12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

A Dynamic Infrastructure for Interconnecting Disparate ISR/ISTAR Assets (the ITA Sensor Fabric)*

Joel Wright,Christopher Gibson
Flavio Bergamaschi
Emerging Technology Services
IBM UK, Hursley Park
Hursley, Winchester UK.
{joel.wright,gibsoncr,flavio}@uk.ibm.com

Kelvin Marcus,Ryan Pressley
Gunjan Verma,Gene Whipps
US Army Research Laboratory
Adelphi, MD, U.S.A.
{kmarcus,ryan.pressley,gunjan.verma,gwhipps}
@arl.army.mil

Abstract – *Modern ISR[1]/ISTAR[2] networks comprise a very diverse and disparate set of asset types and networking technologies, which provide a unique set of challenges in the areas of sensor identification, classification, interoperability and sensor data sharing, dissemination and consumability. This paper presents the ITA Sensor Fabric, developed as part of the International Technology Alliance (ITA) in Network and Information Science, a middleware infrastructure that addresses these challenges by providing unified/universal policy controlled access to, and management of, ISR/ISTAR networks. The ITA Sensor Fabric spans the network from command and control, through forward operating bases, and out to mobile forces and fielded (both mobile and/or fixed) sensors in the area of operations. This paper also presents a use case scenario developed in partnership with the U.S. Army Research Laboratory (ARL) and deployed in ARL's Wireless Emulation Laboratory (WEL), that demonstrates the ITA Sensor Fabric applicability to coalition operations.*

Keywords: Sensor networks, middleware, fabric, policy, ISR, ISTAR.

1 Introduction

The diversity of sensors and networking technologies commonly used in fielded sensor networks, particularly in a coalition context, provide a unique set of challenges [1] in several areas including:

- Sensor identification and discovery.
- Sensor access and control.
- Sensor data consumability.
- Policy-based interoperability and trust.

The ITA Sensor Fabric (or Fabric) is an evolving middleware infrastructure, developed as part of the International Technology Alliance in Network and Information Science [2], that addresses these challenges by providing unified access to, and management of, sensor networks. The Fabric is designed to simplify the development and operation of sensor network solutions, specifically those concerned with how sensors are attached, discovered, and utilized. In this paper we describe the Fabric, and its application to a simulated representative coalition operation scenario.

The Fabric spans the network from the data centre to deployed sensors and mobile personnel. It tracks the sensors, nodes, and users of the sensor network facilitating universal access to sensor data from any point, and maximising its availability and utility to applications and users. The Fabric is an extensible platform. Its plug-in architecture allows new functions (such as filters, transformations, policies, security, and event detection algorithms) to be deployed into the sensor network and selectively applied to sensor messages en route to the user.

The Fabric, in addition to its use with fielded sensor networks, can also be used as a research and development tool that bridges the worlds of simulators and fielded systems. Its plug-in architecture, sensor data record and playback capability, sensor simulation, and performance measurement features make it ideal as a research platform that offers a high degree of fidelity with fielded systems (indeed it is such a system). Many technologies, techniques, and algorithms can be easily integrated with the Fabric providing significant opportunities for experimentation, evaluation, and an accelerated route to use in the field.

*Manuscript received May 22, 2009. This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

[1]Intelligence, Surveillance, and Reconnaissance.

[2]Intelligence, Surveillance, Target Acquisition, and Reconnaissance.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2009		**06-07-2009 to 09-07-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Dynamic Infrastructure for Interconnecting Disparate ISR/ISTAR Assets (the ITA Sensor Fabric)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Emerging Technology Services,IBM UK, Hursley Park,Hursley, Winchester, UK, ,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002299. Presented at the International Conference on Information Fusion (12th) (Fusion 2009). Held in Seattle, Washington, on 6-9 July 2009. U.S. Government or Federal Rights License.

14. ABSTRACT
Modern ISR1/ISTAR2 networks comprise a very diverse and disparate set of asset types and net- working technologies, which provide a unique set of chal- lenges in the areas of sensor identi cation, classi cation interoperability and sensor data sharing, dissemination and consumability. This paper presents the ITA Sensor Fabric, developed as part of the International Technol- ogy Alliance (ITA) in Network and Information Science a middleware infrastructure that addresses these chal- lenges by providing uni ed/universal policy controlled access to, and management of, ISR/ISTAR networks. The ITA Sensor Fabric spans the network from com- mand and control, through forward operating bases, and out to mobile forces and elded (both mobile and/or xed) sensors in the area of operations. This paper also presents a use case scenario developed in partnership with the U.S. Army Research Laboratory (ARL) and de- ployed in ARL's Wireless Emulation Laboratory (WEL) that demonstrates the ITA Sensor Fabric applicability to coalition operations.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Public Release**	**8**	

