This document is supplied to DSTL under DEFCON 703

SAPIENT
Interface Control Document

Dr Gillian Marshall and D.A.A. Faulkner
QINETIQ/TIS/C4ISR/ICD1400047/v4.1
27th November 2015

© Crown copyright (2015), Dstl. This material is licensed under the terms of the Open Government Licence except where otherwise stated. To view this licence, visit http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3 or write to the Information Policy Team, The National Archives, Kew, London TW9 4DU, or email: psi@nationalarchives.gsi.gov.uk
# Administration page

## Customer Information

<table>
<thead>
<tr>
<th>Customer reference number</th>
<th>CCS TIN 3-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project title</td>
<td>SAPIENT Lead Systems Integrator</td>
</tr>
<tr>
<td>Customer Organisation</td>
<td>Dstl</td>
</tr>
<tr>
<td>Customer contact</td>
<td>Paul Thomas</td>
</tr>
<tr>
<td>Contract number</td>
<td>DSTLX-1000086622 (TIN 3-20)</td>
</tr>
<tr>
<td>Milestone number</td>
<td>Milestone 1</td>
</tr>
<tr>
<td>Date due</td>
<td>31&lt;sup&gt;st&lt;/sup&gt; October 2014</td>
</tr>
</tbody>
</table>

## Principal author

<table>
<thead>
<tr>
<th>Name</th>
<th>Dr G.F. Marshall 01684 895730</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QinetiQ Ltd, Malvern Technology Centre, St Andrews Road, Malvern, WR14 3PS</td>
</tr>
<tr>
<td></td>
<td>D.A.A. Faulkner 01684 894871</td>
</tr>
<tr>
<td></td>
<td>QinetiQ Ltd, Malvern Technology Centre, St Andrews Road, Malvern, WR14 3PS</td>
</tr>
</tbody>
</table>

## Release Authority

<table>
<thead>
<tr>
<th>Name</th>
<th>Mrs V. Snow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post</td>
<td>Project Manager</td>
</tr>
<tr>
<td>Date of issue</td>
<td>See Record of Changes</td>
</tr>
</tbody>
</table>

## Record of changes

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Detail of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>10&lt;sup&gt;th&lt;/sup&gt; Jan 2014</td>
<td>First issue</td>
</tr>
<tr>
<td>1.1</td>
<td>14&lt;sup&gt;th&lt;/sup&gt; Jan 2014</td>
<td>For distribution to SAPIENT project</td>
</tr>
<tr>
<td>2.0</td>
<td>31&lt;sup&gt;st&lt;/sup&gt; March 2014</td>
<td>Second issue, providing further clarification</td>
</tr>
<tr>
<td>3.0</td>
<td>7&lt;sup&gt;th&lt;/sup&gt; November 2014</td>
<td>Third Issue for Review</td>
</tr>
<tr>
<td>3.1</td>
<td>18&lt;sup&gt;th&lt;/sup&gt; December 2014</td>
<td>Revised following Review</td>
</tr>
<tr>
<td>4.0</td>
<td>22&lt;sup&gt;nd&lt;/sup&gt; October 2015</td>
<td>Minor revisions following September 2015 Demonstration</td>
</tr>
<tr>
<td>4.1</td>
<td>27&lt;sup&gt;th&lt;/sup&gt; November 2015</td>
<td>Updated for wider release</td>
</tr>
</tbody>
</table>
Abstract

This document is the Interface Control Document (ICD) for the SAPIENT (Sensing for Asset Protection using Integrated Electronic Network Technology) asset protection project. It defines the interfaces between the High Level Decision Making Module (HLDMM), the Autonomous Sensor Modules (ASM), and the middleware and Graphical User Interface (GUI) provided by the Lead Systems Integrator (LSI). The purpose of this document is to enable individual module developers to build component modules compatible with the overall system. This fourth issue of the ICD provides further clarification of the details of the xml schema, following completion of the SAPIENT Project Phase 2 Final Demonstration.
# List of contents

1 Introduction  
   1.1 The SAPIENT Project 9  
   1.2 Purpose of the document 9  
   1.3 Scope of the document 9  
   1.4 Intellectual Property Considerations 10  

2 System Overview  
   2.1 Physical Interconnection 10  
   2.2 Data Interface 11  
   2.2.1 File Share 11  
   2.3 Messages passed via XML 11  
   2.3.1 Null Termination 11  
   2.4 System Clock 12  
   2.5 Power Supplies 12  

3 Architecture Design 12  

4 System Architecture 14  
   4.1 Ethernet Switches 15  
   4.2 Data Agents VM 15  
   4.3 QinetiQ ASM Sensor, QinetiQ ASM Engineering PC 15  
   4.4 HLDMM Data Agent VM 16  
   4.5 LSI Engineer PC 16  
   4.6 Database VM 16  
   4.7 GUI PC 16  

5 Interface Description Overview 16  
   5.1 Message Descriptions 16  
   5.2 High level process walkthrough 16  
   5.2.1 Initialisation 16  
   5.2.2 Normal Operation 17  
   5.2.3 Control messages 17  
   5.2.4 ASM Shutdown 17  
   5.3 Initialisation 18  
   5.3.1 Message Protocol 18  
   5.3.2 Location Information 18  
   5.3.3 Sensor ID / Source ID 18  
   5.3.4 Sensor ID Allocation 19  
   5.3.5 Initial Heartbeat message 19  
   5.4 Ongoing messages from ASMs during normal operation 19  
   5.4.1 Heartbeat Messages 19  
   5.4.2 Detection Messages 20  
   5.4.2.1 Object ID 20  
   5.4.3 Alert Messages 20
5.4.4 Message Protocol
5.5 Control Messages
5.5.1 Sensor Modes
5.5.2 Task Control
5.5.3 Message Protocol
5.6 Acknowledgement Messages
5.7 Alert Response Messages
5.7.1 Reject Responses

6 Guidance on using the xml schema
6.1 General Guidance
6.1.1 Location Information
6.1.2 Time Information
6.2 Sensor Initialisation
6.3 Heartbeat Messages
6.3.1 Sensor Mode
6.3.2 Sensor Location
6.3.3 Sensor Field of View
6.3.4 Sensor Coverage
6.3.5 Obscurations
6.3.6 Status Reporting
6.4 Detection Messages
6.5 Control Messages
6.5.1 Task ID
6.5.2 Region-based Tasks
6.5.3 Default Commands
6.5.4 ClassificationThreshold Command
6.5.5 LookAt Command
6.5.6 LookFor Command
6.5.7 Mode Command
6.6 Acknowledgement Messages
6.7 Error Messages
6.8 Alert Messages
6.9 Further Guidance on XML fields
6.9.1 Text Fields
6.9.2 Class Types
6.9.3 Human Sub Class Types
6.9.4 Vehicle Sub Class Types
6.9.5 Behaviour Types
6.9.6 Reporting Classes, Sub-classes and Behaviour
6.10 Managing Real-Time System Performance

7 XML Schemas
7.1 Initialisation
7.2 Heartbeat Messages
7.3 Detection Messages
7.4 Control Messages
7.5 Acknowledgement Messages
7.5.1 Registration Acknowledgement
7.5.2 Control Acknowledgement
7.5.3 Null Response to ‘LookAt’ command
7.6 Error Messages
7.7 Alert Messages
7.8 Alert Response Messages

8 Example Messages
8.1 Example Registration Message
8.2 Example Heartbeat Messages
8.2.1 Initial heartbeat message
8.2.2 Additional Obscuration Detected, Illumination Level changed
8.2.3 Status Unchanged
8.3 Example Detection Messages
8.3.1 Example Detection
8.3.2 Track Lost from View
8.3.3 Track Lost in View
8.4 Example Control and Acknowledgement Messages
8.4.1 Default Tasking
8.4.2 Acknowledgement Message for Default Task
8.4.3 Task 1 to over-ride Default tasking
8.4.4 Acknowledgement Message for Task 1
8.4.5 Stop Task 1 and return to default task
8.4.6 Acknowledgment of Stop Message
8.4.7 Set Detection Threshold High Command
8.4.8 Acknowledgement of Successful Threshold High Command
8.4.9 Acknowledgement of Rejected Threshold High Command
8.4.10 Request Reset Command
8.4.11 LookAt Command
8.4.12 Detection Report response to LookAt Command
8.4.13 Null Response to LookAt Command

9 HLDMM – ASM Database Interaction
9.1 ASM – HLDMM Message Protocol Summary
9.1.1 Initialisation
9.1.2 Ongoing Messages from ASMs during normal operation
9.1.3 Control Messages
9.2 Initialisation
9.2.1 Sensor_Registration Table
9.3 Heartbeat Message
9.3.1 Status_Report Table
9.3.2 Status_Report_Region Table
9.3.3 Status_Report_Range_Bearing Table 61
9.3.4 Status_Report_Messages Table 61
9.4 Detection Message 62
9.4.1 Detection_Report_Location Table 62
9.4.2 Detection_Report_Range_Bearing Table 63
9.4.3 Detection_Report_Track_Info Table 64
9.4.4 Detection_Report_Class Table 65
9.4.5 Detection_Report_Subclass Table 65
9.4.6 Detection_Report_Behaviour Table 66
9.4.7 Detection_Report_Assoc_File Table 67
9.5 Control Message 67
9.5.1 Sensor_Task Table 67
9.5.2 Sensor_TaskAck Table 68
9.6 Alert Message 68
9.6.1 Alert Table 69
9.6.2 Alert_Location Table 69
9.6.3 Alert_Range_Bearing Table 70
9.6.4 Alert_Assoc_File Table 71
9.6.5 Alert_Assoc_Detection Table 71
10 Interface between HLDMM and GUI 72
10.1 HLDMM – GUI Database 72
10.1.1 Initialisation 72
10.1.2 Heartbeat Message 72
10.1.3 Detection Messages 73
10.1.4 Control Messages 73
10.1.5 Alert Messages 73
10.2 Tracks and Detection Points 73
10.3 Alerts 74
10.4 Sensor Field of View and Coverage 74
10.5 High-Level Tasking and Regions of Interest 74
10.6 Operator Response to Alerts 74
A Cross Reference Table for XML fields 75
B SAPIENT Phase 2 Demonstration-Specific Information 80
B.1 SAPIENT Phase 2 Demonstration-Specific Architecture 80
B.1.1 Physical Connection 80
B.1.2 Firewalls and Network Security 80
B.1.3 Database 80
B.1.4 Additional GUI PCs 80
B.1.5 Sensor ID Allocation 81
B.1.6 Sensor Location 81
B.2 ASM Database Configuration 81
B.2.1 Demonstration Configuration 81
B.2.2 Database Server Installation 81
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.2.3</td>
<td>Database Creation</td>
<td>81</td>
</tr>
<tr>
<td>B.2.4</td>
<td>HLDM Database access</td>
<td>81</td>
</tr>
<tr>
<td>B.3</td>
<td>File Share</td>
<td>81</td>
</tr>
<tr>
<td>C</td>
<td>Initial distribution list</td>
<td>83</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 The SAPIENT Project

Sensing for Asset Protection, using Integrated Electronic Networked Technology (SAPIENT) is a programme for the development of autonomous collaborative distributed sensing for asset protection. The objectives for the SAPIENT programme are:

- Protection of land based high-value assets with defined borders;
- Reduction in operator workload during monitoring activities;
- Improved autonomy for decision making (automatic identification of threat activities to reduce false alarm rates, allowing for autonomous context driven decision making);
- Improved autonomy for sensor management.

The SAPIENT concept considers a system of systems approach, consisting of multiple Autonomous Sensor Modules (ASM) that are all connected to a single High Level Decision Making Module (HLDMM).

1.2 Purpose of the document

This document concisely describes the interface specification for the SAPIENT project.

This Interface Control Document (ICD) has been up-issued following the successful demonstration of the SAPIENT project.

1.3 Scope of the document

This document covers the interfaces between the Autonomous Sensor Modules (ASM), the High Level Decision Making Modules (HLDMM) and the middleware and Graphical User Interface (GUI) provided by the Lead Systems Integrator (LSI).

The SAPIENT concept is intended to support a range of different sensing modalities, a significant range of levels of object detection, localisation and classification, and a range of interpretations of what “autonomy” means. In developing this ICD, a balance has therefore been struck between the specific needs of the SAPIENT demonstration system and a desire to provide a flexible and extensible interface that could cope with unknown future sensor and processing modules in potential future iterations of the SAPIENT system beyond the current programme.

The interfaces between the ASMs and the middleware are covered in the main body of the document up to Section 8. Section 9 defines the interface between the HLDMM and the ASM database in the middleware. Section 10 defines the interface between the HLDMM and the GUI. Appendix A contains a list of all the XML data fields and their usage. Appendix B contains the implementation-specific information from the SAPIENT Phase 2 Demonstration September 2015.
1.4 Intellectual Property Considerations

This Interface Control Document has been produced under DEFCON 703, and hence is Crown Copyright. It draws heavily on the background and foreground Intellectual Property (IP) of the eight ASM suppliers and two HLDMM suppliers from Phase 1, and the down-selected suppliers (four ASMs and one HLDMM) in Phase 2. All Phase 1 suppliers have provided details of their proposed system inputs and outputs to the LSI under a set of two-way Non-Disclosure Agreements (NDA), with contractual obligations on both sides to share the work with Dstl. Phase 2 suppliers have shared their system details under a multi-way NDA with each other and the LSI, again with contractual obligations to share the work with Dstl.

It is the nature of an interface specification that it needs to describe fully all of the possible interactions across the interface: the HLDMM supplier in particular needs to understand the types of information it could receive from each of the candidate ASMs. Care has therefore been taken to generalise the message specifications wherever possible, and to anonymise the details of what the sensors can provide, so that there is no traceability from this document to any individual sensor’s capabilities.

In conclusion, although the interface specification has been derived from discussions with multiple suppliers under a number of different contracts, this resulting Interface Control Document is provided as DEFCON 703, and is therefore releasable at MOD’s discretion both within and outside the SAPIENT project.

2 System Overview

The SAPIENT system architecture is based around a central database, which each ASM will populate with declarations and detections, including as much information as the sensor modules are capable of providing. NB This is not intended to include streaming raw data. A data agent will manage the data-flow into the database. This will allow the data agent to monitor and if necessary limit data bandwidth. The ASMs will communicate with the data agent using messages as described in section 2.2.

The HLDMM will query the database, and perform higher-level fusion and reasoning based on the data therein. The HLDMM will task the ASMs via the HLDMM data agent. This will pass the task on to appropriate ASM via the data agent.

Separately, the HLDMM will populate another database table via the HLDMM data agent with data describing fused tracks and alerts. This data will drive a GUI to provide overall situational awareness during operation. The specifics of the interface to the GUI should not be relevant to the ASM providers. A high-level view of the interface to the GUI is provided in Section 10.

2.1 Physical Interconnection

The physical interconnect technology that will be used is Ethernet, with each ASM providing a single connection into the middleware. The network was implemented in the SAPIENT project using a specific approach discussed in Appendix B.1. This is likely to be implementation specific.
If an ASM consists of a network of multiple sensors and/or processing nodes, the ASM supplier will be responsible for providing an appropriate bridge onto the LSI-supplied Ethernet, such that each ASM has a single connection to the wider system.

2.2 Data Interface

The main data interface between the ASMs and the LSI-supplied Data Agent will be a TCP socket. The Data Agent will be the server, the ASM the client. The only valid data will be XML messages as defined in this document. The only exceptions are certain additional data files that are shared on a File Share described below. It is the ASM's responsibility to monitor the connection and attempt to reconnect if it detects a loss of connection. If the ASM is unable to connect to the Data Agent, then it should attempt to retry every 10 seconds until it is successful.

It is anticipated that there will be a separate instance of the Data Agent for each instance of an ASM. This will allow the Data Agents to be spread across multiple servers if required. The IP address and port of the Data Agent will be provided by the LSI at integration time. A suggested port range would be 14001-14100 for the Data Agents to listen on. This range is not expected to conflict with other applications, but if a conflict is found an alternative will be assigned as required. A single listening port would be used for each Data Agent. Ephemeral Ports as assigned by the server and client operating systems will be used to complete the connections.

It is recommended that the TCP_NO_DELAY flag is set for all socket connections to minimise the latency through the system.

2.2.1 File Share

The LSI will provide one or more file shares for sending additional data files from the ASMs to the GUI. This is intended solely to provide the system operator with additional information such as image snapshots that they would need to respond correctly to an alert generated by the system. It is intended that the autonomous behaviour of the system should be encapsulated within the XML messages of the main data interface.

The File Share was implemented in the SAPIENT project using a specific approach discussed in Appendix B.3. This is likely to be implementation specific.

2.3 Messages passed via XML

The LSI has defined an xml schema (see section 7) for each ASM to populate with its detections and declarations. The advantage of this approach is that it is readily extendable, and also is easy to adapt for the varying levels of information provided by each ASM. No individual sensor will be expected to populate all the fields, however, every ASM will be expected to populate a small sub-set of these fields.

