Open Geospatial Consortium

Approval Date: 2011-12-09

Publication Date: 2012-05-15

External identifier of this OGC® document: http://www.opengis.net/doc/ER/ows8-tracking-vmti

Reference number of this document: OGC 11-134

Category: Public Engineering Report

Editor(s): Rob Cass, Mark Simms

OGC® OWS-8 Tracking: Moving Target Indicator Process, Workflows and Implementation Results ER

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

Preface

The scope of this report is to provide a description of services, data storage and data movement within the OWS-8 Tracking sub-thread. The paper outlines the development of Sensor Observation Services (SOS), a Web Feature Service(WFS), a Notification Service and a Web Processing Service (WPS) for generating track features. Additionally, implemented encodings will be discussed as examples and in comparison to the encodings detailed in (Simonis, 2011).

This document is a deliverable for the OGC Web Services 8 (OWS-8) testbed activity. OWS testbeds are part of OGC's Interoperability Program, a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver proven candidate standards or revisions to existing standards into OGC's Standards Program, where they are formalized for public release. In OGC's Interoperability Initiatives, international teams of technology providers work together to solve specific geoprocessing interoperability problems posed by the Initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments and interoperability support services - all designed to encourage rapid development, testing, validation and adoption of OGC standards.

The OWS-8 sponsors are organizations seeking open standards for their interoperability requirements. After analyzing their requirements, the OGC Interoperability Team recommend to the sponsors that the content of the OWS-8 initiative be organized around the following threads:

* Observation Fusion

* Geosynchronization (Gsync)

* Cross-Community Interoperability (CCI)

* Aviation

More information about the OWS-8 testbed can be found at:

http://www.opengeospatial.org/standards/requests/74

OGC Document [11-139] “OWS-8 Summary Report” provides a summary of the OWS-8 testbed and is available for download:

https://portal.opengeospatial.org/files/?artifact_id=46176

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, Inc. ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Contents

Page

Figures

Page

Tables

Page

OGC® OWS-8 Tracking: Moving Target Indicator Process, Workflows and Implementation Results ER

1 Introduction

1.1 Scope

The scope of this report is to provide a description of services, data storage and data movement within the OWS-8 Tracking sub-thread. The paper outlines the development of Sensor Observation Services (SOS), a Web Feature Service(WFS), a Notification Service and a Web Processing Service (WPS) for generating track features. Additionally, implemented encodings will be discussed as examples and in comparison to the encodings detailed in (Simonis, 2011)**.**

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Rob Cass	Compusult Ltd.
Mark Simms	Compusult Ltd.
Genong Yu	George Mason University
Hsiu-Yi Ko	GIS.FCU

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2011-09-26	Draft	Rob Cass	All	Initial Draft

1.4 Future work

Implementations discussed in this document were a first attempt at fitting a moving object tracking concept of operations into the OGC framework. This included the implementation of a Sensor Observation Service (SOS) to store MISB 0601.4 and EG0903.0 video metadata, a Web Feature Service (WFS) to store a GML representation of tracks based on the NATO STANAG 4676 data model and a Web Processing Service

(WPS) designed to use the SOS to generate track data and insert that data into a WFS. A number of future work items can be derived from this implementation.