First and foremost the Fabric exists to provide sensor network clients (i.e., the final consumers of sensor data) with controlled access to sensor network assets. At a fundamental level each sensor can be viewed as a source of one or more data feeds. The Fabric provides clients with the means to discover, control, and access the data feeds that they require to complete their tasks [3]. Clients can be individuals accessing the sensor network via one or more applications; they can also be software processes (for example, agents and services) that require access to sensor data to complete a higher-level task that will ultimately provide value to a human user. Clients can be located at any point on the Fabric. They will often be monitoring sensors from a workstation in a command and control centre. Equally, they might be co-located with (or close to) the sensors themselves. For example, and in a commercial setting, an engineer who needs to monitor sensor output whilst performing maintenance in an industrial plant.

In the remainder of this paper we describe the Fabric and its applications in more detail. Section 2 describes the software architecture of the Fabric and its various components, as well as providing a discussion on extending its capabilities. Section 3 details the application of the Fabric in a simulated case study, which provides a motivating example for the use of the Fabric in an ad-hoc networking environment involving coalition operations. To complete the paper, concluding remarks are given in Section 4, and directions for further work in Section 5.

2 The ITA Sensor Fabric

The ITA Sensor Fabric is a two-way messaging bus and set of middleware services connecting all of the network's assets to each other and to users. The Fabric leverages a publish/subscribe messaging model with multi-hop capabilities, and ensures that messages are propagated efficiently, without duplication, and with the minimum use of valuable network bandwidth. Through the use of the Fabric, an ad-hoc network of communicating nodes can be viewed as a SOA[3]-style service bus, with transparent handling of connections and routing (Figure 1).

A Fabric node is a node on the sensor network that runs the following software stack:

- An instance of a message broker [4].
- An instance of a *Fabric Manager*.
- An instance of a *Fabric Registry*.

Clients also attach to Fabric nodes, consuming both local data and data received from other Fabric nodes in the network (although in both cases the client's interface to the data is via the bus). The Fabric Manager and Fabric Registry processes are described in more detail below.

[3]Service Oriented Architecture

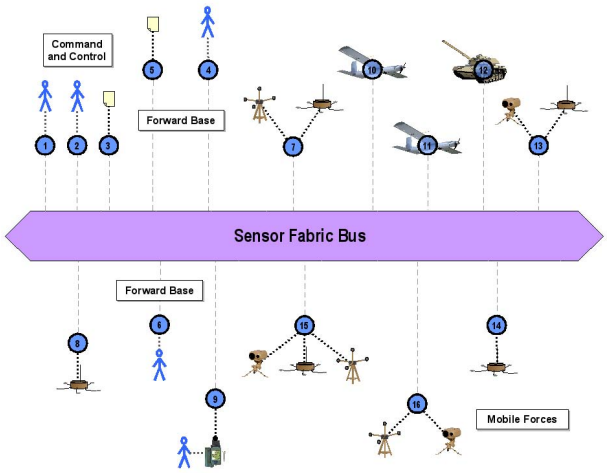

Figure 1: The abstraction of the interconnections within the sensor network provided by the Fabric bus.

When requesting data, clients simply ask their local Fabric Manager for feeds from one or more deployed assets, requiring no knowledge of the structure of the deployed network itself. The Fabric's handling of connections and routing means that a deployed network, such as that seen in Figure 2, may be viewed using the logical bus of Figure 1 by the network's end points, reducing the complexity of data acquisition and dissemination.

2.1 Publish/Subscribe Messaging

The ITA Fabric builds upon the publish/subscribe messaging pattern, which is well known and well established in Enterprise Application Integration. In this section we give an introduction to the basic architecture and benefits of publish/subscribe messaging.

Publish/subscribe messaging provides a one-to-many distribution mechanism that utilizes a central hub or broker through which all messages pass. Thus a single message, published from a device on a low bandwidth and/or high-cost network to a message broker, could be efficiently distributed to a large number of subscribers. The message broker administers the message distribution process using a topic-based approach. It receives subscription requests from client applications, indicating the topics in which they are interested. When a message is received from a publisher on a topic that matches one or more of the subscriptions, the broker sends a copy of that message to each registered subscriber.

The most powerful aspect of broker-based publish/subscribe messaging is the decoupling of publishers from subscribers, as depicted in Figure 3. Clients only have to deal with their connection to the broker; they have no knowledge of the source or destination of the messages, unless it is specifically included in the content of the message.