2.3.1 Null Termination

To improve the reliability of the system in the presence of communication errors, each XML message will be terminated by a null (0) character. This allows the system to reject incomplete or corrupted messages.
2.4 System Clock

The LSI will provide a Network Time Protocol (NTP) Server on the SAPIENT network. All modules will be expected to synchronise with this at regular intervals during system operation. The HLDMM requires timing accuracy to the nearest 100ms. This should be possible assuming the network devices can support it. It is anticipated that an interval of between 15 and 60 minutes between synchronisation requests for each module should be sufficient to maintain time accuracy. Given the variable clock-drift of different devices, this interval should be chosen as appropriate by the ASM supplier.

2.5 Power Supplies

Individual modules may be either battery powered (batteries to be supplied by and remain the responsibility of the module supplier) or mains powered (single phase 230V, 50Hz).

3 Architecture Design

![Architecture Diagram](image)

Figure 1: Architecture diagram illustrating data flow from ASMs via LSI middle-ware to HLDMM and GUI. The red line outlines the area of responsibility of the LSI.
**Figure 2:** Data Flow: Detection reports are generated by the ASM, passed to an LSI-controlled ASM Data Agent, and thence into the LSI-supplied Database. The HLDMM directly queries the database. When the HLDMM generates an alert that it wishes to raise up to the operator, it passes a message to the LSI-controlled HLDMM Data Agent, which populates the database, which in turn is queried by the LSI-supplied GUI.

Figure 1 and Figure 2 illustrate the data flow between the ASMs, the LSI-supplied middle-ware and the HLDMM in normal operation. The ASMs are decoupled from the database to allow middleware to control data flow and to allow future expansion of the database. The HLDMM has direct access to the database for read but not for write so that responsibility for the integrity of the database lies with LSI.

**Figure 3:** Architecture diagram illustrating control flow from the GUI and HLDMM to the ASMs. As before, the red line outlines the boundary of the LSI’s responsibility.
Figure 4: Control flow to the ASMs from the HLDMM. Provision has been made for an operator to provide high level instructions from the GUI, via the database, to inform the HLDMM of the “Commander’s Intent”, e.g. at mission start-up. In normal operation, most tasks and instructions will be generated autonomously by the HLDMM, and passed via the HLDMM Data Agent and the ASM Data Agent to the appropriate ASM.

Figure 3 and Figure 4 illustrate the control flow from the GUI to the HLDMM and from the HLDMM to the ASMs.

4 System Architecture

The LSI has responsibility for deploying and configuring the network infrastructure so that the ASM and HLDMM components can connect to the wider system.
Figure 5: SAPIENT LSI System Architecture

Figure 5 shows the proposed system architecture of the SAPIENT system. The area inside the red outline is the responsibility of the LSI and consists (at a high level of description) of the following:

4.1 Ethernet Switches

Gigabit Ethernet network switches and cabling to connect the servers that host the various VMs, the QinetiQ ASM PCs, the Firewalls to the third-party ASM and HLDMM supplied equipment and the Remote GUI PC. These will also be deployed in the remote viewing room.

4.2 Data Agents VM

This is a Virtual Machine for hosting the Data Agent software for several or all of the ASMs. If required for performance reasons, the Data Agents can be split across multiple VMs.

4.3 QinetiQ ASM Sensor, QinetiQ ASM Engineering PC

These are PCs required for the QinetiQ ASM.
4.4 HLDMM Data Agent VM

This is a Virtual Machine for hosting the Data Agent software for the HLDMM supplier.

4.5 LSI Engineer PC

This is a PC used to configure and monitor the LSI deployed VMs.

4.6 Database VM

This is a VM for hosting the database used to store the ASM data and the database used to store the fused data to be displayed on the GUI.

4.7 GUI PC

This is a Workstation PC with dual displays. The first display will show a map-based situational awareness picture overlaid with sensor detections and alerts. The second display will show an alert list and text messages from the HLDMM.

5 Interface Description Overview

5.1 Message Descriptions

All interaction between the ASMs and the HLDMM will be via the LSI-provided database, and will be in three stages.

1. Initialisation;
2. Ongoing messages during normal operation;
3. Control messages from the HLDMM to individual ASMs;

These will now be described in more detail.

5.2 High level process walkthrough

5.2.1 Initialisation

- ASM connects as a client to LSI-supplied data agent;
- ASM sends Registration message to HLDMM via LSI-supplied data agent;
- Data agent sends acknowledgement message to ASM; ASM sends SensorTask message to HLDMM via LSI-supplied data agent defining its default tasking (if any);
- ASM sends initial heartbeat message defining its initial field of view (if available);
- EITHER
  - ASM begins activity detection as per its default tasking (for those sensors which have defaults);
• OR
  o HLDMM describes default tasks and uses the protocol in section 5.2.3 to instruct the ASM via the data agent;
  o ASM begins activity detection.

NB: If the HLDMM defines a default task for an ASM, that task over-rides any ASM-generated default task previously defined.

5.2.2 Normal Operation

• ASM performs activity detection;
• ASM sends detection messages as per current tasking;
• ASM sends regular heartbeat messages as outlined in Registration Message;
• HLDMM queries the database, and reasons over/fuses the detections;
• HLDMM populates the GUI-database via the LSI-supplied HLDMM data agent with data to be displayed to the operator;
• If HLDMM requires a change in ASM behaviour, a control message will be sent.
• If ASM wishes to inform the HLDMM that it is about to do something, the alert message can be used to inform or alert the HLDMM of this. An example of this might be a sensor deciding to do a detailed scan of an object or area.
• If the alert message merits a response from the HLDMM, the Alert Response message can used.
• If the HLDMM chooses to deny permission to carry out the action, the HLDMM will respond with an Alert Response message with ‘status’ field of value ‘Reject’. The ‘reason’ field may be used to provide a strategy for retrying the action in future. (See section 5.7.1)

5.2.3 Control messages

• HLDMM sends Control Message to ASM via LSI-supplied data agent;
• ASM sends acknowledgement of message to HLDMM via data agent – confirming acceptance of tasking or command;
• ASM changes detection behaviour as per command;
• ASM returns to Normal Operation.

5.2.4 ASM Shutdown

• ASM sends heartbeat message with “goodbye” tag to show impending loss of communication from this ASM;
• ASM closes network connection to data agent.
5.3 Initialisation

The initial stage will happen on start-up. As part of the system initialisation, each ASM will be expected to produce an initialisation message stating what its capabilities are. This will include sensor type, supported detection modes, a list of what it is able to detect, track and/or classify, a declaration of error characteristics associated with the detections/tracks/classifications, a description of the contents of its "heartbeat" message (see below) and a list of how it is able to interact with the HLDMM – e.g. taskings it is able to accept. The message will also include the performance implications of different detection modes. Performance can be specified in terms of geometric (location) error, detection performance and classification performance.

The initialisation messages will contain declarations of sensor capability. An initial outline of the xml schema for the initialisation declarations is given in section 7.1. The message will only need to be passed once, when the ASM process connects to the system.

5.3.1 Message Protocol

The Initialisation Message shall be sent by the ASM to the Data Agent before any other messages. No other messages shall be sent until an acknowledgement message or error message is received in response.

The Data Agent shall forward the message to the HLDMM Data Agent where if required, a sensor ID will be allocated to this ASM and the acknowledgement message sent back to the ASM via the Data Agent.

If the ASM does not receive an acknowledgement message within 30 seconds of sending the initialisation message, the ASM should close the existing socket connection, open a new socket connection to the Data Agent and send the message again.

The Initialisation message will be forwarded on to the HLDMM by the HLDMM Data Agent. It will also be stored in the database for future reference.

5.3.2 Location Information

The Initialisation message is used to declare the real-world coordinates system that the ASM will provide location information to the SAPIENT system in all subsequent messages. Section 6.1.1 provides guidance on how location information should be provided.

5.3.3 Sensor ID / Source ID

Each module instance must use a single, unique module identifier number in all messages. The purpose of this identifier is to allow the middleware and HLDMM to identify the origin of messages. Registration messages, control messages and their acknowledgements are specific to an ASM and so ‘sensorID’ is used in these messages. Heartbeat, detection and alert messages can be generated by ASMs or HLDMMs and so ‘sourceID’ is used in these messages. The same identifier number shall be used across all messages. High-Level Control messages from the GUI to the HLDMM shall use ‘sensorID’ value zero (0).
5.3.4 Sensor ID Allocation

The ICD allows for the Sensor ID identifier to be either pre-assigned for each ASM or allocated during the initialisation phase by the HLDMM Data Agent. For the first approach, the pre-assigned sensor ID is populated in the registration message. For the second approach, the ‘sensorID’ element is left empty in the registration message. In both cases, the value to use for all subsequent messages is returned in the registration acknowledgement message.

The first, simpler approach may be more appropriate where a fixed set of sensors is deployed for a prolonged period or a consistent configuration is required. The second approach may be more appropriate where the set of sensors may be changed and “plug-and-play” is required. In principle, the two approaches can both be used in the same system but this is likely to be harder to maintain than adopting one or other approach for all ASMs.

5.3.5 Initial Heartbeat message

As part of system initialisation, an initial heartbeat message will be sent after the initialisation message. This will indicate the initial state, physical location and field of view of the ASM. Those ASMs which contain a GPS or other real-world locating capability will be expected to use it to provide the physical location of the sensor as part of an initial heartbeat message in the initialisation sequence.

5.4 Ongoing messages from ASMs during normal operation

There will be three main forms of regular messages from the ASM which will be communicated using the defined xml taxonomy (see section 7) and input by the LSI into a common database. Each message type will have its own tables in the database.

5.4.1 Heartbeat Messages

These will be regular messages output by each ASM. This message will contain a time-stamp and the sensor’s unique ID, and will provide the HLDMM with reassurance that the sensor is still operating correctly, even in the absence of any valid detections. In addition, this message will provide additional information as agreed with each of the ASM suppliers (and as declared in advance in their initialisation script). This information could include for example, some of the following fields:

- Sensor location, field of view; coverage; sensor mode; obscuration polygon(s); rain detected; degraded sensor performance; remaining battery power; camera exposure time.

In addition to regular heartbeat messages, one-off heartbeat messages may be sent immediately on important changes. Examples of this might include the following:

- ASM tamper, ASM fatal error, ASM completely obscured

It is recommended that a regular heartbeat should be sent at least once every 10 seconds. Regular heartbeats shall be sent at least once every 60 seconds and not more than once a second. If no heartbeat or detection messages are received by the Data Agent within a 60 second period, the Data Agent will close the socket connection and wait for the ASM to re-establish communication.
At least the latest heartbeat information will be stored by the LSI in the database. Storing some historical heartbeat information may be useful for diagnostic purposes, but to limit redundant data, this may be limited to where information has changed.

An initial outline of the xml schema for heartbeat messages is given in section 7.2.

5.4.2 Detection Messages

These messages will be output by each ASM when they wish to declare an object detection or other result (e.g. classification) to the wider system. The message declaration will contain the sensor’s unique ID, a timestamp, an object unique ID number, the real-world location of the detected object (at least 2D location, some sensors will be able to produce 3D locations and/or location error ellipses), and as much additional information about the detected object as the sensor can provide (and in line with the declaration it made in the initialisation script). This could include for example the detection confidence level, track information, and/or classification information (with confidence levels). There may also be additional information associated with a detection, such as an image snap-shot or point-cloud. The format of the messages from all of the ASMs shall be consistent with the given xml schema, however the actual content of the messages will potentially be different for each ASM, depending on its capabilities. No individual sensor module will be expected to fill in all the possible fields – each ASM will provide as much information as it can, as described in its registration message, and the HLDMM will be expected to reason across it and fuse it appropriately.

An initial outline of the xml schema for the detection messages is given in section 7.3.

5.4.2.1 Object ID

Object ID is an integer number generated by the ASM that uniquely identifies an object. If an ASM is not capable of associating multiple detections of the same object then the object ID should be unique to each detection.

When an object remains static in the field of view for a long time such as a vehicle parking up, it is assumed that it will merge into the background in time. If detected as such, it should be reported as “lostInView”. If supported, it may then be reported as an obscuration.

5.4.3 Alert Messages

These provide a means for an ASM to report something that needs a response from the HLDMM. An example might be that an ASM wishes to autonomously change mode, but would like permission from the HLDMM. Modes requiring permission should include modeParameter type="PermissionToChange" value="Required" in the registration message.

Alert messages also provide the means for the HLDMM to present alerts to the user via the GUI so that the user will provide a response to the system.

They can also be used to provide extra information directly from the ASM to the GUI such as for an engineering GUI. In this case no response would be expected. Finally, if the system includes sensors that can only provide alerts, such as door alarms, these can be reported using Alert Messages. Where a response is required an Alert Response message is used.
5.4.4 Message Protocol

Ongoing messages from ASMs, (Detections, Heartbeats and Alerts) shall be stored in the database by the Data Agent for the HLDMM to read. If an alert message requires an immediate response, such as when an ASM wishes to autonomously change mode, but would like permission from the HLDMM, the message will be additionally forwarded on to the HLDMM via the HLDMM Data Agent.

5.5 Control Messages

Control messages allow the HLDMM to task individual sensor modules. The particular message information will be very much dependent on the capabilities of the individual ASM, as described in the initialisation message. However, the general form will be request ID, sensor ID, and a tasking request, for example

- In Region X look for threats of Type Y;
- Ignore all detections in Region Z;
- Another sensor has detected something at location (X,Y), can you confirm the detection, track the object and classify the activity. This is effectively a cross-cue from one sensor to another;
- Your false alarm rate is unacceptably high (or your P(d) is unacceptably low) – adjust your thresholds accordingly.

The ASM will be required to produce an acknowledgement of the tasking request (or if it is unable to comply, give an error message) – see section 5.6.

An initial outline of the xml schema for these sensor tasking and control messages is given in section 7.4.

When a task completes or is stopped by a control message, the ASM will revert to any previously defined task (such as the default one) or if there are none it will stop detecting until it receives a new control message.

5.5.1 Sensor Modes

An ASM will operate in one or more sensor modes. A sensor mode is defined as a distinct way of operating such that the method of operation, the detection and error characteristics as reported in the initialisation message remain consistent. Typically a non-steerable sensor will operate in a single sensor mode. A steerable sensor could have multiple modes that define how it will respond to tasking from the HLDMM. E.g. a PTZ camera could have one mode where it continually scans an area and another where it lingers at a number of fixed locations. The initialisation message provides the means to communicate the characteristics of the mode. The control message provides the means to tell the ASM to change its mode.

5.5.2 Task Control

The control field in the control message defines what action to perform on that task.
- Start - Initialises and schedules or starts the task or restarts a paused task;
- Stop - stop the task, remove its definition, revert to previous task;
  NB. A stop message does not require a ‘region’ or ‘command’ section.
5.5.3 Message Protocol

Control messages are sent from the HLDMM to the HLDMM Data Agent which stores the message in the database. They are then routed to the Data Agent for the ASM specified in the message which forwards the message on to the ASM. The ASM then shall respond with an acknowledgement message to accept or reject the task in the control message. The acknowledgement message is sent back to the Data Agent by the ASM which forwards the message on to the HLDMM Data Agent. The HLDMM Data Agent forwards the acknowledgement on to the HLDMM and updates the message record in the database with the acknowledgement information.

5.6 Acknowledgement Messages

Acknowledgement Messages will be used to complete the handshake between the components of the system. These are only used in response to specific messages. There are three types of ACK message:

- Acknowledgement from the HLDMM to the ASM that registration has occurred (see section 7.5.1);
- Acknowledgement from the ASM that it has received a sensor tasking message from the HLDMM (see section 7.5.2);
- The ASM provides a null response to a LookAt message – i.e. the ASM had a look at the required coordinates, but was unable to detect an object there (see section 7.5.3, and sections 8.4.11 to 8.4.13 showing an example of how to use this message).

5.7 Alert Response Messages

Alert Response Messages are used where a response is required to an Alert Message. An example of this might be a sensor deciding to do a detailed scan of an object or area and it wants to ask permission from the HLDMM first. The HLDMM would send an Alert Response message with ‘Acknowledge’ or ‘Reject’ status field. Modes requiring permission should include modeParameter type="PermissionToChange" value="Required" in the registration message.

5.7.1 Reject Responses

Where an Alert Response Message has ‘Reject’ in the status field, the reason field can be used to allow the HLDMM to inform the ASM when it can try to send this request again. An empty reason field implies that this request should not be sent again. To allow the ASM to send this request again use ‘Retry:' in the reason field, followed by the time interval and time units.

e.g. ‘Retry:10,seconds’
6  Guidance on using the xml schema

6.1  General Guidance

The SAPIENT system potentially contains a wide range of sensor types and capabilities. The XML Schema has been written in such a way that it should be fit for purpose not only for the modules down-selected for the SAPIENT Demonstration System but also for future sensors which could be subsequently developed and integrated. It is recognised that no one sensor will be expected to fill in all of the fields. Additional guidance is given in Appendix A as to which fields are compulsory and which are optional.