- **SWE Observation representations of VMTI detection metadata**: For OWS-8, VMTI data was obtained from 2 video files provided by the Australian DSTO. The target detection information available within these files fit the EG0903.0 Local Data Set model. As the EG0903 set of standards moves forward (EG0903.2 is now a recommended practice), the level of sophistication for the observation information stored in the SOS must consequently increase. While a straightforward "flat" approach to encoding properties of a Local Data Set could suffice for EG0903.0, developments within the EG0903.2 standard e.g. chip run-length encodings require a hierarchical structure for detections, frame information etc. This hierarchical structure is detailed in (Simonis, 2011).
- **Complete representation of STANAG 4676 concepts as GML**: For OWS-8, derived track information was stored in a WFS. This information was encoded in a GML representation which was based on, but did not fully implement all the concepts developed in the STANAG 4676 standard. The goal for the test bed was to have a geo-referenced model of the essential track information for querying and display purposes. In order to develop a complete and useful representation of the track data emitted from any tracker, there must be support for topological dependencies between the identified feature types such as track points, segments and tracks themselves.
- **Accumulation and storage of evidence:** For the track point information developed by the WPS in OWS-8, the individual frame from which a track point was derived was identified in the TrackItem feature. This was a URL to a frame item in the SOS. Further work needs to be done to identify a shared encoding model for any evidence used by the Tracking software. This would be a document or feature with appropriate evidence links maintained by the tracker as it does its analysis of the raw data. This evidence could be associated with a complete Track or its components. What is a useful degree of detail, and what the evidence encoding should be need to be developed in cooperation with domain experts.
- **Development of Standardized Property Definitions**: As a means for clearly defining each observed property derived from the raw observation data and providing a link between OGC encodings and published standards from MISB and other standards bodies, it may be necessary to develop a set of mappings between MISB and STANAG observables and OGC encoded observables. At the very least, URIs should be identified and agreed upon for observable properties. This was done in a basic way for the test bed, but would need to be developed in further work.
- **Imagery Retrieval within the Tracking Con-Ops:** Currently, there is a service definition available within the OGC architecture for retrieving still imagery, the Web Coverage Service. However, there is no service specifically designed for retrieving Full Motion Video based on time and geographic constraints. For OWS-8, an SOS was implemented that allowed for direct retrieval of still and motion imagery based on temporal constraints (this was possible because of the level of KLV encoding in the video itself). The traditional Sensor Web

Enablement approach is to embed a URL to the imagery in an Observations and Measurement response and then use this URL to retrieve this data OR to embed a base64 encoded version of the stream in the Observations and Measurements XML. While the former approach is reasonable and was implemented in OWS-8, it is not a way to directly deliver the imagery and the latter approach has scaling issues when embedding FMV in XML. A related implication is that the architecture could move to a transactional scenario where the video data is transacted to an SOS as an observation. A clear way to fit such a transaction into the SWE framework might also need to be developed. While traditionally, Motion Imagery has had no rigorous geospatial reference, and would thus fall out of the scope of the OGC, recent developments point to a need to define how to deliver FMV directly as an extension to some existing OGC service or as a new service definition.

- **GMTI Observation Fusion**: While not available within the testbed, there is an expressed need for working with Ground Motion Target Indicators provided from Radar detections. GMTI encodings for SWE have been developed in (Simonis, 2011) and could be implemented provided that test beds have access to GMTI data and decoders. GMTI presents another set of challenges for developing track data and evidence retrieval.
- **Integration with available 4676 data:** Further work in implementing tracking architectures would benefit from exposure to simulated to track data as it evolved from points, segments etc. into track features. References have been made to the track life cycle and reference data would be beneficial for analysis and implementation guidance.

1.5 Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

In addition to this document, this report includes an XML Schema Document file as specified in Annex B.

3 Terms and definitions

Most terms used in this document are from standards identified in the bibliography. The following terms and definitions are listed below to aid the reader.

3.1 track

Definition from **(NATO Standardization Agency, 2011)**,a set of associated state vector estimates describing the time-referenced, kinematic behavior, and other relevant characteristics of an object or group of objects.