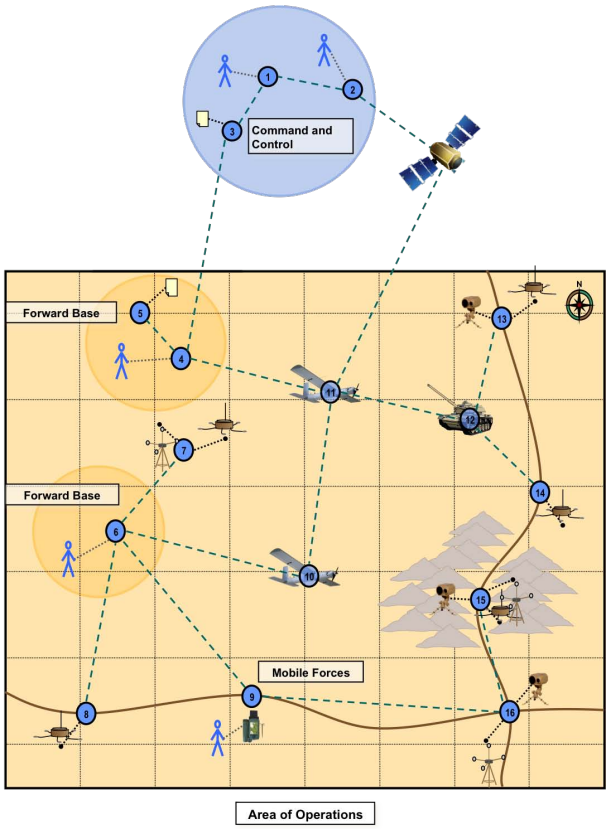

Figure 2: The topology abstracted by the Fabric bus.

2.2 The Fabric Messaging Bus

The Fabric is a two-way messaging bus that connects assets and clients on a sensor network. The connection is not direct; it is decoupled via the publish/subscribe paradigm. That is to say, producers of data (sensors) publish to the bus, and the consumers (clients) then subscribe to the bus in order to receive the sensor data. The data itself is identified on the bus using a globally unique topic name and, as such, every data feed from every sensor is uniquely identifiable.

Not only sensors publish data onto the bus; software services that aggregate, fuse, transform, and filter sensor data will also publish their output using the same technique. Furthermore, the Fabric uses the publish/subscribe model not only to connect publishers with subscribers, but also to send control and management messages.

Sensor networks consist of a set of interconnected nodes. Typically there is not a direct network connection between each node, and so the Fabric provides multi-hop communication capabilities. Poorly connected nodes at the edge of the network can fully participate on the Fabric bus giving Fabric users a single, fully connected, system. The distributed publish/subscribe capability is implemented using interconnected brokers, one per Fabric node, with the message propagation handled by the Fabric. Every effort is made to route messages efficiently, without duplication, and with the minimum use of valuable network bandwidth. To achieve this, the Fabric Manager employs an algorithm to determine whether messages have been altered by the application of a plug-in, and to combine those messages, destined for separate clients, that are identical. This reduces the network bandwidth usage when the transformed message is re-published to the next node(s) en route to the subscriber(s), and the process then begins again on the next node in the journey.

At the pure messaging layer the communication is anonymous. In a distributed messaging fabric, this means that a publishing sensor node or subscribing application need only be concerned with the connection to its local message broker. The Fabric transparently orchestrates the broker-to-broker communications between the sending and receiving applications.

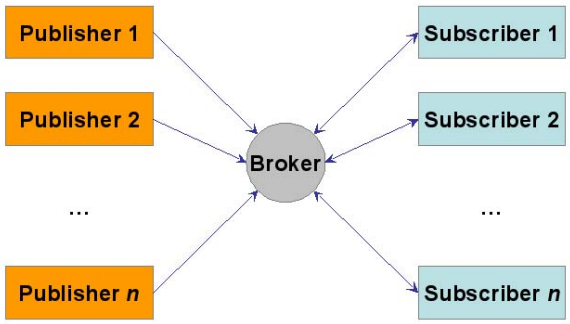

Figure 3: Decoupling publishers from subscribers.

2.3 The Fabric Manager

As the main Fabric service running on a node, the Fabric Manager has wide ranging responsibilities. It builds the major capabilities of the Fabric on top of the message broker (which provides the actual communications infrastructure) and the Fabric Registry (which is responsible for storing the configuration and operational status of the Fabric infrastructure). The major features and functionality provided by the Fabric Manager are as follows:

- Establishes the communication channels between nodes, handling the routing of message between publishers and subscribers attached to the bus.
- Tracks the operational status of connected sensors and nodes, and in the future will register local data fusion algorithms as intelligence assets with the Fabric Registry.
- Provides a container for running plug-ins, in-network information fusion, filtering, and other algorithms.