6.1.1  Location Information

To support a variety of sensor types, the SAPIENT system will accept location information in a number of different real-world coordinates systems. For example, object locations in GPS coordinates, UTM, or range/bearing/elevation from a known sensor position could all be acceptable, depending on what is the native format of the individual ASM. To keep this flexibility bounded, this will be limited to Cartesian (x,y,z coordinates, together with a corresponding reference frame, either UTM or GPS) or Range-bearing. This could be easily extended with additional coordinates systems in future if required.

There are a number of different types of locations provided in the XML messages; (i.e. detections, sensor location, sensor field of view, areas of obscuration, regions). Table 1 provides guidance on which coordinates system to use for the different types of location. Where a location type is supported by an ASM, the coordinates system used shall be declared in the Initialisation message. This coordinates system shall be used consistently for all messages that contain location data of that type. Non-selected coordinates systems shall then be ignored entirely. For Cartesian coordinates, the ASM shall use the same reference frame for all location types.

Table 1: Coordinates Systems for different types of location information

<table>
<thead>
<tr>
<th>Location Type</th>
<th>Coordinates System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detections</td>
<td>Any</td>
</tr>
<tr>
<td>Sensor Location</td>
<td>Cartesian</td>
</tr>
<tr>
<td>Sensor Field of View and Coverage</td>
<td>Any (Range-Bearing may be preferable for steerable sensors)</td>
</tr>
<tr>
<td>Areas of obscuration</td>
<td>Any</td>
</tr>
<tr>
<td>Regions</td>
<td>Any – Cartesian preferred because regions may be independent of sensors</td>
</tr>
</tbody>
</table>

All location reports shall provide real-world coordinates in the chosen reference frame, and any object sizes, speeds etc. reported shall also be given in real-world units. Internal frames of reference (e.g. camera-centric, measured in terms of pixels) should be calibrated and translated into real-world values (with associated errors if appropriate) before being reported.

The HLDMM shall task the individual ASMs using the chosen coordinate system of the ASM as specified by the TaskDefinition section of the Registration message.
All azimuth bearings shall be reported relative to north, Grid North should be used by default but magnetic or true north can be specified in the initialisation message.

6.1.2 Time Information

Note that all timestamps are in UTC Zulu using ISO8601 format. Most messages are time-stamped down to the current second. Detection and heartbeat messages have sub-second resolution, to ease the task of the HLDMM in cross-correlating them. The SAPIENT system will provide a network time reference, Time information shall be synchronised to this time reference.

6.2 Sensor Initialisation

On first power-up, the ASM will send a file to the HLDMM explaining its capabilities, selected as a sub-set of the fields provided in the Initialisation Schema. For any given declaration where the option Yes/No is provided in the schema, a “No” response is equivalent to omitting the field altogether. Care should be taken to ensure that fields declared in the initialisation message are the only ones subsequently used in the other message types.

6.3 Heartbeat Messages

This file is sent regularly to give the HLDMM information about the ASM operating mode, and to provide confidence that it is still working correctly (or conversely that performance can be expected to degrade). Only fields relevant to the particular sensor should be included; all others should be omitted. So, for example, a radar sensor would not report on its exposure or white balance, and a mains-powered sensor would not report battery level.

6.3.1 Sensor Mode

Sensor mode is an optional field reported where a sensor provides multiple modes or states. E.g. Transition, Stopped, Failed, Default, 3D Scanning, Scan and Search. If omitted, then ‘Default’ mode will be assumed. Normally, the ASM will work in ‘Default’ mode but it can be forced into other modes specified in the initialisation message. When a sensor is "on the move" and not detecting it should report as ‘Transition’. If an ASM has received a ‘Stop’ Request command, it should report ‘Stopped’. If a fatal error has occurred, then ‘Failed’ should be reported.

6.3.2 Sensor Location

Sensor location is an optional field that reports the current geographical location of the sensor. Typically, this will only be reported during initialisation.

6.3.3 Sensor Field of View

Sensor Field of View is an optional field that reports the current field of view of the sensor. Typically with a fixed sensor, this will only be reported during initialisation. For steerable sensors, this will be reported regularly.

6.3.4 Sensor Coverage

Sensor Coverage is an optional field that reports the extent of the area a sensor can cover but not necessarily all at once. Typically, this will only be reported for
scanning or steerable sensors. For example, a Pan Tilt Zoom camera may have 360 degree coverage but only a field of view of 45 degrees.

6.3.5 Obscurations

Obscuration is an optional field that can define one or more regions that the sensor can report as being unable to detect within. This refers to the region beyond a static obstacle, rather than the region in the fov beyond a moving target. If the HLDMM wants to know what is obscured behind a moving target, it can work it out itself from the reported object size and location in the detection report.

6.3.6 Status Reporting

The ‘Status’ element of the heartbeat message provides a way for ASMs to report conditions that may affect the performance of this or other ASMs. The ‘Level’ field indicates the severity of the condition.

For example, the following indicates that rain has been detected that does not affect this sensor but may affect other sensors.

```
<status level = "Information" type = "Weather" value = "Rain" />
```

The following indicates that the performance of the ASM has been degraded by external conditions e.g. rain, snow, but it is still able to perform its current tasking.

```
<status level = "Warning" type = "ExternalFault" />
```

The following indicates that the performance of the ASM has been degraded by external conditions e.g. illumination, and so it is unable to perform its current tasking:

```
<status level = "Error" type = "ExternalFault" value = "Illumination" />
```

If a fatal error has occurred then report "Failed" under sensor mode as follows:

```
<sensorMode> Failed</sensorMode>
```

6.4 Detection Messages

This message is sent by the ASM to report each detection (or, optionally, group of detections or track). Track data is an optional element used to provide the HLDMM with more information about the movement of an object, where available. Where sensor performance allows, classification information should also be included for each detected object.

6.5 Control Messages

These are messages sent by the HLDMM to an individual ASM, commanding it to undertake a specific task. The overall philosophy used is that commands and filtering options are task-based, with an associated region defined in the task command.

If an ASM cannot comply with a control message, it sends a SensorTaskACK acknowledgement message with an appropriate error message.
6.5.1 Task ID
Control messages must include a task ID. The task ID is an integer number generated by the HLDMM to uniquely identify a task so that the acknowledgement message and any subsequent detection messages have a reference. Task 0 indicates the default task to be undertaken when there is no other task allocated to the ASM.

6.5.2 Region-based Tasks
Region tasks are designed to control the geographical area within the coverage of steerable ASMs or field of view of fixed ASMs that the ASM should report detections from. Typically by default, the ASM will detect from its full field of view and so a region task will limit the area to report from.

In addition, if supported by the ASM, object type can be used to restrict the types of objects that will be reported.

Any task overrides the default task. If a second override task comes and the ASM can't support multiple tasks then it needs to report that in the acknowledgement message when tasked with the second task. If regions are to be removed from the default task then a default task message is sent with all remaining regions.

6.5.3 Default Commands
The command list in the initialisation message gives the list of commands that an ASM supports, including the default commands that all ASMs should support. Currently this list of "default" commands is:

- DetectionThreshold
- DetectionReportRate
- Request:Registration
- Request:Heartbeat
- Request:Reset
- Request:Stop
- Request:Start

The individual ASMs are expected to exhibit a level of autonomy. Therefore the HLDMM is not expected to use a "long screwdriver" to set low-level parameters within an individual ASM. The HLDMM may however choose to request an ASM to adjust its internal thresholds to reduce its FAR, or increase its P(d), based for example on comparing the detection performance of one ASM with another with an overlapping field of view. This is covered by the command ‘DetectionThreshold’.

The HLDMM may also choose to request an individual ASM to alter the frequency at which it reports data, for example to throttle back the detections if they were supplied too frequently. This is covered by the command ‘DetectionReportRate’ which can be changed up or down by the HLDMM, or used to request a specific output rate.

The ‘Request:Registration’ command allows the HLDMM to request the ASM to send the initialisation message again.
The ‘Request:Heartbeat’ command allows the HLDMM to request a full heartbeat message with the current value of all fields supported by that ASM.
The ‘Request:Reset’ command allows the HLDMM to stop the ASM’s current task and return to its default state.

The ‘Request:Stop’ command makes the ASM stop sending all detection messages. The ASM should continue sending heartbeat messages to maintain the connection to the data agent.

The ‘Request:Start’ command makes the ASM resume sending detection messages after a ‘Request:Stop’ command.

### 6.5.4 ClassificationThreshold Command

This is a placeholder for future functionality.

### 6.5.5 LookAt Command

The ‘lookAt’ command tasks the ASM to suspend its current tasking, take a quick look at a specific location to see if it can see anything and then return to its ongoing tasking. It will return a detection message if it detects an object at that location, or a ‘SensorTaskACK’ Acknowledgement message with the ‘Status’ field “Nothing found”. Both will include the task ID of the ‘lookAt’ command.

The ASM will determine the most appropriate field of view to use in response to the ‘lookAt’ command. If this is not the default field of view, then this will be reported by a heartbeat message to accompany any detection message response to the ‘lookAt’ command.

### 6.5.6 LookFor Command

This is a placeholder for future functionality.

### 6.5.7 Mode Command

The ‘mode’ message forces the sensor to change to one of its operating modes defined in the Initialisation message. If a sensor only supports one mode, then this command can be ignored.

### 6.6 Acknowledgement Messages

These short messages complete the handshaking, and allow an ASM to accept or reject a task. If an ASM cannot comply with a control message, it sends a SensorTaskACK acknowledgement message with an appropriate error message.

For example, the following indicates that the ASM can perform the task:

```
<SensorTaskACK><timestamp>2014-03-14 14:00:00Z</timestamp>
<sensorID>1</sensorID><TaskID>1</TaskID>
>Status>Accepted</Status></SensorTaskACK>
```

The following indicates that the ASM can perform the task but with performance degraded by external conditions e.g. rain snow:

```
<SensorTaskACK><timestamp>2014-03-14 14:00:00Z</timestamp>
<sensorID>1</sensorID><TaskID>1</TaskID>
>Status>Accepted</Status>
<Reason>Performance Degraded</Reason></SensorTaskACK>
```
The following indicates that the ASM is unable to perform the task due to its performance being severely affected by external conditions e.g. illumination:

```xml
<SensorTaskACK><timestamp>2014-03-14 14:00:00Z</timestamp><sensorID>1</sensorID><TaskID>1</TaskID><Status>Rejected</Status><Reason>Performance Degraded</Reason></SensorTaskACK>
```

The following indicates that the ASM has received a control message that it cannot interpret or contains errors:

```xml
<SensorTaskACK><timestamp>2014-03-14 14:00:00Z</timestamp><sensorID>1</sensorID><TaskID>1</TaskID><Status>Rejected</Status><Reason>Message Error</Reason></SensorTaskACK>
```

### 6.7 Error Messages

If an invalid message is received by the Data Agent from either an ASM or HLDMM client then the Data Agent will respond by sending an error message back to the source. This contains the time of the message the content of the invalid message and a message describing the error.

### 6.8 Alert Messages

If an ASM wishes to inform the HLDMM that it is about to do something, the alert message can be used to inform or alert the HLDMM of this. An example of this might be a sensor deciding to do a detailed scan of an object or area.

If the alert message merits a response from the HLDMM, the Alert Response message can be used.

For information only alert messages, with the type field ‘information’ no response is expected.

This free text field within the Alert message can also be used to pass information on the status of the ASM directly to the SAPIENT GUI. This is entirely optional. However, a potential use of this facility could be to pass information on the ASM internal behaviour (e.g. rejection of false alarms, detection of benign events) for debugging purposes and display on a simple “Engineering display window” via the SAPIENT GUI.

### 6.9 Further Guidance on XML fields

To keep the XML messages concise, it is valid to omit fields that are not mandated. This is equivalent to specifying ‘no’ in the report lists in the initialisation message.

#### 6.9.1 Text Fields

Where XML fields include text, it is assumed that the first letter will be capitalised.

#### 6.9.2 Class Types

It is up to each ASM supplier to define the list of class types that they can classify.

The following is the current core list:

- Human
- Vehicle
• Animal
• Static Object
• Unknown

Unknown means 'I am certain this is an unknown class' rather than 'I can't tell what class this is'

Vehicle can be used when the ASM cannot discriminate between different types of vehicle. Alternatively it can be used as part of a class hierarchy if multiple vehicle types can be discriminated.

Typically, ASMs will report any detection against a fixed set of classes. A target that doesn't fit well into those classes will return low confidence across all classes. This is the 'I can't tell what class this is' case. Additional class types could be added in the future to cover other cases of uncertainty as they arise.

6.9.3 Human Sub Class Types

The Human class can have the following sub class types:
• Male
• Female
• Person ID (e.g. Name or Staff Pass Number)

6.9.4 Vehicle Sub Class Types

The Vehicle class can have the following sub classes:

<table>
<thead>
<tr>
<th>Sub Class Level 1</th>
<th>Sub Class Level 2</th>
<th>Sub Class Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Wheeled</td>
<td>Heavy</td>
<td>Artic, Truck, Bus, Tracked Vehicle, Military Vehicle</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
<td>Van, People Carrier, 4X4</td>
</tr>
<tr>
<td></td>
<td>Light</td>
<td>Car, Unmanned Ground Vehicle (UGV)</td>
</tr>
<tr>
<td>2 Wheeled</td>
<td></td>
<td>Motorbike</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bicycle</td>
</tr>
</tbody>
</table>

6.9.5 Behaviour Types

Where object behaviour can be detected, this is reported independently of object type using the behaviour types. Any confidence values associated with the behaviour should be expected to relate to the object class type with the highest confidence. E.g. If an object is detected as person with a probability of 0.6 and animal with a probability of 0.3, any behaviour probability is expected to be in relation to it being a person.

The following is the current core list of behaviours. This list can be easily extended in future releases of the ICD. Not all of these behaviours will necessarily be supported on a given system. This is likely to be implementation specific.
• Walking
• Running
• Crawling
• Climbing
• Digging
• Throwing
• Loitering
• Active
• Passive

6.9.6 Reporting Classes, Sub-classes and Behaviour

It is preferred that each detection message should include a list of probabilities (confidence values) for each class, sub-class and behaviour that the ASM has said it can provide in the initialisation message. In this case the ‘confidenceDefinition’ tag in the initialisation message should be included with the value ‘Multiple Class’.

If an ASM can only provide the most likely class for each object, then the ‘confidenceDefinition’ tag in the initialisation message should be included with the value ‘Single Class’.

6.10 Managing Real-Time System Performance

Whilst the Sapient system architecture has been designed to provide sufficient data throughput for the HLDMM and end-user real-time requirements, it is prudent to have the ability to manage data bandwidth if issues do arise.

If the HLDMM decides it is receiving too much data from one or more ASMs to process in real-time, then the DetectionThreshold and DetectionReportRate commands can be used to limit the data being sent across the system.

If LSI-Supplied Data Agent detects an issue that may not be apparent to the HLDMM, it can send an Alert Message to the HLDMM to appraise it of the issue so that it can then decide on the most appropriate course of action.