3.2 target

an encoded moving object detection embedded within a VMTI or GMTI data stream

NOTE For the purposes of this report, a VMTI data stream. Targets are encoded as VTargets within a VTargetSeries provided in a Local Data Set packet according to (Motion Imagery Standards Board, 2009)

3.3 detection

a point in motion imagery determined to represent a moving indicator in pixel space

3.4 feature

representation of some real world object

3.5 tracker

a process or system that takes detections as input and creates tracks

4 Conventions

4.1 Abbreviated terms

FMV	Full Motion Video
KLV	Key Length Value
KLT	Kanade-Lucas-Tomasi tracking Algorithm
LDS	Local Data Set
MISB	Motion Imagery Standards Board
OASIS	Organization for the Advancement of Structured Information Standards
O&M	Observations and Measurements
SIFT	Scale Invariant Feature Transform
SOS	Sensor Observation Service
SWE	Sensor Web Enablement
UAV	Unmanned Aerial Vehicle

UML	Unified Modeling Language
URL	Uniform Resource Locator
VMTI	Video Moving Target Indicator
WFS	Web Feature Service
WPS	Web Processing Service
WSN	Web Services Notification
XML	eXtensible Markup Language

4.2 UML notation

Some diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

5 Implementation results overview

This Engineering report addresses implementations of components of the proposed architecture for the Observation Fusion Tracking sub-thread. The intent of this sub-thread was to develop a concept of operations for tracking using service and information models from the OGC. The following diagram illustrates a proposed architecture developed within OWS-8.

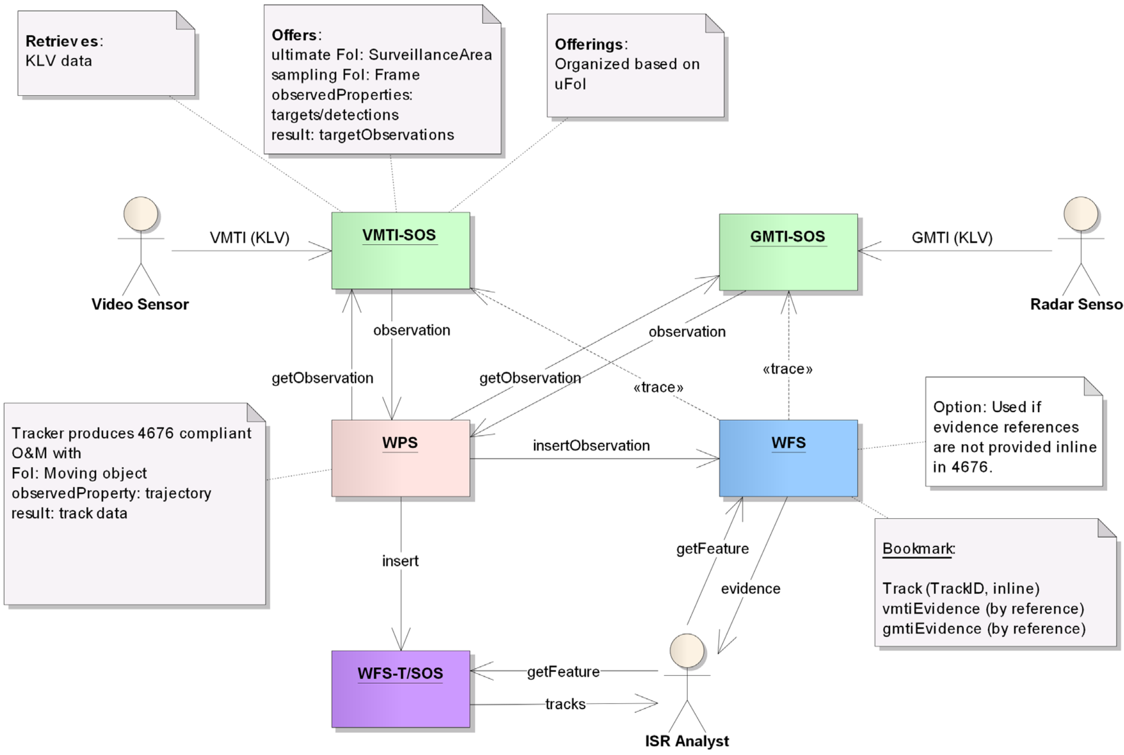


Figure 1 – Proposed Architecture (Simonis, 2011)

Within Figure 1 we see the core of the process is built around three services: an SOS for storing VMTI/GMTI data, a WPS for creating track data from the VMTI data and a WFS for storing the generated track data.