- Provides the point from which the capabilities of the Fabric may be extended.

2.4 The Fabric Registry

Infrastructure information about the Fabric, including details on all deployed nodes and other ISR/ISTAR assets, is recorded in the Fabric Registry. As new assets are added to the Fabric they are automatically included in the Registry, making them available for use by all Fabric users. The Fabric Registry is implemented as a database, storing information about the currently deployed infrastructure in a number of tables defining: asset types, asset instances, networking and routing information, tasks and task associations, Fabric users, and Fabric extensions.

Four database tables detail the physical assets deployed on the Fabric: nodes, platforms, sensors, and feeds. Each of these tables includes details such as type, physical location, operational characteristics, neighbours, task commitments, and current operational status/availability, along with a human readable description of each asset. Every class of asset (node, platform, sensor and feed) has an associated table describing the various types known to the Fabric. These types form a hierarchy when defining individual asset instances: platforms are connected to nodes, sensors are mounted on platforms, and one or more feeds are published by sensors. Another group of related tables define the configuration of the plug-ins and extension points (i.e., application defined message processing) deployed on each node, with plug-ins configured either to be applied to each message passing through a node, or to be applied to messages destined for specific tasks or clients.

Further tables define tasks, and associations between tasks and collections of sensors and nodes. This allows clients to subscribe to a group of nodes required to perform a required task without having to subscribe individually. The database also maintains a table describing which clients are subscribed to which tasks.

Routing information is also defined within the Registry. A table in the database details the neighbours of a node, including how those two nodes are connected, and whether the connection is currently active. One or more routes between two nodes may then be described including the individual hops in the route, along with the cost associated with each hop. This information may then be used to choose the route for messages, depending on criteria defined by the subscribing clients, for example to minimize the total cost to the network for transmitting the data.

The current development version of the Fabric employs a distributed Registry implemented using the Gaian distributed/federated database [5], also developed under the ITA program. GaianDB is an extension to Apache Derby [6] whose principal feature is its ability to autonomically discover and federate other GaianDBs, usung a scale-free algorithm. Its connectivity model is biologically inspired in that it strives to minimize network diameter and maximize connections to the fittest nodes. GaianDB advocates a flexible "store locally, query anywhere" (SLQA) paradigm.

By implementing the Fabric Registry on a distributed database, the Fabric has in-built resilience to network failures. Should a portion of the Fabric become disconnected then the fragments can continue to operate as Fabric-islands until they are reconnected. Each Fabric-island effectively has its own Fabric Registry, composed of the distributed database elements that are still connected, meaning that the node information stored in the visible database fragment describes exactly those nodes and sensors available on the island. Note that this process also works in reverse, for example when two fabric islands connect. Optionally, clients can choose that subscriptions to message feeds be persistent; i.e. when Fabric islands reconnect the interrupted persistent subscriptions can be re-established.

2.5 Sensors

ISR/ISTAR sensor assets publish data to topics in a global name space. They have no notion of whom or what is interested in their data; they simply respond to received control messages, or requests to capture readings (and publish them for use to the broker on their local Fabric node). Subscribers may be client applications or software agents such as fusion algorithms or filters. Subscribers need not be concerned with the technical details of connecting to individual sensor nodes to retrieve data. Instead they simply subscribe to their local broker for one or more topics in the global name space and wait for the data to arrive. It is the role of the Fabric to establish the communication channel between the two brokers, transparently to the Fabric end points.

A Fabric deployment may consist of physical sensors, software sensors, or any mixture of the two. Software sensors include data fusion algorithms that, for example, aggregate data from a number of other sensors and publish the results as a new sensor data feed. A subclass of software sensors are simulated sensors. These are software applications that publish generated artificial sensor readings, or replay pre-recorded sensor data in the same fashion as a real sensor. By default, applications running on the Fabric are unable to distinguish between real and simulated sensor data[4], and this makes simulation a powerful technique for creating a repeatable experimental framework.

Attaching a new sensor to the Fabric requires one of the following configurations:

- The sensor runs a Fabric Manager and broker, allowing it to connect directly to the Fabric.

[4]The Fabric does provide an optional mechanism to distinguish between replayed and live sensor data.

- The sensor runs as a broker client, connecting to a more capable utility node in the Fabric that runs a Fabric Manager and a broker.
- The sensor uses an alternative communication mechanism, such as a direct serial connection, to connect to a utility node, whose Fabric Manager then provides a bridge to the broker.

2.6 Extending the Fabric

The core Fabric provides the minimum set of services required to implement a distributed bus framework. Additional capabilities can be added in the form of pluggable modules ("plug-ins"). Several types of plug-in are available, each intended to augment a specific aspect of the Fabric's operation.