It is recommended that the TCP_NO_DELAY flag is set for all socket connections to minimise the latency through the system.
7 XML Schemas

7.1 Initialisation

```xml
< SensorRegistration >
  < timestamp > yyyy-MM-ddTHH:mm:ss.fffZ </ timestamp > <!-- UTC (Zulu) time message sent -->
  < sensorID > <!-- Included if the ASM already has a sensorID associated with it. If omitted, the SAPIENT system will assign a sensorID for the ASM. -->
  </ sensorID >
  < sensorType > Acme EO ASM </ sensorType > <!-- Unique Type identifier for this ASM -->
  < heartbeatDefinition >
    < heartbeatInterval units = "seconds" value = "10" /> <!-- time between heartbeats, 0 means on request -->
    < locationType units = "decimal degrees-metres" datum = "WGS84" zone = "30U" north = "Grid" 
      > UTM, GPS, RangeBearing </ locationType >
    < locationType units = "decimal degrees-metres" datum = "WGS84" zone = "30U" north = "Grid" 
      > UTM, GPS, RangeBearing </ locationType >
    < locationType units = "decimal degrees-metres" datum = "WGS84" zone = "30U" north = "Grid" 
      > UTM, GPS, RangeBearing </ locationType >
    < locationType units = "decimal degrees-metres" datum = "WGS84" zone = "30U" north = "Grid" 
      > UTM, GPS, RangeBearing </ locationType >
    < locationType units = "decimal degrees-metres" datum = "WGS84" zone = "30U" north = "Grid" 
      > UTM, GPS, RangeBearing </ locationType >
    < locationType units = "decimal degrees-metres" datum = "WGS84" zone = "30U" north = "Grid" 
      > UTM, GPS, RangeBearing </ locationType >
  </ heartbeatDefinition >
  < heartbeatReport category = "sensor" type = "sensorLocation" units = "" onChange = "true" />
  < heartbeatReport category = "sensor" type = "fieldOfView" units = "locationList, rangeBearingCone" 
    onChange = "true" />
  < heartbeatReport category = "sensor" type = "coverage" units = "locationList, rangeBearingCone" 
    onChange = "true" />
  < heartbeatReport category = "sensor" type = "obscurations" units = "locationList, rangeBearingCone" 
    onChange = "true" />
  < heartbeatReport category = "power" type = "status" units = "OK, Fault" onChange = "true" />
  < heartbeatReport category = "power" type = "level" units = "percentage" onChange = "true" />
  < heartbeatReport category = "mode" type = "" units = "Default, Others as defined in registration message" 
    onChange = "true" />
</ SensorRegistration >
```

<!-- list of all reported parameters - this list can be extended -->

<!-- this specifies that source and status attributes of the power field will be reported -->

<!-- if power level is specified then status is also required -->

<!-- the SAPIENT system will assign a sensorID for the ASM -->

<!-- Unique Type identifier for this ASM -->

<!-- time between heartbeats, 0 means on request -->

<!-- UTC (Zulu) time message sent -->

<!-- Included if the ASM already has a sensorID associated with it. If omitted, the SAPIENT system will assign a sensorID for the ASM. -->

<!-- Unique Type identifier for this ASM -->
<heartbeatReport category="status" type="InternalFault" units="" onChange="true" />
<heartbeatReport category="status" type="ExternalFault" units="" onChange="true" />
<heartbeatReport category="status" type="Illumination" units="Bright, Dark, Normal" onChange="true" />
<heartbeatReport category="status" type="Weather" units="text" onChange="true" />
<heartbeatReport category="status" type="Clutter" units="Low, Medium, High" onChange="true" />
<heartbeatReport category="status" type="MotionSensitivity" units="probability" onChange="true" />
<heartbeatReport category="status" type="PTZStatus" units="Moving, Stopped" onChange="true" />
<heartbeatReport category="status" type="PD" units="probability" onChange="true" />

<modeDefinition type = "Permanent">
  <!-- mandatory name to associate with this mode -->
  <modeName>Default</modeName>
  <!-- optional description to associate with this mode -->
  <modeDescription>Free form text</modeDescription>
  <!-- time to settle to normal performance -->
  <settleTime units="seconds" value="10"/>
  <!-- maximum detection latency -->
  <maximumLatency units="seconds" value="5"/>

  <!-- Sensor Field of View behaviours - If omitted will assume fixed -->
  <scanType>Fixed/Scanning/Steerable</scanType>

  <!-- ASM tracking capabilities - If omitted will assume none -->
  <trackingType>None/Tracklet/Track/TrackWithReID</trackingType>

  <!-- Include this if permission is required to enter this mode by the ASM sending an alert message and awaiting an alert response message before proceeding -->
  <modeParameter type="PermissionToChange" value="Required"/>

  <!-- This section describes the format of Detection Messages -->
  <detectionDefinition>
    <!-- location units to be used by detection messages -->
  </detectionDefinition>
</modeDefinition>
<locationType units="decimal degrees-metres" datum="WGS84" zone="30U" north="Grid"
>UTM, GPS, RangeBearing</locationType>

<!-- list of location error characterisations -->
<geometricError type = "Standard Deviation" units="metres" variationType="Linear with Range,
Squared with Range">
  <performanceValue type = "eRmin" value="0.1" />
  <performanceValue type = "eRmax" value="0.5" />
</geometricError>

<!-- list of all reported detection parameters - this list can be extended -->
<detectionReport category="detection" type="confidence" units="probability" />
<detectionReport category="track" type="confidence" units="probability" />
<detectionReport category="track" type="speed" units="m-s" />
<detectionReport category="track" type="az" units="degrees" />
<detectionReport category="track" type="dR" units="metres" />
<!-- Object Change in Range in metres up to two decimal places -->
<detectionReport category="track" type="dAz" units="degrees" />
<!-- Object change in azimuth in degrees up to two decimal places -->
<detectionReport category="track" type="predictedLocation" units="geo" />
<detectionReport category="track" type="predictionTimestamp" units="UTC" />
<detectionReport category="object" type="dopplerSpeed" units="m-s" />
<detectionReport category="object" type="dopplerAz" units="degrees" />
<detectionReport category="object" type="majorLength" units="metres" />
<detectionReport category="object" type="majorAxisAz" units="degrees" />
<detectionReport category="object" type="minorLength" units="metres" />
<detectionReport category="object" type="height" units="metres" />
<detectionReport category="object" type="colour" units="none" />
<detectionReport category="object" type="state" units="none" onChange="true" />

<!-- To keep this simple, this will be file path to a shared storage folder or a URL to a server location -->
<detectionReport category="associatedFile" type="image" units="none" />
<detectionReport category="associatedFile" type="pointCloud" units="none" />

<!-- list of Performance measures - only provide if available -->
<detectionPerformance type = "PD" units="Per Frame" unitValue="1" variationType="None">
  <performanceValue type = "scalar" value="0.9" />
</detectionPerformance>

<detectionPerformance type = "FAR" units="Per Period" unitValue="1" variationType="Linear with Range">
  <performanceValue type = "eRmin" value="0.1" />
  <performanceValue type = "eRmax" value="0.5" />
</detectionPerformance>

<detectionPerformance type = "FAR" units="Per Period" unitValue="1" variationType="Inverse Square with Range">
  <performanceValue type = "eRmin" value="0.1" />
</detectionPerformance>

<detectionClassDefinition>
  <confidenceDefinition>Single Class, Multiple Class</confidenceDefinition>
  <!-- omit if not providing confidence -->
  <!-- Single Class - only provide confidence for the most likely class -->
  <!-- Multiple Class - provide confidence for all classes -->
  <confusionMatrix type="" />
  <!-- omit if not providing -->
  <classPerformance type = "FAR" units="Per Period" unitValue="1" variationType="Linear with Range">
    <performanceValue type = "eRmin" value="0.1" />
  </classPerformance>
</detectionClassDefinition>
<performanceValue type="eRmax" value="0.5"/>
</classPerformance>

<!-- list all the object classification types the ASM can detect -->
<classDefinition type="Human">
  <!-- omit if not providing confidence -->
  <confidence units="probability"/>
</classDefinition>

<!-- list of sub-classes that can be detected for each class - can be omitted if there are none -->
<subClassDefinition level="1" type="PersonID" values="names">
  <confidence units="probability"/>
</subClassDefinition>

<!-- list of possible behaviours that can be detected and reported -->
<behaviourDefinition type="Walking">
  <confidence units="probability"/>
</behaviourDefinition>

<behaviourDefinition type="Running">
  <confidence units="probability"/>
</behaviourDefinition>

<behaviourDefinition type="Crawling">
  <confidence units="probability"/>
</behaviourDefinition>

<behaviourDefinition type="Climbing">
  <confidence units="probability"/>
</behaviourDefinition>
<behaviourDefinition type = "Throwing">
    <confidence units="probability" /> <!-- omit if not providing confidence -->
</behaviourDefinition>

<regionType>Area of Interest</regionType>
<regionType>Ignore</regionType>
<regionType>Boundary</regionType>

<settleTime units="seconds" value="1"/> <!-- time to settle to normal performance -->

<regionType>Area of Interest defines an area to look for detections -->
<regionType>Ignore defines an area to ignore detections -->
<regionType>Boundary defines a transition area between two locations where only transitions should be reported -->
<locationType units="decimal degrees-metres" datum="WGS84" zone="30U" north="Grid" >UTM, GPS, RangeBearing</locationType>

<classFilterDefinition type = "All">
    <!-- list of class parameters the ASM can filter on -->
    <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<classFilterDefinition type = "Human">
    <!-- list of class parameters the ASM can filter on -->
    <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<classFilterDefinition type = "Vehicle">
    <!-- list of class parameters the ASM can filter on -->
    <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<subClassFilterDefinition level = "1" type="PersonID">
    <!-- list of sub-class parameters the ASM can filter on -->
    <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</subClassFilterDefinition>

<subClassFilterDefinition level = "1" type="Vehicle Class">
    <!-- list of sub-class parameters the ASM can filter on -->
</subClassFilterDefinition>
<filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />

<!-- list of sub-classes that can be filtered on for each sub-class - can be omitted if there are none -->
<subClassFilterDefinition level="2" type="Size">
  <!-- list of sub-class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</subClassFilterDefinition>

<!-- list of sub-classes that can be filtered on for each sub-class - can be omitted if there are none -->
<subClassFilterDefinition level="3" type="Vehicle Type">
  <!-- list of sub-class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</subClassFilterDefinition>

</classFilterDefinition>

<classFilterDefinition type="Animal">
  <!-- list of class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<classFilterDefinition type="Unknown">
  <!-- list of class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<!-- list all the behaviour types the ASM can filter on - can be omitted if no filtering available -->
<behaviourFilterDefinition type="Walking">
  <!-- list of behaviour parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</behaviourFilterDefinition>

<behaviourFilterDefinition type="Running">
  <!-- list of behaviour parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</behaviourFilterDefinition>

<behaviourFilterDefinition type="Crawling">
  <!-- list of behaviour parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</behaviourFilterDefinition>

</regionDefinition>

<!-- list all types of command the ASM can support -->
<command name="Request" units="Registration, Reset, Heartbeat, Sensor Time, Stop, Start, Take Snapshot" completionTime="10" completionTimeUnits="seconds"/>
<command name="DetectionThreshold" units="Auto, Low, Medium, High" completionTime="1" completionTimeUnits="seconds"/>
<command name="DetectionReportRate" units="Auto, Low, Medium, High" completionTime="1" completionTimeUnits="seconds"/>
<command name="ClassificationThreshold" units="Auto, Low, Medium, High" completionTime="1" completionTimeUnits="seconds"/>
<command name="Mode" units="Default, Others as defined in registration message" completionTime="2" completionTimeUnits="seconds"/>
<!-- Normally, the ASM will work in 'Default' mode but it can be forced into another mode defined in the modeDefinition section of the registration message -->
<command name="LookAt" units="locationList, rangeBearingCone" completionTime="30" completionTimeUnits="seconds"/>
<command name="LookFor" units="image" completionTime="60" completionTimeUnits="seconds"/>
</taskDefinition>
</modeDefinition>
<modeDefinition type = "Temporary">
<modeName>3D Scan</modeName> <!-- mandatory name to associate with this mode -->
<modeDescription>Free form text</modeDescription> <!-- optional description to associate with this mode -->
<settleTime units="seconds" value="2"/> <!-- time to settle to normal performance -->
<maximumLatency units="seconds" value="5"/> <!-- maximum detection latency -->
<scanType>Scanning</scanType>
<trackingType>None</trackingType>
<duration units="seconds" value="60"/> <!-- can be omitted if mode type is Permanent -->

<!-- This section describes the format of Detection Messages -->
<detectionDefinition>
<!-- location units to be used by detection messages -->
<locationType units="decimal degrees-metres" datum="WGS84" zone="30U" north="Grid">UTM, GPS, RangeBearing</locationType>

<!-- location error characterisation -->
<geometricError type = "Standard Deviation" units="metres" variationType="Linear with Range">
<performanceValue type = "eRMin" value="0.1"/>
<performanceValue type = "eRMax" value="0.5"/>
</geometricError>

<!-- list of all reported detection parameters - this list can be extended -->
<detectionReport category="detection" type="confidence" units="probability"/>
<detectionReport category="track" type="confidence" units="probability"/>
<detectionReport category="track" type="speed" units="m-s"/>
<detectionReport category="track" type="az" units="degrees"/>
<detectionReport category="track" type="dR" units="metres"/> <!-- Object Change in Range in metres up to two decimal places -->
<detectionReport category="track" type="dAz" units="degrees"/>  <!-- Object change in azimuth in degrees up to two decimal places -->
<detectionReport category="track" type="predictedLocation" units="geo"/>
<detectionReport category="object" type="dopplerSpeed" units="m-s"/>
<detectionReport category="object" type="dopplerAz" units="degrees"/>
<detectionReport category="object" type="majorLength" units="metres"/>
<detectionReport category="object" type="majorAxisAz" units="degrees"/>
<detectionReport category="object" type="minorLength" units="metres"/>
<detectionReport category="object" type="height" units="metres"/>
<detectionReport category="object" type="colour" units="none" />
<detectionReport category="object" type="state" units="none" onChange="true" />

<!-- To keep this simple, this will be file path to a shared storage folder or a URL to a server location -->
<detectionReport category="object" type="state" units="none" />
<detectionPerformance type = "PD" units = "Per Frame" unitValue="1" variationType="None">
  <performanceValue type = "scalar" value="0.9"/>
</detectionPerformance>

<detectionPerformance type = "FAR" units = "Per Period" unitValue="1" variationType="Linear with Range">
  <performanceValue type = "eRmin" value="0.1"/>
  <performanceValue type = "eRmax" value="0.5"/>
</detectionPerformance>

<detectionPerformance type = "FAR" units = "Per Period" unitValue="1" variationType="Inverse Square with Range">
  <performanceValue type = "eRmin" value="0.1"/>
</detectionPerformance>

<detectionClassDefinition>
  <confidenceDefinition type="Single Class, Multiple Class"/>
  <classDefinition type="Human">
    <confidence units="probability"/>
    <subClassDefinition level = "1" type="PersonID" values = "names">
      <confidence units="probability"/>
    </subClassDefinition>
  </classDefinition>
  <classDefinition type="Unknown">
    <confidence units="probability"/>
  </classDefinition>
</detectionClassDefinition>

<behaviourDefinition type="Walking">
  <confidence units="probability"/>
</behaviourDefinition>
<detectionDefinition/>

<!-- This section describes the format of Control Messages -->

<taskDefinition>
<concurrentTasks>1</concurrentTasks>
<!-- the number of different tasks that can run at a time in this mode -->

#regionDefinition>

<!-- list all types of region the ASM can support - can be omitted if there are none -->
<regionType>Area of Interest</regionType>
<regionType>Ignore</regionType>
<regionType>Boundary</regionType>

<!-- Area of Interest defines an area to look for detections -->
<!-- Ignore defines an area to ignore detections -->
<!-- Boundary defines a transition area between two locations where only transitions should be reported -->
<locationType units="decimal degrees-metres" datum="WGS84" zone="30U" north="Grid" >UTM, GPS, RangeBearing</locationType>

<!-- list all the classification types the ASM can filter on - can be omitted if no filtering available -->
<classFilterDefinition type = "All">
  <!-- list of class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<classFilterDefinition type = "Human">
  <!-- list of class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<!-- list all the sub-classes that can be filtered on for each class - can be omitted if there are none -->
<subClassFilterDefinition level = "1" type="PersonID">
  <!-- list of sub-class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</subClassFilterDefinition>

<classFilterDefinition type = "Unknown">
  <!-- list of class parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</classFilterDefinition>

<behaviourFilterDefinition type = "Walking">
  <!-- list of behaviour parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</behaviourFilterDefinition>

<!-- list all the behaviour types the ASM can filter on - can be omitted if no filtering available -->
<behaviourFilterDefinition type = "Walking">
  <!-- list of behaviour parameters the ASM can filter on -->
  <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None" />
</behaviourFilterDefinition>
<behaviorFilterDefinition/>
</regionDefinition>

<!-- list all types of command the ASM can support -->
<command name="Request" units="Registration, Reset, Heartbeat, Sensor Time, Stop, Start, Take Snapshot" completionTime="10" completionTimeUnits="seconds" />
<command name="DetectionThreshold" units="Auto, Low, Medium, High" completionTime="1" completionTimeUnits="seconds" />
<command name="DetectionReportRate" units="Auto, Low, Medium, High" completionTime="1" completionTimeUnits="seconds" />
<command name="ClassificationThreshold" units="Auto, Low, Medium, High" completionTime="1" completionTimeUnits="seconds" />
<command name="Mode" units="Default, Others as defined in registration message" completionTime="2" completionTimeUnits="seconds" />
<!-- Normally, the ASM will work in 'Default' mode but it can be forced into another mode defined in the modeDefinition section of the registration message -->
<command name="LookAt" units="locationList, rangeBearingCone" completionTime="30" completionTimeUnits="seconds" />
<command name="LookFor" units="image" completionTime="60" completionTimeUnits="seconds" />
</taskDefinition>
</modeDefinition>
</SensorRegistration>