Due to data availability, GMTI data was left out of the implementation. Furthermore, WFS for storing track evidence was not implemented. STANAG 4676 provides a URL for referencing Track data and this was used for evidence references in the test bed. This leads us to the architecture illustrated in

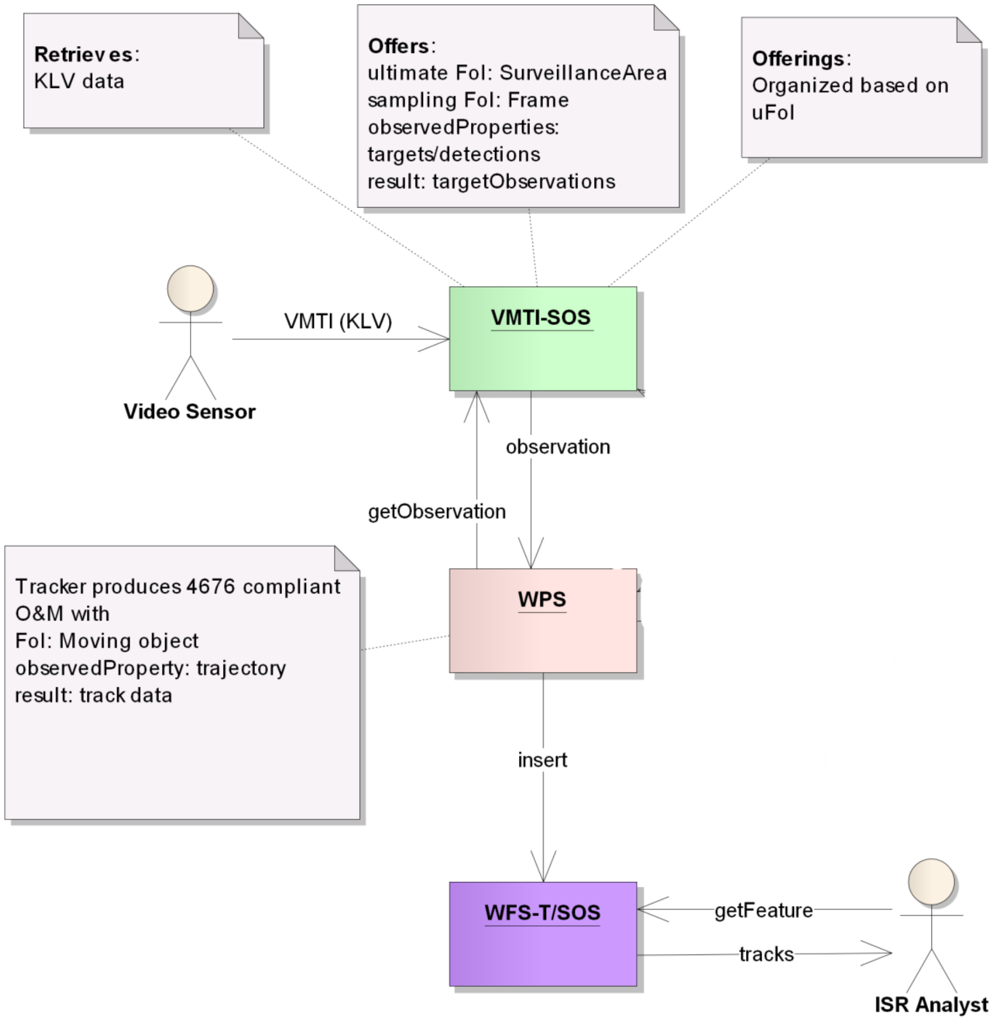


Figure 2 – Implemented Architecture

The VMTI SOS was implemented to support Observations and Measurements 2.0 and the new SWE set of standards. Observations were modeled using SWECommon.