2.6.1 Fabric Extension Points

The Fabric is designed to allow major functionality to be added and allow base functionality to be replaced without the need to alter the core messaging bus. The mechanism used is the Fabric Extension Point; an approach that builds upon the Fabric's core message passing functions. All inter-node Fabric communications are message-based, where each message is a combination of meta-data, routing information, and payload data. Furthermore, each message is given a type associating it with a specific extension point. Note that extension points represent a very different approach from the Fabric plug-in model that we will describe later: extension points provide the means to fundamentally alter the behavior of the middleware layer to accommodate new techniques and methods; plug-ins primarily deliver sensor feed subscription-based services.

Extension points have access to the core Fabric services such as messaging, the Registry, and event handling (for example, handling sensor disconnection events). As such they can be fully integrated into, and therefore extend, the operation of the Fabric. For example, the standard sensor subscription service provided by the Fabric is implemented as an extension point and can therefore be replaced should it be superseded by a future data dissemination algorithm. Furthermore, this technique ensures that the Fabric can be deployed onto nodes in a fully modular fashion, ensuring that the footprint can be optimized to suit each individual target.

2.6.2 Plug-ins & Message Flow

Messages flowing through the Fabric are available for processing at each node on their journey. Plug-ins are small code modules loaded into the Fabric Manager that perform processing tasks directly on messages as they flow through the Fabric from publishers to subscribers.

Three types of plug-ins are automatically applied to messages passing though a Fabric node: node, task and client plug-ins. They typically perform operations such as policy enforcement, filtering, transformation, logging, caching, encryption, or routing, and their application order is well defined. Node plug-ins are applied to every message passing through the node, whereas task and client plug-ins are applied to either messages for a specific task, or for a specific client associated with a specific task, respectively. Since the node plugins are applied to every message regardless of task or client subscription, we can reduce the processing overhead by applying each of these transforms only once before moving on to the task related, and lastly the client related, application paths.

The subscription mechanism manages the loading, invocation and unloading of plug-ins, based on their association to the various tasks and clients registered for the incoming message. Figure 4 illustrates how messages received by the broker on a Fabric node are passed to the message dispatcher, which applies the appropriate plug-ins based on the currently active subscriptions.

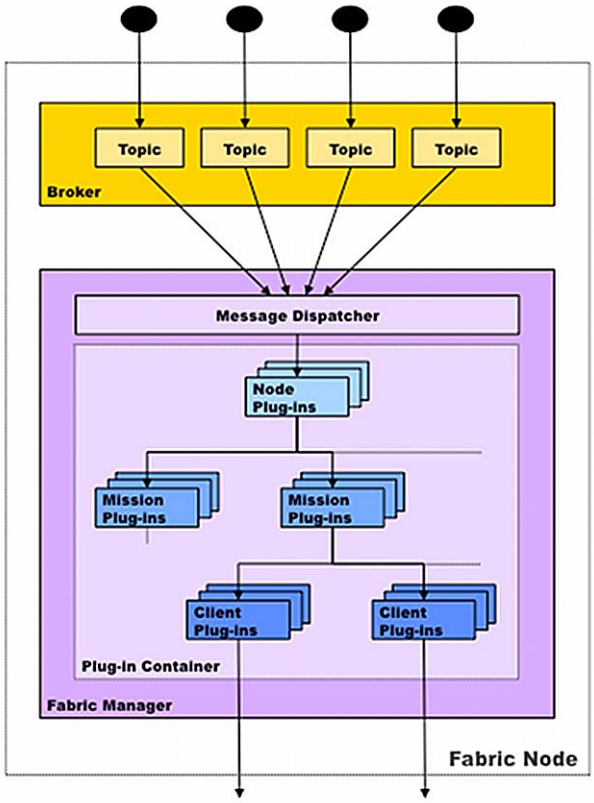

Figure 4: The flow of messages through a Fabric node.

2.6.3 Fablets

These are independent programs running within the Fabric Manager, with all the freedoms that implies. A typical application would be to introduce data into the Fabric from sensors that are not Fabric-aware, or data fusion algorithms that require access to more than one Fabric feed.

2.7 Policy

As a direct result of the collaboration between teams within the ITA program, the Fabric has been policy enabled with the Watson Policy Management Library (WPML), allowing authorization and obligation policies to be enforced on messages routed between nodes. Figure 5 illustrates how the WPML, along with the plug-in architecture of the Fabric, enables dynamic deployment and configuration of policies. WPML builds on a widely accepted policy architecture that consists of four basic elements: a policy management tool, policy repository, policy decision points, and policy enforcement points. WPML employs the Common Information Model Simplified Policy Language (CIM-SPL) [7][8] and the Apache Imperius policy engine [9] to specify and execute declarative rules that control the Fabric message flows.