7.2 Heartbeat Messages

<?xml version="1.0"?>
<StatusReport> <!-- These are the heartbeat messages -->
<timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC time of message in ISO 8601 format -->
<sourceID></sourceID> <!-- unique numerical identifier of the ASM or other source sending the report -->
<reportID></reportID> <!-- numerical identifier of this report, incremented by the source -->
<system>OK/Error/Tamper/GoodBye</system> <!-- overall status of sender, ASM sends good bye to close the communication cleanly in case it needs to shut down and reboot. -->
<info>Unchanged/New/Additional</info> <!-- whether the heartbeat message is indicating an unchanged status, new information or additional information to previous messages. -->
<activeTaskID>0,1</activeTaskID> <!-- current tasks being performed by the ASM -->
<power source = "Mains, Battery" status = "OK, Fault" level="50" />
<mode>Transition, Stopped, Failed, Default, Others as defined in registration message</mode> <!-- Mode is an optional field reported where a sensor provides multiple modes or states -->
<!-- If a fatal error has occurred then report "Failed" under sensor mode -->
<location of sensor - if known and fixed only needs reporting once at initialisation -->
<sensorLocation>
<location>
<X>-2.5832462</X> <!-- longitude -->
<Y>51.4497158</Y> <!-- latitude -->
<Z>76.0</Z> <!-- altitude -->
<eX>0.0000035</eX> <!-- GPS error -->
<eY>0.0000045</eY> <!-- GPS error -->
<eZ>5.0</eZ> <!-- Altitude error -->
</location>
</sensorLocation>

<!-- current field of view of sensor -->
<fieldOfView>
<!-- provide location list or range bearing -->

<!-- Points should be in order around the boundary of the field of view -->
<locationList>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
</locationList>

<!-- more points . . . -->

<!-- alternatively describe field of view as a cone -->
<rangeBearingCone>
  <R>20.0</R> <!-- Detection range of sensor in metres up to 2 decimal places -->
  <Az></Az> <!-- Angle of detection centre axis relative to North - based on system wide concept of north. -->
  <Ele></Ele> <!-- Angle of detection centre axis above/below horizontal in degrees up to 2 decimal places - if omitted then horizontal will be assumed. -->
  <hExtent></hExtent> <!-- Horizontal field of view in degrees up to 2 decimal places -->
  <vExtent></vExtent> <!-- Vertical field of view in degrees up to 2 decimal places -->
  <minR></minR> <!-- Minimum range of sensor in metres up to 2 decimal places -->
    <eR>0.5</eR> <!-- error in maximum range -->
    <eAz>1.5</eAz> <!-- compass error -->
    <eEle>1.5</eEle> <!-- elevation angle error -->
    <ehExtent>0.5</ehExtent> <!-- error in horizontal field of view -->
    <evExtent>0.5</evExtent> <!-- error in vertical field of view -->
    <eminR>0.5</eminR> <!-- error in minimum range -->
</rangeBearingCone>

<coverage>
<!-- full area that sensor can cover by moving or changing settings -->
<!-- provide location list or range bearing -->

<!-- Points should be in order around the boundary of the field of view -->
<locationList>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
</locationList>

<!-- more points . . . -->

<!-- alternatively describe field of view as a cone -->
<rangeBearingCone>
  <R></R> <Az></Az> <Ele></Ele> <hExtent></hExtent> <vExtent></vExtent> <minR></minR>
    <eR></eR> <eAz></eAz> <eEle></eEle> <ehExtent></ehExtent>
  <evExtent></evExtent> <eminR></eminR>
</rangeBearingCone>

<coverage>

<!-- list of obscuration regions -->
<!-- provide location list or range bearing -->

<!-- Points should be in order around the boundary of the obscuration -->
<locationList>
  <location> <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ> </location>
</locationList>
Detection Messages

<?xml version="1.0"?>
<timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC time of detection in ISO 8601 format -->
<sourceID></sourceID> <!-- unique numerical identifier of the ASM or other source sending the report -->
<reportID></reportID> <!-- numerical identifier of this report, incremented by the source -->
<objectID></objectID> <!-- unique numerical identifier of this object generated by ASM -->
<taskID></taskID> <!-- task that this detection is generated from -->
<state>lostInView,lostFromView</state> <!-- optional field to allow reporting loss of track within field of view -->

<!-- provide location or range bearing -->
<location>

</location>

</rangeBearing>
<rangeBearing>

</rangeBearing>
</predictedLocation>

</predictedLocation>

<objectInfo type="dopplerSpeed" value="2.0" e="0.1"/>
<objectInfo type="dopplerAzimuth" value="270.0" e="0.5"/>
<objectInfo type="majorLength" value="1.2" e="0.1"/>
<objectInfo type="majorAxisAzimuth" value="270.0" e="5.0"/>
<objectInfo type="minorLength" value="1.4" e="0.1"/>
<objectInfo type="height" value = "1.8" e = "0.1" />
<colour>red,RGB</colour>

<!--[if providing multiple classes and sub-classes, this list should have the same size and types as given in the registration message -->
<class type="Human">
  <confidence>0.7</confidence>
  <subClass level = "1" type="PersonID" value = "Joe Bloggs">
    <confidence>0.9</confidence>
  </subClass>
</class>
<class type="Vehicle">
  <confidence>0.2</confidence>
  <subClass level = "1" type="Vehicle Class" value = "4 Wheeled, 2 Wheeled">
    <confidence>0.9</confidence>
    <subClass level = "2" type="Size" value = "Heavy, Medium, Light">
      <confidence>0.5</confidence>
      <subClass level = "3" type="Vehicle Type" value = "Truck, Bus, Van, People Carrier, 4X4, Car, Motorbike, Bicycle">
        <confidence>0.5</confidence>
      </subClass>
    </subClass>
  </subClass>
</class>
<class type="Animal">
  <confidence>0.01</confidence>
</class>
<class type="Unknown">
  <confidence>0.01</confidence>
</class>
<behaviour type="Walking">
  <confidence>0.5</confidence>
</behaviour>
<behaviour type="Running">
  <confidence>0.4</confidence>
</behaviour>
<behaviour type="Crawling">
  <confidence>0.1</confidence>
</behaviour>
<associatedFile type="image, pointCloud" url="" />
<!--[if To keep this simple, this will be file path to a shared storage folder or a URL to a server location -->
</DetectionReport>

### 7.4 Control Messages

<?xml version="1.0"?>
<SensorTask>
<timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC (Zulu) time message sent in ISO 8601 format-->
<sensorID></sensorID> <!-- unique numerical identifier of the ASM to task -->
<taskID></taskID> <!-- generated by the HLDMM, 0 for default task -->
<taskName>Default, Look At Gate</taskName> <!-- optional name to associate with this task ID -->
<taskDescription>
  <taskDescription> <!-- optional description to associate with this task -->
  </taskDescription>
</taskDescription>
<taskStartTime>yyyy-MM-ddTHH:mm:ssZ</taskStartTime> <!-- UTC (Zulu) time to start this task - if omitted, do this immediately -->
<taskEndTime>yyyy-MM-ddTHH:mm:ssZ</taskEndTime> <!-- UTC (Zulu) time to stop this task - if omitted, do this until told to stop or do something else -->
<control>Start/Stop/Pause/Default</control>
<!-- What to do with this task. -->
<!-- Start - Initialise and schedules or starts the task or restarts a paused task -->
<!-- Stop - stop the task, remove its definition, revert to previous task -->
<!-- Pause - stop the task but keep it defined for later restart, revert to previous task -->
<!-- Default - start or revert to the default task and remove all other task definitions -->

<!-- Define one or more regions for this task -->
<region>
  <!-- Area of Interest, Ignore, Boundary
  Area of Interest defines an area to look for detections --
  Ignore defines an area to ignore detections -->
  Boundary defines a transition area between two locations where only transitions should be reported -->
</region>

<regionID><regionID><regionName><regionName>

<!-- Points should be in order around the boundary of the region -->
<locationList>
  <!-- Error values are not required here but are supplied for consistency with other messages -->
  <location> <X/> <Y/> <Z/> <eX/> <eY/> <eZ/> </location>
</locationList>

<!-- alternatively describe region as a cone -->
<rangeBearingCone>
  <!-- Error values are not required here but are supplied for consistency with other messages -->
  <R/> <Az/> <Ele/> <hExtent/> <vExtent/> <vEvExtent/> <evExtent/>
</rangeBearingCone>

<!-- list of class Filters to apply to region -->
<!-- if omitted then no filtering is applied -->
<classFilter type="All" priority="Low">
  <parameter name="confidence" operator="Greater Than" value="0.9"/>
</classFilter>

<classFilter type="Human" priority="High">
  <parameter name="confidence" operator="Greater Than" value="0.3"/>
</classFilter>

<classFilter type="Vehicle" priority="Medium">
  <parameter name="confidence" operator="Greater Than" value="0.4"/>
  <subClassFilter level = "1" type="PersonID" value="Joe Bloggs" priority="Medium">
    <parameter name="confidence" operator="Greater Than" value="0.2"/>
  </subClassFilter>
  <parameter name="confidence" operator="Greater Than" value="0.2"/>
  <subClassFilter level = "2" type="Size" value = "Medium" priority="Medium">
    <parameter name="confidence" operator="Greater Than" value="0.3"/>
  </subClassFilter>
  <subClassFilter level = "3" type="Vehicle Type" value = "Car">
    <parameter name="confidence" operator="Greater Than" value="0.3"/>
  </subClassFilter>
</classFilter>
<classFilter type="Animal" priority="Low">
<parameter name="confidence" operator="Greater Than" value="0.1" />
</classFilter>

<behaviourFilter type="Walking" priority="High">
<parameter name="confidence" operator="Greater Than" value="0.5" />
</behaviourFilter>

<region>

<command>
:request>Registration, Reset, Heartbeat, Sensor Time, Stop, Start, Take Snapshot</request>
<detetectionThreshold>Auto, Low, Medium, High, Lower, Higher</detetectionThreshold>
<detetectionReportRate>Auto, Low, Medium, High, Lower, Higher</detetectionReportRate>
<classificationThreshold>Auto, Low, Medium, High, Lower, Higher</classificationThreshold>

<me>Default, Others as defined in registration message</me>
</command>

<!-- change the operating mode of the ASM. -->

<!-- Normally, the ASM will work in 'Default' mode but it can be forced into another mode defined in the registration message -->

<!-- Not required where a sensor only operates in one mode -->

<!-- direct sensor to look at a specific location then either send a detectionReport or "nothing found" feedback message -->

<!-- provide location list or range bearing -->

<!-- Points should be in order around the boundary of the field of view -->

<!-- If a single point is provided, the sensor will focus on that location -->

<!-- Error values are not required here but are supplied for consistency with other messages -->

<!-- alternatively describe field of view as a cone -->

<!-- Error values are not required here but are supplied for consistency with other messages -->

<!-- look for type="image" url="image_snapshot.jpg" -->

<lookFor type="image" url="image_snapshot.jpg" />
</command>
</SensorTask>

7.5 Acknowledgement Messages

7.5.1 Registration Acknowledgement

<?xml version="1.0"?>
<SensorRegistrationACK>
<sensorID>1</sensorID> <!-- The SAPIENT system will assign/confirm the sensorID for each ASM. -->
</SensorRegistrationACK>
7.5.2 Control Acknowledgement

```xml
<?xml version="1.0"?>
<SensorTaskACK>
  <timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC (Zulu) time message sent -->
  <sensorID></sensorID>
  <taskID></taskID>
  <status>Accepted, Rejected, Complete</status>
  <reason>Outside Field of View, Location Obscured, Conflicts with task N, Not Supported, Performance Degraded, Message Error, Task Error, System Error, Out Of Range, Nothing Found, No ASM with this ID</reason>
  <associatedFile type="image" url="filenameYYMMDD_HHMSS.jpg" />
</SensorTaskACK>
```

7.5.3 Null Response to ‘LookAt’ command

```xml
<?xml version="1.0"?>
<SensorTaskACK>
  <timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC (Zulu) time message sent -->
  <sensorID></sensorID>
  <taskID></taskID>
  <status>Complete</status>
  <reason>Nothing Found</reason>
</SensorTaskACK>
```

7.6 Error Messages

```xml
<?xml version="1.0"?>
<Error>
  <timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC (Zulu) time message sent -->
  <packet>the message that caused the error</packet>
  <errorMessage>clean error message giving some details as to issues</errorMessage>
</Error>
```

7.7 Alert Messages

```xml
<?xml version="1.0"?>
<Alert>
  <timestamp>yyyy-MM-ddTHH:mm:ss.fffZ</timestamp> <!-- UTC (Zulu) time message sent -->
  <sourceID></sourceID>
  <alertID></alertID>
  <alertType>Mode Change, Information</alertType> <!-- optional field to specify a type -->
  <status>Active, Acknowledge, Reject, Ignore, Clear</status> <!-- optional field to specify initial status -->
  <description>Text description of alert</description> <!-- by default Active will be assumed -->
  <location>
    <X></X> <Y></Y> <Z></Z> <eX></eX> <eY></eY> <eZ></eZ>
  </location>
</Alert>
```

If available may be calculated by summing the relative value to the sensor and the known sensor altitude so that terrain data is not required. -->

1- eX is x error of location in metres up to two decimal places -->
1- eY is y error of location in metres up to two decimal places -->
1- eZ is z error of location in metres up to two decimal places -->
7.8 Alert Response Messages

```xml
<?xml version="1.0"?>
<AlertResponse>
  <timestamp>yyyy-MM-ddTHH:mm:ssZ</timestamp> <!-- UTC (Zulu) time message sent -->
  <sourceID/></sourceID>
  <alertID/></alertID>
  <status>Acknowledge, Reject, Ignore, Clear</status>
  <reason></reason>
</AlertResponse>
```
8 Example Messages

In order to help clarify the use of the xml schema, example messages are provided for a fictional generic EO sensor, and its interaction with the HLDMM. This sensor is a fixed field of view camera, which has the ability to detect and track objects, classify them as human or vehicle. Further, it can classify humans according to a number of activities being undertaken, and further discriminate vehicles into a number of vehicle types.

Examples are given of
- the sensor registration message,
- heartbeat messages, including as the situation changes,
- a detection message
- control and acknowledgement messages, showing a dialogue between the ASM and HLDMM

8.1 Example Registration Message

```xml
<?xml version="1.0"?>
<SensorRegistration>
  <timestamp>2014-02-26T11:31:25.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
  <sensorID></sensorID>  <!-- Included if the ASM already has a sensorID associated with it. If omitted, the Sapient system will assign a sensorID for the ASM. -->
  <sensorType>ACME Generic EO Sensor</sensorType>  <!-- Unique Type identifier -->

  <!-- This section describes the format of Heartbeat Messages -->
  <heartbeatDefinition>
    <heartbeatInterval units="seconds" value="10"/> <!-- time between heartbeats -->
    <fieldOfViewDefinition>
      <locationType units="decimal degrees-metres" datum="WGS84" zone="30U" north="Grid">
        GPS</locationType>
      </fieldOfViewDefinition>
      <obscurationDefinition>
        <locationType units="decimal degrees-metres" datum="WGS84" zone="30U" north="Grid">
          GPS</locationType>
      </obscurationDefinition>

    <!-- list of all reported parameters - this list can be extended -->
    <heartbeatReport category="sensor" type="sensorLocation" units="" onChange="true"/>
    <heartbeatReport category="sensor" type="fieldOfView" units="locationList" onChange="true"/>
    <heartbeatReport category="sensor" type="obstructions" units="locationList" onChange="true"/>
    <heartbeatReport category="power" type="status" units="" onChange="true"/>
    <heartbeatReport category="status" type="InternalFault" units="" onChange="true"/>
    <heartbeatReport category="status" type="ExternalFault" units="" onChange="true"/>
    <heartbeatReport category="status" type="Illumination" units="Bright, Dark, Normal" onChange="true"/>
  </heartbeatDefinition>

  <modeDefinition type = "Permanent">
    <modeName>Default</modeName>  <!-- mandatory name to associate with this mode -->
    <modeDescription>Normal Operation</modeDescription>  <!-- optional description to associate with this mode -->
    <settleTime units="seconds" value="10"/>
    <maximumLatency units="seconds" value="5"/> <!-- maximum detection latency -->
</SensorRegistration>
```
<scanType>Fixed</scanType>

<trackingType>Tracklet</trackingType>

<!-- This section describes the format of Detection Messages -->
<definition>

<locationType units="decimal degrees-metres" datum="WGS84"
north="Grid">GPS</locationType>

<!-- list of location error characterisations -->
<geometricError type="Standard Deviation" units="metres" variationType="Linear with Range">
  <performanceValue type="eRmin" value="0.1" />
  <performanceValue type="eRmax" value="0.5" />
</geometricError>

<!-- list of all reported detection parameters - this list can be extended -->
<report category="detection" type="confidence" units="probability" />
<report category="track" type="confidence" units="probability" />
<report category="track" type="speed" units="m-s" />
<report category="track" type="az" units="degrees" />
<report category="object" type="majorLength" units="metres" />
<report category="object" type="majorAxisAz" units="degrees" />
<report category="object" type="minorLength" units="metres" />
<report category="object" type="height" units="metres" />
<report category="object" type="colour" units="none" />
<report category="associatedFile" type="image" units="none" />
<definition type="Multiple Class">
<confidence type=" FAR" units="Per Period" unitValue="1" variationType="Linear with Range">
  <performanceValue type="eRmin" value="0.1" />
  <performanceValue type="eRmax" value="0.5" />
</confidence>