The WPS implementation was designed to be invoked using direct references to the SOS GetObservation requests and WFS Transactions to extract target and motion imagery data, and to use this data to calculate tracks for moving objects and insert these objects into the WFS. The WPS was designed to be compliant with OGC WPS specification version 1.0.0.

The WFS was built according to the WFS 1.1.0 specification. In addition to storing a representational model of STANAG 4676 track data, the WFS was designed to provide notifications to subscribers based on a subscription / notification pattern derived from WSN and work done by the OGC PubSub working group. Subscription notifications were delivered as raw GML features or brokered as GeoSMS messages.

Software clients were developed to subscribe for updated track features. These clients could be used to visualize the track geometries, associated track metadata and video frame evidence encoded in the track data.

In the process of implementing these components, some key points were brought to attention :

- The PubSub / notification pattern can work, but requires further development. This will be addressed by the work of the PubSub working group of the OGC. Issues evolve around event patterns, clearly identifying event topics and message structures at the service level and delivery mechanisms for these events. During OWS-8 guidance was taken from work done in OWS-6, PubSub working group and WSN, but the ultimate implementation across services is a work in progress. Further development along these lines should be closely integrated with PubSub proposals as they become available.
- STANAG 4676 can be represented using GML features in a WFS.
- Track generation can be accomplished using a WPS using stored target detections and motion imagery from an SOS, however, real-world operations may cause a need to defer the use of the SOS until after tracks are generated.
- The tracker WPS is required to accumulate and store “evidence” metadata for the generated tracks. This evidence can be referenced but not stored in the 4676 track data, pointing to a need for a standardized format that encompasses all types of evidence for tracks.
- A number of shortcomings in the data set provided for the test bed caused a set of challenges for the tracking algorithm. These are discussed in detail in section 7.3.
- The SOS standard may need to provide a way to directly access binary information like video data as observations.
- The current architecture is not set up for real time delivery of track information. Bottle necks can occur in processing video data to the SOS and track production at the WPS.

6 SOS implementation

The SOS implementations for this sub-thread were aimed at providing Capabilities and Observation results based on the new SWE suite of protocols and service standards. These included SOS 2.0, SWE common and Observations and Measurements 2.0. The emphasis for these SOSes was on using a temporal filter for retrieving observations of frame, target and motion imagery directly as binary data such as mpegs or jpegs or as metadata expressed as SWE Common and O&M 2.0. The SOS was determined to be an adequate container for the data within the proposed concept of operations. Implementation issues revolved around the precision of temporal filters extracted video clip and data excluded from the data model.

6.1 Available data sets

For OWS-8, experimental data was provided as two motion imagery files from the Australian DSTO. These files, in addition to video data, contained metadata multiplexed into the video stream according to MISB standards 0601.4 and EG0903.0. In order to accurately insert this data into the SOS, ingestion software was developed to parse the KLV encoded metadata from the data streams and insert the metadata into two offerings, one being frame metadata and the other being target metadata.

The ingestion software was built on the software packages xuggler and ffmpeg. Video data was parsed and packets on the metadata stream were identified as KLV. These packets were cross referenced with the corresponding MISB KLV encoding standards and inserted into a database designed to house the pertinent metadata properties.

Four SOS offerings were created for each supplied video stream.