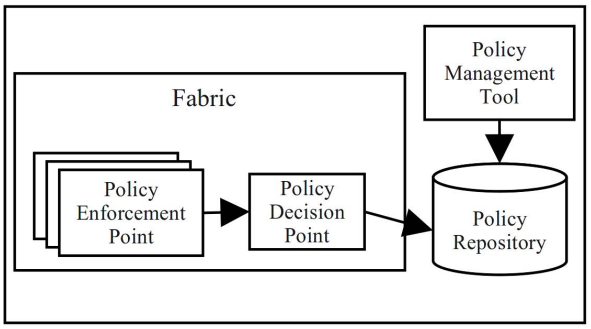

Figure 5: WPML architectural overview.

A policy is a condition-action pair, where the specified action is executed if the condition evaluates to true. Both the condition and action can be defined over sensor data (e.g. the bytes of an image) flowing over the fabric, or over metadata describing the message (e.g. the organizational affiliation of the client of the message).

The two main types of policies are authorization (e.g. allow/disallow access to a sensor data feed) and obligation (e.g. downgrade the quality of the sensor data feed). A policy repository (PR) stores policies that are available for execution; these may be activated or deactivated based on mission requirements. A policy decision point (PDP) is a logical entity which evaluates applicable policies in the repository to message data and/or metadata and returns the result to a policy enforcement point (PEP), a logical entity that is tasked with making a decision, e.g. whether or not to grant a requesting client access to a sensor data feed. The PEP requests the decision from the PDP and enforces the response received.

Figure 6 illustrates a policy used to implement authorization utilizing client affiliation in order to allow or disallow access to high-resolution imagery by U.S. clients. The example policy, when activated, has the effect of downgrading the resolution of images destined to U.S. affiliated clients by 50%.

```
Import Class
com.ibm.watson.pml.pep.IAuthorizer:authorizer;
Import Class
com.ibm.watson.pml.fabric.runtime.IFabricMessageRuntime:fabricMessageRuntime;
Import Class
com.ibm.watson.pml.fabric.messaging.messages.IImageMessage:imageMessage;

Strategy Execute_All_Applicable;
Policy {
  Declaration {
  }
  Condition {
    fabricMessageRuntime.getClient().getAffiliation() == "UK"
  }
  Decision {
    authorizer.allow()
  }
}:1;
```

Figure 6: Policy example.

3 A Coalition Case Study

Civilian and military coalition operations often require that two or more organizations form an ad-hoc partnership to achieve a common goal. Each participating organization operates under a set of inherent restrictions, often stated as a set of security and legal policies, which might have to be combined and harmonized across the coalition. One of the key aspects of coalition operations is the reliance on fusion, dissemination and sharing of information originated from an ad-hoc heterogeneous and disparate network of ISR/ISTAR assets, such as human intelligence, unattended sensors (fixed or mobile), data extraction, fusion and correlation providers, and network elements, belonging to the participating coalition organizations. The dissemination and sharing of information in such environments is subject to a set of organization specific and common policies.

The scenario for this case study involves two coalition countries, the United States (U.S.) and the United Kingdom (U.K.). Both countries will participate in an ISR/ISTAR operation and will share information from their specialized sensors. The goal of the operation is to identify, locate and photograph the source of an acoustic event, and make all the sensed data (including the imagery) available to the command and control sites of both countries.

The U.S. will contribute: a set of unattended ground sensors (US:UGS) [10] capable of reporting the line of bearing (LOBR) of an acoustic event; an unmanned aerial vehicle (US:UAV) to provide a communications link between the UGS and the rest of the Fabric; and a data fusion application (US:Analytics) to fuse the information from the UGS, calculate and publish the most likely location (US:LOCR) of the acoustic event. The U.K. will contribute a high-resolution camera mounted on an unmanned aerial vehicle (UK:Reaper), which is controlled/commanded from an analytics application (UK:Analytics) that based on the US:LOCR data calculates the pan, tilt & zoom (PTZ) commands (UK:CamCmd) to be issued to the camera. Upon responding to the PTZ commands, the high-resolution camera will be commanded to photograph the location

of the acoustic event and make the imagery (UK:Image) available on the Fabric. All the sensed data and imagery, subject to policies, is made available for consumption to personnel at the command and control centers of each country.

