicle previously tagged as “Highly Suspicious”. PISR System alerts watch officer and suggests that Alpha place a Radio Frequency Identification RFID tag on the vehicle.
40	High Interest Objects (for example, objects that are currently threatening) can carry with them additional data products, demand more user attention and use more system resources.	User Alpha is monitoring the movement of a threat vehicle. Threat vehicle is approaching a FOB. User Alpha upgrades threat to a warning. The vehicle is currently being tracked by an overhead asset. The Warning is sent to all users specified by a predetermined prioritized Warning procedure with a link for the live track file. Each user who selects the live track is provided the PISR system resources to view the track of the vehicle, even if lower priority alerts and notifications are delayed.

74	Collection Managers can establish COIs to monitor and assure that PISR assets are being used efficiently to support current information requirements.	Collection Manager enters a COI that request assessment of the efficiency of the PISR system to meet current information requirements. Manager is presented with a list of requirements that are being met and a list that is not being met. Collection Manager uses this information to develop a new collection plan.
94	System prioritizes the value of information from different sources so information from a High Credibility (trusted) source can replace information from a Lower Credibility source and then, as appropriate, automatically update any associated conditions of interest and subscriptions.	PISR system has been using a wide area surveillance sensor to look for moving targets along a road section. Several Case Files depend on the wide area sensor to detect traffic past a check point on the road. A new High resolution sensor is installed at the check point. PISR system notifies all users subscribing to the wide area sensor and suggests that the new high resolution sensor is available for their use. Notification includes a map indicating the coverage of the present sensor and the coverage area of the new sensor.
95	PISR System supports Collection Manager at each level in determining priorities for PISR asset utilization. Collection Requests are linked to high priority needs. Users are made aware of current priorities and the authority establishing the priority each time they construct an Information Request.	User Alpha is constructing a COI requiring 24/7 surveillance of a building. Alpha is presented with a list of current priorities and users of each PISR asset. If Alpha needs an asset to implement his COI, Alpha uses the reported users and priorities in constructing his Information Request. Alpha attaches several Human Intelligence reports to his IR to strengthen his priority request.
118	System uses knowledge of terrain in the battlespace to support optimizing PISR deployment and utilization	User wants to allocate aerial RapidPro asset to monitor for people occupying positions overlooking a planned convoy. The system identifies the specific advantageous overlook points and generates these as targets for surveillance.
45	System autonomously applies knowledge to draw inferences, make connections, and predict threats. System automatically shares this information across different components and user groups who have specified corresponding COIs or that it predicts would value the information. .	An intelligence gathering dismounted patrol is entering a village. PISR System alerts Platoon leader that a GBOSS tower in this AOI indicates that unidentified vehicles are approaching the village from the South which the system interprets as a plausible threat. Intelligence Analyst at the COC also gets the alert. Analyst believes the vehicles to be enemy activity and advises the Platoon leader using a VHF radio that an enemy attack is possible within 20 minutes.
56	Collection manager is notified by PISR System when the collections requirements of his collection plan are satisfied. Collections manager is also notified if his plan expires without being satisfied.	User Alpha develops a collection plan to support the movement of a FOB to a new location. The PISR system informs him that the collection plan is implemented as requested. After the move is complete, the collection manager terminates that plan.
59	Users can have different levels of access to make global, regional, local or no additions or changes to system data files.	System administrator is adding new users. User Alpha is given universal add/change/delete rights to information. User Bravo is given read only access. User Charlie is given read-only access to one information source (MarineLink).
73	Users can de-clutter a PISR System Common Operational Picture (COP) by selecting and deselecting PISR assets and information feeds.	User requests a list of all objects in his area of interest that are actively being tracked. User is presented with a map showing the latest position of each actively tracked entity. User selects a track file for a particular Vehicle of Interest (VOI) and deselects all other information. User is presented with a list of information about the remaining track file and vehicle. User selects historical positions of that VOI for the last 3 days. User is presented with 3 days of track data including position and other state data for this VOI. User can manipulate a time slider to alter what data is viewed.

75	The PISR System maintains an assessment of all sensors fielded with the system and updates the assessment whenever a new capability is added to the system or a previously available capability is removed.	Current unattended ground sensors are assessed as providing high value to user. New unattended ground sensor systems are added with increased spatial resolution and target classification ability. Reassessment by PISR system indicates that both systems are now providing high value.
110	The system should work-around communication problems where possible to assure timely delivery of important information.	User Alpha has a Case File with a subscription to all information about IEDs. User Bravo enters a suspected IED location in the PISR database. User Alpha is not online. PISR system uses Blue Force Tracker (BFT) information to determine Alpha's location in a Humvee. PISR System sends text notification to Alpha through the FBCB2 system in the vehicle.

This page intentionally left blank.

Appendix B. Tier 1 Test, Evaluation, and Certification Measures of Effectiveness