<!-- list all the object classification types the ASM can detect -->
<class type="Human"><confidence units="probability" />
</class>
<class type="Vehicle">
  <subClassification level="1" type="Vehicle Class" values="4 Wheeled">
    <confidence units="probability" />
    <subClassification level="2" type="Size" values="Light">
      <confidence units="probability" />
      <subClassification level="2" type="Vehicle Type" values="Car">
        <confidence units="probability" />
    </subClassification>
  </subClassification>
</class>
</definition>

<behaviour type="Walking"><confidence units="probability" />
</behaviour>

<behaviour type="Running"><confidence units="probability" />
</behaviour>

<behaviour type="Crawling"><confidence units="probability" />
</behaviour>
<detectionDefinition>
  <!-- This section describes the format of Control Messages -->
  <taskDefinition>
    <concurrentTasks>1</concurrentTasks>
    <regionDefinition>
      <!-- list all types of region the ASM can support - can be omitted if there are none -->
      <regionType>Area of Interest</regionType>
      <regionType>Ignore</regionType>
      <settleTime units="seconds" value="1"/> <!-- time to settle to normal performance -->
      <locationType units="decimal degrees" datum="WGS84" north="Grid"/>
    </regionDefinition>
    <classFilterDefinition type = "Human">
      <!-- list of class parameters the ASM can filter on -->
      <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </classFilterDefinition>
    <classFilterDefinition type = "Vehicle">
      <!-- list of class parameters the ASM can filter on -->
      <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </classFilterDefinition>
    <subClassFilterDefinition level = "1" type="Vehicle Class">
      <!-- list of sub-class parameters the ASM can filter on -->
      <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </subClassFilterDefinition>
    <subClassFilterDefinition level = "2" type="Size">
      <!-- list of sub-class parameters the ASM can filter on -->
      <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </subClassFilterDefinition>
    <subClassFilterDefinition level = "3" type="Vehicle Type">
      <!-- list of sub-class parameters the ASM can filter on -->
      <filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </subClassFilterDefinition>
    <behaviourFilterDefinition type = "Walking">
      <!-- filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </behaviourFilterDefinition>
    <behaviourFilterDefinition type = "Running">
      <!-- filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </behaviourFilterDefinition>
    <behaviourFilterDefinition type = "Crawling">
      <!-- filterParameter name="confidence" operators="All, Greater Than, Less Than, Equals, None"/>
    </behaviourFilterDefinition>
  </taskDefinition>
</detectionDefinition>
8.2 Example Heartbeat Messages

8.2.1 Initial heartbeat message

```xml
<StatusReport>
    <timestamp>2014-02-27T13:06:43.100Z</timestamp>
    <sourceID>5</sourceID>
    <reportID>67</reportID>
    <system>OK</system>
    <info>New</info>
    <activeTaskID>0</activeTaskID>
    <power source = "Mains" status = "OK" />
    <fieldOfView>
        <locationList>
            <location><X>123450</X><Y>345670</Y><eX>2</eX><eY>3</eY></location>
            <location><X>123480</X><Y>345690</Y><eX>2</eX><eY>3</eY></location>
            <location><X>123450</X><Y>345670</Y><eX>2</eX><eY>3</eY></location>
            <location><X>123450</X><Y>345670</Y><eX>2</eX><eY>3</eY></location>
        </locationList>
    </fieldOfView>
    <obscuration>
        <locationList><location><X>123475</X><Y>345680</Y><eX>2</eX><eY>3</eY></location>
        <location><X>123470</X><Y>345690</Y><eX>2</eX><eY>3</eY></location>
        <location><X>123477</X><Y>345690</Y><eX>2</eX><eY>3</eY></location>
    </locationList>
</obscuration>
</StatusReport>
```

8.2.2 Additional Obscuration Detected, Illumination Level changed

```xml
<StatusReport>
    <timestamp>2014-02-27T13:06:53.200Z</timestamp>
    <sourceID>5</sourceID>
    <reportID>68</reportID>
    <system>OK</system>
    <info>Additional</info>
    <activeTaskID>0</activeTaskID>
    <obscuration>
        <locationList>
            <location><X>123465</X><Y>345680</Y><eX>2</eX><eY>3</eY></location>
            <location><X>123460</X><Y>345690</Y><eX>2</eX><eY>3</eY></location>
        </locationList>
    </obscuration>
</StatusReport>
```
8.2.3 Status Unchanged

8.3 Example Detection Messages

8.3.1 Example Detection

8.3.2 Track Lost from View
8.3.3 Track Lost in View

```xml
<?xml version="1.0"?>
<DetectionReport>
  <timestamp>2014-02-27T14:08:25.365Z</timestamp>
  <sourceID>5</sourceID>
  <reportID>3</reportID>
  <objectID>34</objectID>
  <taskID>0</taskID>
  <state>lostInView</state>
</DetectionReport>
```

8.4 Example Control and Acknowledgement Messages

8.4.1 Default Tasking

```xml
<?xml version="1.0"?>
<SensorTask>
  <timestamp>2014-02-27T12:00:00.000Z</timestamp>
  <sensorID>5</sensorID>
  <taskID>0</taskID>
  <taskName>Default</taskName>
  <taskDescription>Default task - detect intruders in car park</taskDescription>
  <control>Default</control>
  <locationList>
    <location><X>123450</X><Y>345670</Y></location>
    <location><X>123480</X><Y>345670</Y></location>
    <location><X>123480</X><Y>345690</Y></location>
    <location><X>123450</X><Y>345690</Y></location>
    <location><X>123450</X><Y>345670</Y></location>
  </locationList>
</SensorTask>
```

8.4.2 Acknowledgement Message for Default Task

```xml
<?xml version="1.0"?>
<SensorTaskACK>
  <timestamp>2014-02-27T12:00:01.000Z</timestamp>
  <sensorID>5</sensorID>
  <taskID>0</taskID>
  <status>Accepted</status>
</SensorTaskACK>
```
8.4.3 Task 1 to over-ride Default tasking

```xml
<?xml version="1.0"?>
<SensorTask>

<!-- start alternate task -->
<timestamp>2014-02-27T13:00:00.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID> <!-- unique numerical identifier of the ASM to task -->
<taskID>1</taskID> <!-- generated by the HLDMM, 0 for default task -->
<control>Start</control> <!-- What to do with this task. Default sets the default task to undertake when no other tasks are given. -->

<region>
<type>Area of Interest</type>
<regionID>2</regionID> <!-- Region ID Number generated by the HLDMM -->
<regionName>Trial Area</regionName>

<!-- Points should be in order around the boundary of the region -->
<locationList>
<location><X>123550</X><Y>345670</Y></location>
<location><X>123580</X><Y>345690</Y></location>
<location><X>123550</X><Y>345690</Y></location>
<location><X>123550</X><Y>345670</Y></location>
</locationList>

<classFilter type="Human">
<parameter name="confidence" operator="Greater Than" value="0.5" />
</classFilter>

<classFilter type="Vehicle"> <!-- ignore all vehicles -->
<parameter name="confidence" operator="None" value="0" />
</classFilter>

<behaviourFilter type = "Walking">
<parameter name="confidence" operator="Greater Than" value="0.5" />
</behaviourFilter>

<behaviourFilter type = "Running">
<parameter name="confidence" operator="Greater Than" value="0.3" />
</behaviourFilter>

</region>
</SensorTask>
```

8.4.4 Acknowledgement Message for Task 1

```xml
<?xml version="1.0"?>
<SensorTaskACK>

<timestamp>2014-02-27T13:00:01.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>1</taskID>
<status>Accepted</status>
</SensorTaskACK>
```

8.4.5 Stop Task 1 and return to default task

```xml
<?xml version="1.0"?>
<SensorTask>

<!-- stop alternate task and revert to default task -->
<timestamp>2014-02-27T14:00:00.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID> <!-- unique numerical identifier of the ASM to task -->
<taskID>1</taskID> <!-- generated by the HLDMM, 0 for default task -->
<control>Stop</control> <!-- Stopping this task will revert to default task -->
</SensorTask>
```
8.4.6 Acknowledgment of Stop Message

```xml
<?xml version="1.0"?><SensorTaskACK>
<timestamp>2014-02-27T14:00:01.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>1</taskID>
<status>Complete</status>
</SensorTaskACK>
```

8.4.7 Set Detection Threshold High Command

```xml
<?xml version="1.0"?><SensorTask>
<timestamp>2014-02-27T12:30:00.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>0</taskID> <!-- generated by the HLDMM, 0 for default task -->
<control>Start</control>
<command><detectionThreshold>High</detectionThreshold></command>
</SensorTask>
```

8.4.8 Acknowledgement of Successful Threshold High Command

```xml
<?xml version="1.0"?><SensorTaskACK>
<timestamp>2014-02-27T12:30:01.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>0</taskID>
<status>Accepted</status>
</SensorTaskACK>
```

8.4.9 Acknowledgement of Rejected Threshold High Command

```xml
<?xml version="1.0"?><SensorTaskACK>
<timestamp>2014-02-27T12:30:01.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>0</taskID>
<status>Rejected</status>
<reason>Not Supported</reason>
</SensorTaskACK>
```

8.4.10 Request Reset Command

```xml
<?xml version="1.0"?><SensorTask>
<timestamp>2014-02-27T14:30:00.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>0</taskID>
<control>Start</control>
<command><request>Reset</request></command>
</SensorTask>
```

8.4.11 LookAt Command

The 'lookAt' command tasks the ASM to suspend its current tasking, take a quick look at a specific location to see if it can see anything and then return to its on-going tasking. The actual time spent undertaking the lookAt command is at the discretion of the ASM.

```xml
<?xml version="1.0"?><SensorTask>
<timestamp>2014-02-28T12:45:00.000Z</timestamp> <!-- UTC (Zulu) time message sent -->
<sensorID>5</sensorID>
<taskID>10</taskID> <!-- generated by the HLDMM, 0 for default task -->
<taskName>Look At Gate</taskName> <!-- optional name for this task ID -->
</SensorTask>
```
<taskDescription>Have a quick look at the gate</taskDescription> <control>Start</control>

<lookAt> <!-- direct sensor to look at a specific location then either send a detectionReport or "nothing found" feedback message -->
  <!-- provide location list or range bearing -->
  <locationList>
    <location><X>123550</X><Y>345670</Y></location>
  </locationList>
</lookAt>

8.4.12 Detection Report response to LookAt Command

<?xml version="1.0"?>
<DetectionReport>
  <timestamp>2014-02-28T12:45:05.365Z</timestamp> <!-- UTC time of detection -->
  <sourceID>5</sourceID>
  <reportID>123</reportID>
  <objectID>84</objectID>
  <taskID>10</taskID> <!-- task that this detection is generated from; -->
  <location>
    <X>123550</X><Y>345670</Y><Z>1.0</Z>
    <eX>1</eX><eY>0.5</eY><eZ>0.25</eZ>
  </location>
  <detectionConfidence>0.93</detectionConfidence>
  <objectInfo type="majorLength" value="1.3" e="0.1"/>
  <objectInfo type="height" value="1.6" e="0.1"/>
  <colour>red</colour>
  <class type="Human"> <confidence>0.7</confidence></class>
  <class type="Vehicle"> <confidence>0.01</confidence></class>
</DetectionReport>

8.4.13 Null Response to LookAt Command

<?xml version="1.0"?>
<SensorTaskACK>
  <timestamp>2014-02-28T12:45:05.000</timestamp> <!-- UTC time message sent -->
  <sensorID>5</sensorID>
  <taskID>10</taskID>
  <status>Complete</status>
  <reason>Nothing Found</reason>
</SensorTaskACK>

9 HLDMM – ASM Database Interaction

9.1 ASM – HLDMM Message Protocol Summary

This section summarises the message protocol between the ASMs and the HLDMM.

In general, all messages from the ASM to the HLDMM will be stored in a database with each message type having one or more database tables assigned to it.
9.1.1 Initialisation

As described in section 5.3.1. The Initialisation Message shall be sent by the ASM to the Data Agent before any other messages.

The Data Agent shall forward the message to the HLDMM Data Agent where if required, a sensor ID will be allocated to this ASM and the acknowledgement message sent back to the ASM via the Data Agent.

The Initialisation message will be forwarded on to the HLDMM by the HLDMM Data Agent. It will also be stored in the database for future reference.

9.1.2 Ongoing Messages from ASMs during normal operation

As described in section 5.4.4, ongoing messages from ASMs, (Detections, Heartbeats and Alerts) shall be stored in the database by the Data Agent for the HLDMM to read.

If an alert message requires an immediate response, such as when an ASM wishes to autonomously change mode, but would like permission from the HLDMM, the message will be additionally forwarded on to the HLDMM via the HLDMM Data Agent.

9.1.3 Control Messages

As described in section 5.5.3, control messages are sent from the HLDMM to the HLDMM Data Agent which stores the message in the database. They are then routed to the Data Agent for the ASM specified in the message which forwards the message on to the ASM. The ASM then shall respond with an acknowledgement message to accept or reject the task in the control message. The acknowledgement message is sent back to the Data Agent by the ASM which forwards the message on to the HLDMM Data Agent. The HLDMM Data Agent forwards the acknowledgement on to the HLDMM and updates the message record in the database with the acknowledgement information.

9.2 Initialisation

The initialisation message will be stored in the database table ‘sensor_registration’.

9.2.1 Sensor_Registration Table

It contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensor_id</td>
<td>Bigint</td>
<td>ASM sensor identifier as provided in registration message or allocated by HLDMM Data Agent</td>
</tr>
<tr>
<td>sensor_type</td>
<td>Text</td>
<td>ASM type description from registration message</td>
</tr>
<tr>
<td>message_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of registration message</td>
</tr>
<tr>
<td>text</td>
<td>Xml</td>
<td>Full XML text of registration message</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
</tbody>
</table>
Example Query:
SELECT "time", sensor_id, sensor_type, text
FROM sensor_registration

9.3 Heartbeat Message

The heartbeat message will be split over a number of database tables for efficiency of performance and storage. A primary table will be provided for the HLDMM to poll to check that messages are being regularly received. The sensor location and power status is included in this table. Regional information such as sensor field of view, coverage and areas of obscuration are stored in either the region or range_bearing tables as appropriate to each ASM. Additional status information is stored as one or more entries in the messages table.

9.3.1 Status_Report Table

This is the primary table for heartbeat messages; it contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this heartbeat report – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>message_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of report – always populated</td>
</tr>
<tr>
<td>active_task_id</td>
<td>Text</td>
<td>Identifier(s) of current tasks associated with this ASM – always populated</td>
</tr>
<tr>
<td>system</td>
<td>Text</td>
<td>‘system’ element of message (Null if not populated)</td>
</tr>
<tr>
<td>info</td>
<td>Text</td>
<td>‘info’ element of message (Null if not populated)</td>
</tr>
<tr>
<td>mode</td>
<td>Text</td>
<td>‘mode’ element of message (Null if not populated)</td>
</tr>
<tr>
<td>power_source</td>
<td>Text</td>
<td>‘source’ attribute of ‘power’ element (Null if not populated)</td>
</tr>
<tr>
<td>power_status</td>
<td>Text</td>
<td>‘status’ attribute of ‘power’ element (Null if not populated)</td>
</tr>
<tr>
<td>power_level</td>
<td>Int</td>
<td>‘level’ attribute of ‘power’ element (Null if not populated)</td>
</tr>
<tr>
<td>x</td>
<td>Double</td>
<td>ASM location X (Eastings) coordinate (Null if not populated)</td>
</tr>
<tr>
<td>y</td>
<td>Double</td>
<td>ASM location Y (Northings) coordinate (Null if not populated)</td>
</tr>
<tr>
<td>z</td>
<td>Double</td>
<td>ASM location Altitude (Null if not populated)</td>
</tr>
</tbody>
</table>
### 9.3.2 Status_Report_Region Table

This table holds ASM field of view, coverage and areas of obscuration data in Cartesian coordinates systems. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this heartbeat report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>message_time</td>
<td>Timestamp</td>
<td>Timestamp of report</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>Type of region e.g. ‘fieldOfView’, ‘coverage’</td>
</tr>
<tr>
<td>region_id</td>
<td>Int</td>
<td>Unique identifier where there are multiple regions of the same type from the same ASM e.g. areas of obscuration</td>
</tr>
<tr>
<td>location</td>
<td>Polygon</td>
<td>Location polygon</td>
</tr>
<tr>
<td>location_e</td>
<td>Polygon</td>
<td>Error values for location polygon</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>status_id</td>
<td>Bigint</td>
<td>Common key for status tables</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

```sql
SELECT source_id, message_time, report_id, type, region_id, location, location_e
FROM status_report_region;
```
9.3.3 Status_Report_Range_Bearing Table

This table holds ASM field of view, coverage and areas of obscuration data in Range-Bearing coordinate system. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this heartbeat report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>message_time</td>
<td>Timestamp</td>
<td>Timestamp of report</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>Type of region e.g. 'fieldOfView', 'coverage'</td>
</tr>
<tr>
<td>r</td>
<td>Double</td>
<td>Range coordinate</td>
</tr>
<tr>
<td>az</td>
<td>Double</td>
<td>Azimuth angle of centre of region</td>
</tr>
<tr>
<td>ele</td>
<td>Double</td>
<td>Elevation angle of centre of region</td>
</tr>
<tr>
<td>hextent</td>
<td>Double</td>
<td>Horizontal angle of extent of region</td>
</tr>
<tr>
<td>vextent</td>
<td>Double</td>
<td>Vertical angle of extent of region</td>
</tr>
<tr>
<td>er</td>
<td>Double</td>
<td>Range coordinate error</td>
</tr>
<tr>
<td>eaz</td>
<td>Double</td>
<td>Azimuth angle of centre of region error</td>
</tr>
<tr>
<td>eele</td>
<td>Double</td>
<td>Elevation angle of centre of region error</td>
</tr>
<tr>
<td>ehextent</td>
<td>Double</td>
<td>Horizontal angle of extent of region error</td>
</tr>
<tr>
<td>evextent</td>
<td>Double</td>
<td>Vertical angle of extent of region error</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>status_id</td>
<td>Bigint</td>
<td>Common key for status tables</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

```sql
SELECT source_id, message_time, report_id, type, r, az, ele, hextent, vextent, er, eaz, eele, ehextent, evextent
FROM status_report_range_bearing;
```

9.3.4 Status_Report_Messages Table

This table holds one or more status elements from the message. This is populated when one or more 'status' elements is present in the heartbeat message. Where a heartbeat message contains multiple values, multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
</table>

Example Query:

```sql
SELECT source_id, message_time, report_id, type, r, az, ele, hextent, vextent, er, eaz, eele, ehextent, evextent
FROM status_report_range_bearing;
```
The detection message will be split over a number of database tables for efficiency of performance and storage. A primary table will be provided for detections in Cartesian coordinates and a primary table will be provided for detections in Range-Bearing coordinates. Supplementary information such as object type will be stored in separate tables. The primary tables should be polled for new detections and then the other tables queried as required.