3.1 ARL Wireless Emulation Lab

For this case study we deployed the scenario on a mobile ad-hoc network (MANET) emulation environment, the Wireless Emulation Laboratory (WEL) [11], hosted at the U.S. Army Research Laboratory (ARL). The WEL provides a controlled, repeatable emulation environment for the analysis of algorithms, protocols, and applications pertaining to MANETs. The WEL supports several areas of research including coalition warfare, network science and network security for MANETs. These areas of research are facilitated by a suite of software tools running in the WEL ranging from experiment/scenario design to real-time network emulation and visualization (Figure 7).

The real-time emulation software is based on the Mobile Ad-hoc Network Emulator (MANE) originally developed by the Naval Research Laboratory [12]. MANE only emulates the physical and MAC layers and therefore offers the flexibility to support a broad range of network applications such as the ITA Sensor Fabric. The software controls network connectivity between nodes by forwarding and/or corrupting packets according to a user specified mobility and propagation loss model. MANE contains multiple propagation models including range-dependent, free-space loss, and Terrain-Integrated Rough-Earth Model (TIREM).

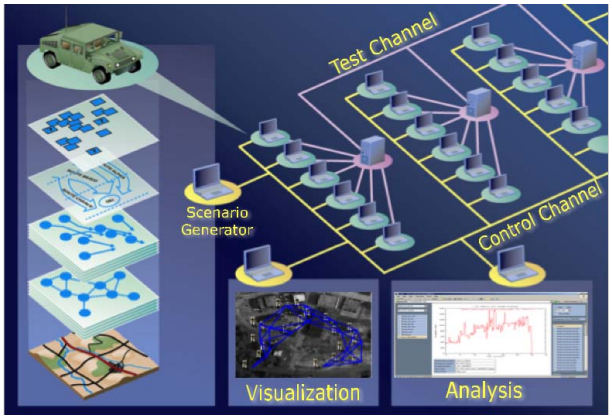

Figure 7: WEL architecture overview.

To implement the scenario described below we used four unattended ground sensors (UGS), one acoustic event generator, one high-resolution camera and five Fabric nodes deployed on the WEL.

3.2 Scenario deployment

Our deployment used a subset of the WEL consisting of 25 test nodes in a wireless mesh configuration, five of which hosted Fabric nodes, analytic applications and policies. Figure 8 illustrates the scenario setup and coalition affiliations. The four US:UGS were bridged to node AG which acts as an auto discovery client. Node A (US:UAV) is a Fabric node configured as an auto discovery server and provider of communication to the rest of the Fabric. When the four US:UGS are discovered, they are added to the Fabric Registry and their US:LOBR data feeds made available for consumption on the Fabric bus. Node B (US:Analytics), consumes US:LOBR data provides a US:LOCR feed. Node C (UK:Analytics) consumes US:LOCR data, calculates the required PTZ parameters and publishes them (UK:CamCmd). Node D (UK:Reaper) consumes UK:CamCmd, uses the PTZ parameters to point the camera to the source of the acoustic event, photographs the site and makes the imagery (UK:Image) available on the Fabric. The U.S. and U.K. command and control centers (US:CmdCtrl and UK:CmdCtrl) are connected to node E (USUK:CmdCtrl) which manages and controls the dissemination of information to the two coalition partners. At the command and control center of each country, a Fabric application consumes and displays the US:LOBR, US:LOCR and UK:Image data. The availability of the information to each country is controlled through access policies that are evaluated by the Fabric at the USUK:CmdCtrl node.

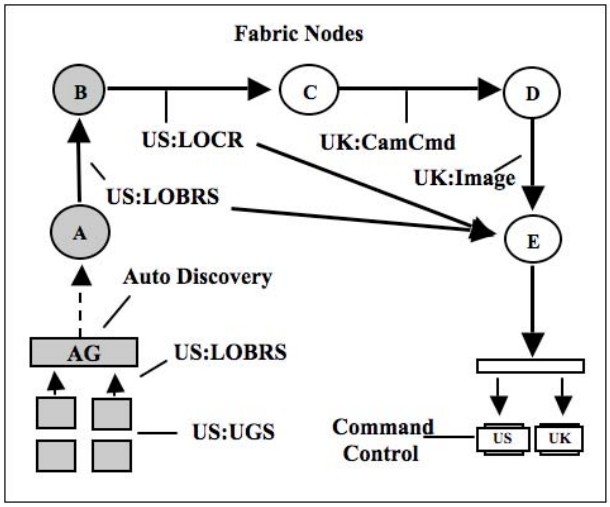

Figure 8: The coalition scenario.

4 Conclusion

We have produced the basis for a middleware for sensor networks which facilitates universal access to intelligence data from any connected point on the network, which monitors and controls connected assets, and has built-in resilience to failures. It is capable of spanning the network from the data centre to deployed sensors and personnel, and maximising the availability and utility of

intelligence data through task planning services, analysis applications (including fusion algorithms and agents), human analysts, and mobile personnel.