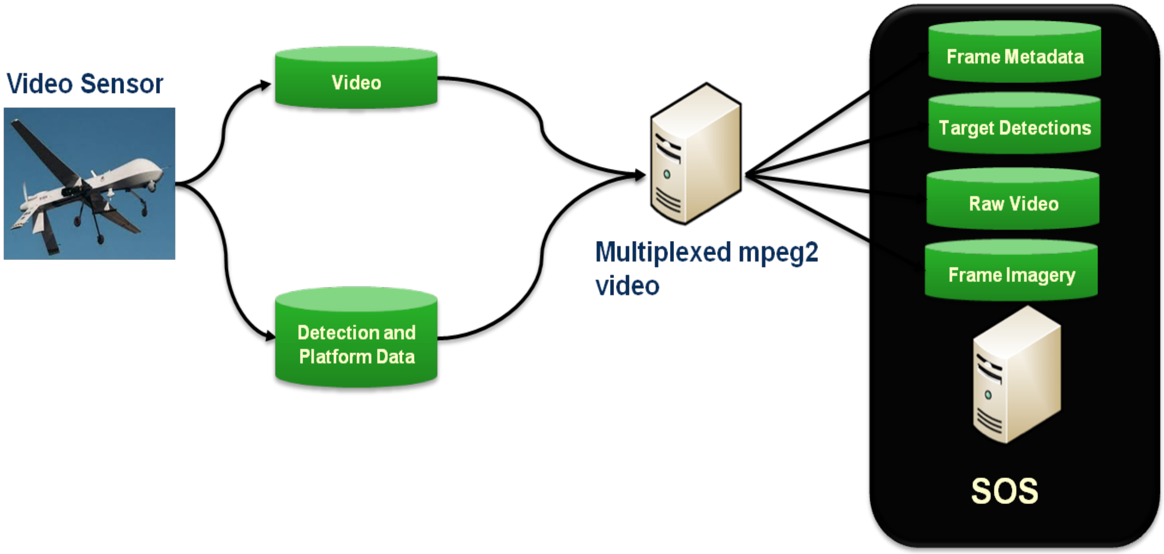


Figure 3 – Storage of Video Frame and Detection Information in SOS

Offerings are described in the sections tables below. Of importance to note is that the relevant and available data from each MISB dataset has been captured based on the data provided. Expansion of the offerings based on more sophisticated data should be part of future work. Some values were derived from the raw metadata. Derived values will be discussed.

6.2 The Frame offering

The frame offering is built from the MISB 0601.4 and EG 0903.0 dataset encoded in the video stream. It encompasses each individual frame in the imagery stream for which there is frame metadata encoded. Uses of this offering include the development of tracks as targets are cross-referenced with frame imagery and plotting track points on frame imagery to be presented to the user.

Table 1 – Frame Offering shows the observed properties available for this offering. In addition to each property presented in the table, two relevant properties required in an O&M observation are present for any observation from this offering: the phenomenon time which is derived from the frame timestamp and the sampling geometry which is derived from the latitude and longitude offsets from the frame center as in the following example.

Figure 4 – Example of Frame Polygon Sampling Feature

```
<om:parameter>
   <om:NamedValue>
   <om:name
xlink:href="http://www.opengis.net/req/omxml/2.0/data/samplingGeometry"
/>
<om:value>
<gml:Polygon srsName="EPSG:4326"
xmlns:gml="http://www.opengis.net/gml">
   <gml:exterior>
         <gml:LinearRing>
                 <gml:posList srsDimension="2">
                138.50555359453864 -34.84310949013986
                138.5017563289971 -34.84733935555323
                138.49644610834827 -34.84410515651151
                138.4999366628696 -34.84007671326068
                138.50555359453864 -34.84310949013986
                </gml:posList>
          </gml:LinearRing>
   </gml:exterior>
</gml:Polygon>
</om:value>
</om:NamedValue>
</om:parameter>
```

The sampling geometry is represented as an inline GML 3.2 Polygon of two dimensions. The representation could be improved to include altitude if an elevation model was available. It was decided to derive the polygon representation of the frame corners as this would aid selection by spatial filtering and provide a sampling geometry directly. Because of this, the frame offsets were not retained because this was deemed to be redundant.

Where possible, observed properties in the offering were given names directly based on the names of the fields in the MISB standard. A frame URL observed property was derived from the source data based on the timestamp. This URL is designed to be retrievable as a jpeg image.