### 9.4.1 Detection_Report_Location Table

This is the primary table for detection messages in Cartesian coordinates systems.

The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>object_id</td>
<td>bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report – always populated</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of detection – always populated</td>
</tr>
<tr>
<td>task_id</td>
<td>Bigint</td>
<td>Identifier of any task associated with this detection (Null if not populated)</td>
</tr>
</tbody>
</table>
### 9.4.2 Detection_Report_Range_Bearing Table

This is the primary table for detection messages in Range-Bearing coordinate system. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report – always populated</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of detection – always populated</td>
</tr>
<tr>
<td>task_id</td>
<td>Bigint</td>
<td>Identifier of any task associated with this detection (Null if not populated)</td>
</tr>
<tr>
<td>state</td>
<td>Text</td>
<td>‘State’ element of message (Null if not populated)</td>
</tr>
<tr>
<td>detection_confidence</td>
<td>Double</td>
<td>Detection confidence value (Null if not populated)</td>
</tr>
<tr>
<td>r</td>
<td>Double</td>
<td>Range coordinate</td>
</tr>
<tr>
<td>az</td>
<td>Double</td>
<td>Azimuth coordinate</td>
</tr>
</tbody>
</table>
9.4.3 Detection_Report_Track_Info Table

This table contains coordinates system independent supplementary detection data. This is populated when one or more ‘trackInfo’ or ‘objectInfo’ elements is present in the detection message. Where a detection message contains multiple values, multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of detection</td>
</tr>
<tr>
<td>info_type</td>
<td>Text</td>
<td>Whether a ‘trackInfo’ or ‘objectInfo’ element</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>‘Type’ attribute of trackInfo’ or ‘objectInfo’ element</td>
</tr>
<tr>
<td>value</td>
<td>Double</td>
<td>‘Value’ attribute of trackInfo’ or ‘objectInfo’ element</td>
</tr>
<tr>
<td>e</td>
<td>Double</td>
<td>‘e’ attribute of trackInfo’ or ‘objectInfo’ element</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
</tbody>
</table>
9.4.4 Detection_Report_Class Table

This table contains the top-level object type classification data. This is only populated when one or more ‘class’ elements is present in the detection message. Where a detection message contains multiple ‘class’ elements; multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of detection</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>‘Type’ attribute of ‘class’ element</td>
</tr>
<tr>
<td>confidence</td>
<td>Double</td>
<td>Value of ‘confidence’ element</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>detection_id</td>
<td>Bigint</td>
<td>Common key for detection tables</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

SELECT source_id, detection_time, report_id, object_id, type, confidence
FROM detection_report_class;

9.4.5 Detection_Report_Subclass Table

This table contains the lower-level object type sub-classification data. This is only populated when one or more ‘subClass’ elements is present in the detection message. Where a detection message contains multiple ‘subClass’ elements; multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp</td>
<td>Timestamp of detection</td>
</tr>
</tbody>
</table>

Example Query:

SELECT source_id, detection_time, report_id, object_id, type
FROM detection_report_subclass;
<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp</td>
<td>Timestamp of detection</td>
</tr>
<tr>
<td>confidence</td>
<td>Double</td>
<td>Value of ‘confidence’ element</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>subclass_id</td>
<td>Bigint</td>
<td>ID of this record in sub class tree</td>
</tr>
<tr>
<td>parent_subclass_id</td>
<td>Bigint</td>
<td>ID of parent class in sub class tree</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT source_id, detection_time, report_id, object_id, class, level, type, value, confidence
FROM detection_report_subclass;
```

9.4.6 Detection_Report_Behaviour Table

This table contains the object behaviour data. This is only populated when one or more ‘behaviour’ elements are present in the detection message. Where a detection message contains multiple ‘behaviour’ elements, multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp</td>
<td>Timestamp of detection</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>‘Type’ attribute of ‘behaviour’ element</td>
</tr>
<tr>
<td>confidence</td>
<td>Double</td>
<td>Value of ‘confidence’ element</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>detection_id</td>
<td>Bigint</td>
<td>Common key for detection tables</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT source_id, detection_time, report_id, object_id, type, confidence
FROM detection_report_behaviour;
```
FROM detection_report Behaviour;

9.4.7 Detection_Report_Assoc_File Table

This table contains any associated file data associated with the detection message. This is only populated when the ‘associatedFile’ element is present in the detection message. Where a detection message contains multiple ‘associatedFile’ elements, multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this detection report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>ASM sensor identifier</td>
</tr>
<tr>
<td>object_id</td>
<td>Bigint</td>
<td>Unique identifier (for this ASM) of the object in this detection report</td>
</tr>
<tr>
<td>detection_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of detection</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>‘type’ attribute of ‘associatedFile’ element</td>
</tr>
<tr>
<td>url</td>
<td>Text</td>
<td>‘url’ attribute of ‘associatedFile’ element</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>detection_id</td>
<td>Bigint</td>
<td>Common key for detection tables</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

SELECT source_id, detection_time, report_id, object_id, type, url
FROM detection_report_assoc_file;

9.5 Control Message

The control message will be passed directly between modules to prevent delay. It will be stored in the database for diagnostics in the sensor_task table. It could be used to restore the system state after a failure.

9.5.1 Sensor_Task Table

The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task_id</td>
<td>Bigint</td>
<td>Unique identifier of this task – always populated</td>
</tr>
<tr>
<td>sensor_id</td>
<td>Bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>taskxml</td>
<td>Text</td>
<td>XML text of this message – always populated</td>
</tr>
<tr>
<td>message_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of this message – always populated</td>
</tr>
<tr>
<td>Field</td>
<td>SQL type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT sensor_id, message_time, task_id, taskxml
FROM sensor_task;
```

9.5.2 Sensor_TaskAck Table

The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task_id</td>
<td>Bigint</td>
<td>Unique identifier of this task – always populated</td>
</tr>
<tr>
<td>sensor_id</td>
<td>Bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>ack_timestamp</td>
<td>Timestamp without time zone</td>
<td>Timestamp of acknowledgement message to this message (Null if not populated)</td>
</tr>
<tr>
<td>ack_status</td>
<td>Text</td>
<td>Text from ‘status’ element of acknowledgement message (Null if not populated)</td>
</tr>
<tr>
<td>ack_reason</td>
<td>Text</td>
<td>Text from ‘reason’ element of acknowledgement message (Null if not populated)</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>task_key_id</td>
<td>Bigint</td>
<td>Key in sensor_task table for this task</td>
</tr>
<tr>
<td>file_type</td>
<td>Text</td>
<td>Optional attached file type</td>
</tr>
<tr>
<td>file_url</td>
<td>Text</td>
<td>Optional attached file url</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT sensor_id, ack_timestamp, task_id, ack_status, ack_reason,
     task_key_id, file_type, file_url
FROM taskack;
```

9.6 Alert Message

The control message will be split over a number of database tables for efficiency of performance and storage. A primary table will be provided with the key text fields and supplementary tables will be used where location information is provided, one for Cartesian coordinates, one for Range-Bearing coordinates.
### 9.6.1 Alert Table

This is the primary table for alert messages. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert_id</td>
<td>BigInt</td>
<td>Unique identifier of this alert message – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>BigInt</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>alert_time</td>
<td>Timestamp</td>
<td>Timestamp of alert event – always populated</td>
</tr>
<tr>
<td>alert_type</td>
<td>Text</td>
<td>Text from ‘alertType’ element of message – always populated</td>
</tr>
<tr>
<td>status</td>
<td>Text</td>
<td>Text from ‘status’ element of message – always populated</td>
</tr>
<tr>
<td>description</td>
<td>Text</td>
<td>Text from ‘description’ element of message – always populated</td>
</tr>
<tr>
<td>priority</td>
<td>Text</td>
<td>Low/Medium/High alert priority</td>
</tr>
<tr>
<td>ranking</td>
<td>Double</td>
<td>Alert Ranking (Higher number more important)</td>
</tr>
<tr>
<td>response_time</td>
<td>Timestamp</td>
<td>Timestamp of latest response message to this alert (Null if not populated)</td>
</tr>
<tr>
<td>response_reason</td>
<td>Text</td>
<td>Text from ‘reason’ element of alert response message (Null if not populated)</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>key_id</td>
<td>BigInt</td>
<td>Primary key for table / Common key for alert tables</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT source_id, alert_time, alert_id, alert_type, status, description, response_time, response_status, response_reason
FROM alert;
```

### 9.6.2 Alert_Location Table

This is the table for alert messages that include locations in Cartesian coordinates systems. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert_id</td>
<td>bigint</td>
<td>Unique identifier of this alert message – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
</tbody>
</table>
Example Query:

```
SELECT source_id, alert_time, alert_id, x, y, z, ex, ey, ez
FROM alert_location;
```

### 9.6.3 Alert_Range_Bearing Table

This is the table for alert messages that include locations in Range-Bearing coordinate system. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert_id</td>
<td>bigint</td>
<td>Unique identifier of this alert message – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>alert_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of alert event – always populated</td>
</tr>
<tr>
<td>r</td>
<td>Double</td>
<td>Range coordinate – always populated</td>
</tr>
<tr>
<td>az</td>
<td>Double</td>
<td>Azimuth coordinate – always populated</td>
</tr>
<tr>
<td>ele</td>
<td>Double</td>
<td>Elevation Angle Coordinate (Null if not populated)</td>
</tr>
<tr>
<td>z</td>
<td>Double</td>
<td>Altitude (Null if not populated)</td>
</tr>
<tr>
<td>er</td>
<td>Double</td>
<td>Range coordinate error (Null if not populated)</td>
</tr>
<tr>
<td>eaz</td>
<td>Double</td>
<td>Azimuth coordinate error (Null if not populated)</td>
</tr>
<tr>
<td>eele</td>
<td>Double</td>
<td>Elevation Angle Coordinate error (Null if not populated)</td>
</tr>
</tbody>
</table>
9.6.4 Alert_Assoc_File Table

This table contains any associated file data associated with the alert message. This is only populated when the ‘associatedFile’ element is present in the alert message. Where an alert message contains multiple ‘associatedFile’ elements, multiple records will be stored. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert_id</td>
<td>bigint</td>
<td>Unique identifier of this alert message – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
<tr>
<td>alert_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of alert event – always populated</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
<td>‘type’ attribute of ‘associatedFile’ element</td>
</tr>
<tr>
<td>url</td>
<td>Text</td>
<td>‘url’ attribute of ‘associatedFile’ element</td>
</tr>
<tr>
<td>update_time</td>
<td>Timestamp without time zone</td>
<td>Timestamp of last update to table</td>
</tr>
<tr>
<td>key_id</td>
<td>Bigint</td>
<td>Common key for alert tables</td>
</tr>
<tr>
<td>table_key_id</td>
<td>Bigserial</td>
<td>Primary key for table</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT source_id, alert_time, alert_id, type, url
FROM alert_assoc_file;
```

9.6.5 Alert_Assoc_Detection Table

This table contains any detections associated with the alert message. This is only populated when the ‘associatedDetection’ element is present in the alert message. Where an alert message contains multiple ‘associatedDetection’ elements, multiple records will be stored. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert_id</td>
<td>bigint</td>
<td>Unique identifier of this alert message – always populated</td>
</tr>
<tr>
<td>source_id</td>
<td>bigint</td>
<td>ASM sensor identifier – always populated</td>
</tr>
</tbody>
</table>

Example Query:

```
SELECT source_id, alert_id, alert_time, source_id
FROM alert_assoc_detection;
```
### 10 Interface between HLDMM and GUI

The following data will be passed between the HLDMM and the GUI.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracks and Detection Points</td>
<td>To GUI</td>
</tr>
<tr>
<td>Alerts</td>
<td>To GUI</td>
</tr>
<tr>
<td>Sensor Field of View and Coverage</td>
<td>To GUI</td>
</tr>
<tr>
<td>New High-Level Tasking / Regions of Interest</td>
<td>From GUI</td>
</tr>
<tr>
<td>Operator Response to alerts</td>
<td>From GUI</td>
</tr>
</tbody>
</table>

#### 10.1 HLDMM – GUI Database

A second set of the database tables defined in Section 9 is used for passing data between the HLDMM and the SAPIENT GUI as follows:

##### 10.1.1 Initialisation

Currently there is no requirement for a registration message to be passed between the HLDMM and the GUI and so there is no database table for this.

##### 10.1.2 Heartbeat Message

A single table is used to store status reports (heartbeat messages) from the HLDMM. It holds region data in Cartesian coordinate systems. All fields shall be populated. The table contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_id</td>
<td>Bigint</td>
<td>Unique identifier of this heartbeat report</td>
</tr>
<tr>
<td>source_id</td>
<td>Bigint</td>
<td>Always zero (source is HLDMM)</td>
</tr>
<tr>
<td>message</td>
<td>Timestamp without time zone</td>
<td>Timestamp of report</td>
</tr>
</tbody>
</table>
**Example Query:**

```
SELECT source_id, message_time, report_id, type, region_id, location, location_e, region_name, status, description
FROM hl_status_report_region;
```

### 10.1.3 Detection Messages

A second set of detection tables is used to store HLDMM fused detections for display on the GUI. These tables have the prefix 'hl_'. In practice, the HLDMM will always provide locations in Cartesian coordinates and so the range-bearing coordinates table is obsolete.

### 10.1.4 Control Messages

A second set of control tables is used to store HLDMM generated alerts for display on the GUI. These tables have the prefix 'hl_'.

### 10.1.5 Alert Messages

A second set of alert tables is used to store HLDMM generated alerts for display on the GUI. These tables have the prefix 'hl_'. In practice, the HLDMM will always provide locations in Cartesian coordinates and so the range-bearing coordinates table is obsolete.

### 10.2 Tracks and Detection Points

The HLDMM will provide these by using the Detection Message as defined in Section 7.3. The HLDMM Data Agent provided by the LSI will then put these in the correct database tables for the GUI to read. GPS coordinates should be used.
10.3 Alerts

The HLDMM will provide these by using the Alert Message as defined in Section 7.7. The HLDMM Data Agent provided by the LSI will then put these in the ‘alert’ table in the database for the GUI to read.

10.4 Sensor Field of View and Coverage

The GUI will read these from the database tables populated by the ASM Data Agents. Where the HLDMM wishes to provide additional field of view information, it will provide a heartbeat message to the HLDMM Data Agent with the region element populated as defined in Section 7.2. GPS coordinates should be used for sensor location, RangeBearing should be used for sensor field of view and coverage.

10.5 High-Level Tasking and Regions of Interest

Tasking from the GUI will be passed to the HLDMM using control messages in the same format as from the HLDMM to the ASMs. The High-Level Tasking messages will be stored in the database. New regions of interest will be defined as new records in the ‘hl_regions’ table in the database. This table is used by the GUI for displaying regions of interest.