Messages are propagated efficiently to minimise the use of valuable network bandwidth, with algorithms (such as data fusion, transformation, filtering, and policy enforcement) delivering in-network processing of data. The capabilities of the Fabric can also be extended through programming interfaces (including Web interfaces) for application or algorithm development, as demonstrated by our example. The Fabric Registry provides management services for Fabric assets by tracking nodes, data sources (hardware assets and software services), users, tasks, and deployed algorithms; and also managing the visibility of, and access to, intelligence data feeds. These features make the Fabric useful both in real network deployments and as a tool for simulation and research; the current development version of the Fabric will be available mid-2009 from the IBM alphaworks site (http://www.alphaworks.ibm.com/).

5 Further Work

The Fabric lays the groundwork for many advanced features, and we will actively extend its capabilities for both field and research use. As suitable new technologies are created by the ITA research programme we will continue to demonstrate their application on the Fabric, with the aim of providing them with an accelerated route to fielded use. The value of this approach has already been demonstrated with the distributed/federated database and policy technologies that are currently being integrated into the Fabric. In particular we will continue our focus on the ITA's policy, security, and networking technologies. The value of empirical feedback from practical applications of the Fabric (such as the case study described above) has also been observed, and we will continue to use this as a source of new, and refinements to existing, capabilities. One area of particular interest is the use of the Fabric on non-IP networks, or more typically, mixed IP/non-IP networks. We will be targeting radio networks that implement serial communications: point to point or multi-point to point (i.e., many sensors into one node). The intention is to deploy a full Fabric node at the edge and allow it to connect into the bus.

The Fabric is considered to be a micro service bus. We will more fully develop its interfaces with the mainstream enterprise services bus technologies that form the backbone of SOAs. The goal will be seamless provision of edge-of-network services, such as those typically deployed on the Fabric, directly into the enterprise. Equally, we will provide users at the network's edge with controlled access to enterprise services and provide a platform for the development of novel solutions that, today, can be complex and expensive using traditional techniques. Note that while such integration is already possible via the Fabric, we will be investigating patterns of use, and identifying technology gaps, towards fully exploiting this capability.

References

[1] G. Pearson, "A Vision of Network-Centric ISTAR and the Resulting Challenges," in *Unattended Ground, Sea, and Air Sensor Technologies and Applications X*, E. M. Carapezza, Ed., vol. 6963, no. 1. SPIE, 2008, p. 696302. [Online]. Available: http://link.aip.org/link/?PSI/6963/696302/1

[2] G.Cirincione and J. Gowens, "The International Technology Alliance in Network and Information Science a U.S.-U.K. Collaborative Venture," *IEEE Comms. Mag,*, vol. 45, pp. 14–18, March 2007.

[3] M. Gomez, A. Preece, M. P. Johnson, G. de Mel, W. Vasconcelos, C. Gibson, A. Bar-Noy, K. Borowiecki, T. L. Porta, D. Pizzocaro, H. Rowaihy, G. Pearson, and T. Pham, "An Ontology-Centric Approach to Sensor-Mission Assignment," in *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management*, 2008.

[4] Really Small Message Broker. [Online]. Available: http://www.alphaworks.ibm.com/tech/rsmb

[5] G. Bent, P. Dantressangle, A. M. D. Vyvyan, and V. Mitsou, "A Dynamic Distributed Federated Database," in *Second Annual Conference of ITA*, September 2008.

[6] Apache Derby. [Online]. Available: http://db.apache.org/derby/

[7] *CIM Simplified Policy Language (CIM-SPL). Specification DSP0231*, Std., Rev. v1.0.0a.

[8] D. Agrawal, S. B. Calo, K.-W. Lee, and J. Lobo, "Issues in Designing a Policy Language for Distributed Management of IT Infrastructures," in *10th IFIP/IEEE International Symposium on Integrated Network Management*, 2007.

[9] Apache Imperius Project. [Online]. Available: http://incubator.apache.org/imperius/index.html

[10] N.Srour and T.Pham, "Acoustic UGS for Todays Battlefield," in *NATO SET-107 Symposium on Battlefield Acoustic Sensing for ISR Applications*, 2006.

[11] I. N., R. B., G. R., L. B., G. D., H. R., M. K., S. L., T. G., and N. B., "A Scalable Testbed For Emulating Wireless Mobile Ad-Hoc Networks," in *Military Communications Conference*, 2007.

[12] M. J. Chao W. and W. J, "NRL Mobile Network Emulator."