Table 1 – Frame Offering

Observed Property	Unit of Measure	Derived (y/n)	MISB LDS Name

urn:misb:def:0601:Checksum	N/A	n	Checksum
urn:misb:def:eg0903:FrameHeight	Pixels	n	Frame Height
urn:misb:def:eg0903:FrameWidth	Pixels	n	Frame Width
urn:misb:def:eg0903:NumberOfDetectedTargets	N/A	n	Total Number of Targets in the Frame
urn:misb:def:eg0903:NumberOfReportedTargets	N/A	n	Number of Reported Targets
urn:misb:def:eg0903:VMTILDSVersionNumber	N/A	n	UAS LDS Version Number
urn:misb:def:eg0903:VMTISensorHFOV	Degrees	n	Sensor Horizontal Field of View
urn:misb:def:eg0903:VMTISensorVFOV	Degrees	n	Sensor Vertical Field of View
urn:misb:def:eg0903:VMTISourceSensor	Text	n	Source Sensor Identifier
urn:misb:def:eg0903:VMTISystemName	Text	n	VMTI System Name
urn:misb:def:eg0903:VideoFrameNumber	N/A	n	Video Frame Number
urn:misb:def:eg0903:FrameURL	Text	y	N/A

The frame URL derived value is given an observed property urn which implies that it is from the 0903 dataset. This is not the case and another property definition should be developed.

An example observation request and response is located in .

6.3 The TargetResult Offering

The TargetResult Offering was developed to support the geographical and frame location of target detections recorded in the target local data pack defined in MISB EG0903 (Motion Imagery Standards Board, 2009). These detections are essential for developing track information. These detections are used as a seed for deriving points along a track occupied by a moving object.

When ingesting the data for this offering, it was necessary to maintain prior frame metadata as parts of the target pack refer to the VMTI LDS data pack associated with the frame. This frame data is not always available directly with the target detections and must be inferred using the last known VMTI LDS entry. Additionally, the target data in the data samples lacked higher order information regarding the target pixel information. These properties were captured in the offering, but were not populated as the supporting data was not in the samples.

For the sampling geometry, a quadrilateral was chosen similar to the quadrilateral chosen for the frame offering. This was in anticipation of area target information and not just point target information. The resulting target geometry was always a polygon with no area. The value for this quadrilateral was calculated using the associated frame center Latitude and Longitude and target offset.

Table 2 – Target Result Offering

Observed Property	Unit of Measure	Derived (y/n)	MISB LDS Name
urn:misb:def:eg0903:Elevation	meters	N	Target Elevation
urn:misb:def:eg0903:Latitude	degrees	Y	Derived from Target Location Latitude Offset and Frame Center Latitude
urn:misb:def:eg0903:Longitude	degrees	Y	Derived from Target Location Latitude Offset and Frame Center Longitude
urn:misb:def:eg0903:NewDetectionTargetHistoryFlag	frames	N	New Detection Flag / Target History
urn:misb:def:eg0903:PercentTargetPixels	Percentage	N	Percentage of Target Pixels
urn:misb:def:eg0903:PixelNumberBottomRight	N/A	N	Bounding Box Bottom Right Pixel Number
urn:misb:def:eg0903:PixelNumberCentroid	N/A	N	Target Centroid Pixel Number
urn:misb:def:eg0903:PixelNumberTopLeft	N/A	N	Bounding Box Top Left Pixel Number
urn:misb:def:eg0903:TargetColor	RGB Color	N	Target Color
urn:misb:def:eg0903:TargetConfidenceLevel	Percentage	N	Target Confidence Value
urn:misb:def:eg0903:TargetID	N/A	N	Target ID Number

urn:misb:def:eg0903:TargetIntensity	Unused – for Infrared Systems	N	Target Intensity
urn:misb:def:eg0903:TargetPriority	N/A	N	Target Priority

One observed property that should have been included in the target offering was the frame number. Because the frame number was not explicitly included, cross-referencing targets to frames was done using time stamps. This was possible, but not ideal.