10.6 Operator Response to Alerts

The GUI will allow the operator to give the system a response to alerts by updating the status field in the ‘alert’ table in the database. This feedback has 2 purposes:

- To remove old or unimportant alerts from the display so that others are not missed. i.e. Clear alert
- To give the system feedback - to build up a picture of false alarms as follows:
  - Acknowledge alert – this is a genuine target, classified correctly
  - Reject alert – the system has not detected a genuine target or has classified it incorrectly
  - Ignore alert – the operator does not wish to be made aware of this or similar alerts
### A Cross Reference Table for XML fields

For completeness, this appendix contains all the valid XML fields, arranged in tabular form; with how many times each field should be used. The final column gives the field in the registration message required if the field is to be supplied in other messages.

**Key:**
- M* = either location or rangeBearing must be supported. Supporting both is unnecessary in this mandatory field
- M** = these fields are mandatory if the region element is supported
- M*** = default commands that all ASMs should support
- O* = either location or rangeBearing must be supported. Supporting both is unnecessary in this optional field

<table>
<thead>
<tr>
<th>DetectionReport</th>
<th>Type</th>
<th>Units</th>
<th>Number of Occurances</th>
<th>Required field in Registration message</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>xs:dateTime</td>
<td>time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>sourceID</td>
<td>xs:int</td>
<td>none</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>reportID</td>
<td>xs:long</td>
<td>none</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>objectID</td>
<td>xs:long</td>
<td>none</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>taskID</td>
<td>xs:int</td>
<td>none</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>xs:string</td>
<td>none</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;state&quot;</td>
</tr>
<tr>
<td>location:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>x, y, z, eX, eY, eZ</td>
<td>xs:doubles</td>
<td>0-1, M*</td>
<td>locationType</td>
</tr>
<tr>
<td>rangeBearing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R, Az, Ele, Z, eR, eAz, eEle, eZ</td>
<td>xs:doubles</td>
<td>0-1, M*</td>
<td>locationType</td>
</tr>
<tr>
<td>detectionConfidence</td>
<td>xs:float</td>
<td>probability</td>
<td>0-1</td>
<td>detectionReport category=&quot;detection&quot; type=&quot;confidence&quot;</td>
</tr>
<tr>
<td>trackInfo:confidence</td>
<td>xs:float</td>
<td>probability</td>
<td>0-1</td>
<td>detectionReport category=&quot;track&quot; type=&quot;confidence&quot;</td>
</tr>
<tr>
<td>trackInfo:speed</td>
<td>xs:float</td>
<td>m-s</td>
<td>0-1</td>
<td>detectionReport category=&quot;track&quot; type=&quot;speed&quot;</td>
</tr>
<tr>
<td>trackInfo:az</td>
<td>xs:float</td>
<td>degrees</td>
<td>0-1</td>
<td>detectionReport category=&quot;track&quot; type=&quot;az&quot;</td>
</tr>
<tr>
<td>trackInfo:dR</td>
<td>xs:float</td>
<td>metres</td>
<td>0-1</td>
<td>detectionReport category=&quot;track&quot; type=&quot;dR&quot;</td>
</tr>
<tr>
<td>trackInfo:dAz</td>
<td>xs:float</td>
<td>degrees</td>
<td>0-1</td>
<td>detectionReport category=&quot;track&quot;</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>predictedLocation: location: x,y,z,eX,eY,eZ</th>
<th>xs:doubles</th>
<th>metres</th>
<th>0-1, O*</th>
<th>detectionReport category=&quot;track&quot; type=&quot;predictedLocation&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>predictedLocation: rangeBearing: R, Az, Ele, Z, eR, eAz, eEle, eZ</td>
<td>xs:doubles</td>
<td>decimal degrees-metres</td>
<td>0-1, O*</td>
<td>detectionReport category=&quot;track&quot; type=&quot;predictedLocation&quot;</td>
</tr>
<tr>
<td>predictedLocation: predictionTimestamp</td>
<td>xs:dateTime</td>
<td>time</td>
<td>0-1</td>
<td>detectionReport category=&quot;track&quot; type=&quot;predictionTimestamp&quot;</td>
</tr>
<tr>
<td>objectInfo: dopplerSpeed</td>
<td>xs:float</td>
<td>m-s</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;dopplerSpeed&quot;</td>
</tr>
<tr>
<td>objectInfo: dopplerAz</td>
<td>xs:float</td>
<td>degrees</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;dopplerAz&quot;</td>
</tr>
<tr>
<td>objectInfo: majorLength</td>
<td>xs:float</td>
<td>metres</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;majorLength&quot;</td>
</tr>
<tr>
<td>objectInfo: majorAxisAz</td>
<td>xs:float</td>
<td>degrees</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;majorAxisAz&quot;</td>
</tr>
<tr>
<td>objectInfo: minorLength</td>
<td>xs:float</td>
<td>metres</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;minorLength&quot;</td>
</tr>
<tr>
<td>objectInfo: height</td>
<td>xs:float</td>
<td>metres</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;height&quot;</td>
</tr>
<tr>
<td>colour</td>
<td>xs:string</td>
<td>none</td>
<td>0-1</td>
<td>detectionReport category=&quot;object&quot; type=&quot;colour&quot;</td>
</tr>
</tbody>
</table>

| class | 0-unbounded | detectionClassDefinition: classDefinition |
| type | xs:string | none | 1 | type |
| confidence | xs:float | probability | 1 | confidence |

| subclass | 0-unbounded | detectionClassDefinition: subClassDefinition |
| level | xs:int | none | 1 | level |
| type | xs:string | none | 1 | type |
| value | xs:string | none | 1 | values |
| confidence | xs:float | probability | 1 | confidence |

| behaviour | 0-unbounded | behaviourDefinition |
| type | xs:string | none | 1 | type |
| confidence | xs:float | probability | 1 | confidence |

<p>| associatedFile | 0-unbounded | detectionReport category=&quot;associatedFil |</p>
<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
<th>none</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>xs:string</td>
<td>none</td>
<td>1</td>
</tr>
</tbody>
</table>

**StatusReport**

<table>
<thead>
<tr>
<th>timestamp</th>
<th>xs:dateTime</th>
<th>time</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceID</td>
<td>xs:int</td>
<td>none</td>
<td>1</td>
</tr>
<tr>
<td>reportID</td>
<td>xs:long</td>
<td>none</td>
<td>1</td>
</tr>
<tr>
<td>system</td>
<td>xs:string</td>
<td>none</td>
<td>0-1</td>
</tr>
<tr>
<td>info</td>
<td>xs:string</td>
<td>none</td>
<td>0-1</td>
</tr>
<tr>
<td>activeTask</td>
<td>xs:string</td>
<td>none</td>
<td>0-1</td>
</tr>
</tbody>
</table>

**power**

<table>
<thead>
<tr>
<th>value</th>
<th>none</th>
<th>0-1</th>
</tr>
</thead>
</table>

**source**

<table>
<thead>
<tr>
<th>value</th>
<th>none</th>
<th>1</th>
</tr>
</thead>
</table>

**status**

<table>
<thead>
<tr>
<th>value</th>
<th>none</th>
<th>1</th>
</tr>
</thead>
</table>

**level**

<table>
<thead>
<tr>
<th>value</th>
<th>percentage</th>
<th>0-1</th>
</tr>
</thead>
</table>

**mode**

<table>
<thead>
<tr>
<th>value</th>
<th>none</th>
<th>0-1</th>
</tr>
</thead>
</table>

**sensorLocation:**

<table>
<thead>
<tr>
<th>value</th>
<th>metres</th>
<th>0-1</th>
</tr>
</thead>
</table>

**fieldOfView**

<table>
<thead>
<tr>
<th>value</th>
<th>metres</th>
<th>0-1, O*</th>
</tr>
</thead>
</table>

**coverage**

<table>
<thead>
<tr>
<th>value</th>
<th>metres</th>
<th>0-1, O*</th>
</tr>
</thead>
</table>

**obscuration**

<table>
<thead>
<tr>
<th>value</th>
<th>unbounded</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocationList: X,Y,Z</td>
<td>xs:double</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>range Bearing Cone: R, Az, Ele, hExtent, vExtent, minR</td>
<td>xs:double</td>
</tr>
<tr>
<td><strong>status</strong></td>
<td></td>
</tr>
<tr>
<td>level</td>
<td>xs:string</td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>value</td>
<td>xs:string</td>
</tr>
<tr>
<td><strong>SensorTask</strong></td>
<td></td>
</tr>
<tr>
<td>timestamp</td>
<td>xs:dateTime</td>
</tr>
<tr>
<td>sensorID</td>
<td>xs:integer</td>
</tr>
<tr>
<td>taskID</td>
<td>xs:integer</td>
</tr>
<tr>
<td>taskName</td>
<td>xs:string</td>
</tr>
<tr>
<td>taskDescription</td>
<td>xs:string</td>
</tr>
<tr>
<td>taskStartTime</td>
<td>xs:dateTime</td>
</tr>
<tr>
<td>taskEndTime</td>
<td>xs:dateTime</td>
</tr>
<tr>
<td>control</td>
<td>xs:string</td>
</tr>
<tr>
<td>region</td>
<td></td>
</tr>
<tr>
<td>range Bearing Cone: R, Az, Ele, hExtent, vExtent, minR</td>
<td>xs:double</td>
</tr>
<tr>
<td><strong>classFilter</strong></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>regionID</td>
<td>xs:integer</td>
</tr>
<tr>
<td>regionName</td>
<td>xs:string</td>
</tr>
<tr>
<td>LocationList: location: x,y,z,eX,eY,eZ</td>
<td>xs:double</td>
</tr>
<tr>
<td><strong>subClassFilter</strong></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>parameter</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>xs:string</td>
</tr>
<tr>
<td>operator</td>
<td>xs:string</td>
</tr>
<tr>
<td>value</td>
<td>float</td>
</tr>
<tr>
<td><strong>heartbeatReport</strong></td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>level</td>
<td>xs:string</td>
</tr>
<tr>
<td>status</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
</tr>
<tr>
<td>level</td>
<td>xs:int</td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>value</td>
<td>xs:string</td>
</tr>
<tr>
<td>parameter</td>
<td>1</td>
</tr>
<tr>
<td>name</td>
<td>xs:string</td>
</tr>
<tr>
<td>operator</td>
<td>xs:string</td>
</tr>
<tr>
<td>value</td>
<td>xs:string</td>
</tr>
<tr>
<td>behaviourFilter</td>
<td>0-unbounded</td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>parameter</td>
<td>1</td>
</tr>
<tr>
<td>name</td>
<td>xs:string</td>
</tr>
<tr>
<td>operator</td>
<td>xs:string</td>
</tr>
<tr>
<td>value</td>
<td>float</td>
</tr>
<tr>
<td>command</td>
<td>0-1</td>
</tr>
<tr>
<td>request</td>
<td>xs:string</td>
</tr>
<tr>
<td>detectionThreshold</td>
<td>xs:string</td>
</tr>
<tr>
<td>detectionReportRate</td>
<td>xs:string</td>
</tr>
<tr>
<td>classificationThreshold</td>
<td>xs:string</td>
</tr>
<tr>
<td>mode</td>
<td>xs:string</td>
</tr>
<tr>
<td>lookAt</td>
<td>command name=&quot;LookAt&quot;</td>
</tr>
<tr>
<td>locationList: location: x,y,z,eX,eY,eZ</td>
<td>xs:doubles</td>
</tr>
<tr>
<td>rangeBearingCone: R,Az,Ele,hExtent,vExtent,minR,eR,eAz,eEle,ehExtent,evExtent,eminR</td>
<td>xs:doubles</td>
</tr>
<tr>
<td>lookFor</td>
<td>0-1</td>
</tr>
<tr>
<td>Type</td>
<td>xs:string</td>
</tr>
<tr>
<td>url</td>
<td>xs:string</td>
</tr>
</tbody>
</table>
B SAPIENT Phase 2 Demonstration-Specific Information

B.1 SAPIENT Phase 2 Demonstration-Specific Architecture

B.1.1 Physical Connection
Each ASM supplier should assume a single 100BaseT, IPv4 connection for each ASM onto the LSI network. 1000BaseT equipment will be used if available. The LSI will assign a single IP address for each ASM to use on the LSI-supplied network.

B.1.2 Firewalls and Network Security

The computer network provided by the LSI will need some level of network security protection to avoid accidental or intentional system damage or data loss due to the connection of systems from multiple suppliers.

In previous projects, the LSI has found it necessary to mandate a security policy on the other suppliers, in order to meet MOD security requirements. It has been stated by Dstl that “security accreditation will be required” for the SAPIENT demonstration system. However, it is recognised that implementing a full DSAS accredited network for a short demonstration will incur significant cost and integration risk, particularly when the system will be stand-alone, OFFICIAL security classification, and not hosted on a Dstl site. Discussions were held with Dstl to identify a suitable balance between risk, costs and security requirements for the proposed SAPIENT infrastructure.

Following discussion with the Dstl Security Accréditeur, as long as the SAPIENT network is not connected to other MOD systems and is a stand-alone network i.e. not connected to the internet then standard good IT practice i.e use of Firewall and Antivirus software is sufficient security measures. Each instantiation of the SAPIENT system will be approved by the Security Accréditeur.

B.1.3 Database

For the SAPIENT Phase 2 Demonstration, it is proposed that a PostGres database is used. This provides a cost-effective solution for demonstration systems requiring a database. The LSI team are also experienced in the use of PostGres databases for surveillance applications.

B.1.4 Additional GUI PCs
A second GUI PC will be used to allow the display of “engineering outputs” that are not part of the core system functionality but will demonstrate the internal workings of one or more of the ASM and or HLDMM suppliers. This data will be passed to the GUI using the Alert message.

Where the system is to be demonstrated from a remote location, a secondary video network and video display GUI PC will be deployed that is not part of the SAPIENT system.
B.1.5 Sensor ID Allocation

For the SAPIENT Phase 2 Demonstration, the HLDM Data Agent will allocate a fixed sensor ID to each ASM. Using consistent sensor IDs for each sensor makes data management for development and test simpler.

B.1.6 Sensor Location

For the SAPIENT Phase 2 Demonstration, for those sensors which do not have GPS, it is the intention of the LSI to survey them in as part of the integration effort in the run-up to the demonstration day, and to provide the ASM suppliers with that data, which they should then include in their heartbeat messages.

B.2 ASM Database Configuration

B.2.1 Demonstration Configuration

For the SAPIENT Phase 2 Demonstration, a PostGres database will be used. This provides a cost-effective solution for demonstration systems requiring a database. The LSI team are also experienced in the use of PostGres databases for surveillance applications.

Using an alternative database for future deployments would not have any effect on the interfaces to the ASMs. The only interface change would be for the HLDM querying the ASM data.

B.2.2 Database Server Installation

PostGres should be installed following the instructions included with the installation media. Typically this will include setting up the database server to listen on TCP port 5432 and setting up a database administration user called 'postgres'.

B.2.3 Database Creation

Running an instance of the Data Agent for the first time will create the required database tables for the ASM information.

B.2.4 HLDM Database access

An alternative user account with read-only access to the database should be created for the HLDM to use.

The HLDM should then be able to read the ASM data from the database as a number of standard SQL queries as outlined below.

B.3 File Share

The LSI will provide one or more file shares for sending additional data files from the ASMs to the GUI. This is intended solely to provide the system operator with additional information such as image snapshots that they would need to respond correctly to an alert generated by the system. It is intended that the autonomous behaviour of the system should be encapsulated within the XML messages of the main data interface.

To enable flexible and well-known file sharing across different platforms, file sharing is implemented using Samba on an Ubuntu Linux machine. This enables the ASMs
that are running Linux based operating systems and the HLDMM and GUI that are running Windows operating systems to easily share files.

Currently File Sharing is only used for sharing image snapshots from the ASMs to the GUI. A shared folder called ‘Images’ is set up on a networked PC visible to both the ASMs and the GUI. Within this folder there is a sub-folder for each ASM named ‘sensor_N’ where N is the sensor identifier of the ASM. The use of sub-folders is to allow simpler file management than one big folder. The sub-folder name shall be included in any urls sent in detection or other messages so that the url can be re-used in fused messages that may not be associated with the original ASM.
### Initial distribution list

#### External
- Paul Thomas – Dstl
- Simon Cordina – Dstl
- The SAPIENT ASM suppliers x 4
- The SAPIENT HLDMM Supplier

#### QinetiQ
- Information Resources
- Project File
This document is the Interface Control Document (ICD) for the SAPIENT (Sensing for Asset Protection using Integrated Electronic Network Technology) asset protection project. It defines the interfaces between the High Level Decision Making Module, the Autonomous Sensor Modules, and the middleware and Graphical User Interface provided by the Lead Systems Integrator. The purpose of this document is to enable individual module developers to build component modules compatible with the overall system.