6.4 The Video Offering

The Video offering was designed to generate video clips defined by a temporal range filter. There are two typical uses for this offering. The first is that a tracker can request segments of video based on detection time ranges. Thus the tracker would not need to download the complete clip, but segments at a time. The second application was for clients to retrieve video evidence directly without needing to download O&M metadata, parse it and then retrieve the video clip using an embedded video link.

In OWS-8, the tracker made use of the video offering, but the presentation of the video data was imperfect and allowances had to be made in software to return the complete video stream. An ideal software implementation would present the video data accurately synched to frame number and timestamps in the detection data. Additionally, video clips would be multiplexed with the original VMTI metadata. Video clips were extracted and returned successfully, but no software library was found that would retain the VMTI metadata.

In addition to direct delivery of video data, the Video Offering presented an O&M response that contained a text result that was a URL to the video clip.

In future work, it is worth investigating the application of spatial filters to video observations that have spatial metadata associated with the frame information.

The SOS standard has no provision for direct delivery of binary data as an output format for a GetObservation request. How this would be implemented for observations that have a time period and not an associated time instant is a matter for discussion outside this document. For VMTI encoded video data, video is also given a spatial component which opens up the possibility of requesting the video using spatial filters, temporal and spatial filters in combination, using other aspects of the video metadata as filter properties etc. .

One could imagine a query that asks for video data that covers a particular area and time that contains more than 10 detections and has a dominant detection pixel color of white. Parameters defining the presentation of the returned data such as the CRS of the video frames and mask colors for areas with no data could also be relevant to such a query.

6.5 The Frame Image Offering

The Frame Image Offering was implemented as a method for retrieving a raw jpeg image for a frame from the motion imagery using a time instant temporal filter. This offering fulfilled the same functions as the video offering, but for a single frame. This offering was used by client software to retrieve video imagery, but was not used by the tracking algorithm.

7 WPS Tracker implementation

7.1 Tracker Algorithm

In order to turn raw video and VMTI detections into a moving object track, a processor, typically known as a tracker is involved. For OWS-8, this processor was implemented using the OGC WPS 1.0.0 specification (Yu & Chen, 2011). The algorithm for detection-to-track is based on the Kanade-Lucas-Tomasi feature tracker. All frames, not just the frames with metadata, must be retrieved together in order to better facilitate the assumption of the tracking algorithm. The algorithm assumes that an object moves only slightly across frames. If there is a big "jump" of tracking object between frames, the algorithm would not work or easily lose the object under tracking. Once the feature is found in the target frame, the closest detection from the metadata is used in the track if there is one such detection point in the metadata that is not farther than 10 pixels in this case. KLT also supports multi-resolution tracking to deal with relatively large displacements of targets between video frames. Affine computation was enabled with KLT to deal with non-consecutive frames. In the implementation, empirical adaptation and adjustment were supported to achieve best results of tracking in a geospatially-aware sense. These include enhancements for :

- being adaptive to the size of target under investigation by defining the tracking target window size,
- being capable of varied range of displacements by adaptively adjusting the parameters for multi-resolution / pyramid, and
- being tolerant to occlusion by using a momentum to track the target through occluded area.

7.2 Tracker Inputs and Execution Sequence

The three SOS offerings used by the WPS were Frame, TargetResult and Video. These were defined as inputs to an instance of a WPS "Execute" request. An example of the Execute Request/Response documents can be seen in Annex section A.1. After the immediate response is returned, clients have the opportunity to test the status of the executing process by using the returned URI to the status servlet. A positive successful status response would be followed by a request for results which would return a reference to the results of the process which is an insertion into a WFS. The following diagram indicates the sequence of events involved as a client invokes the tracking WPS.

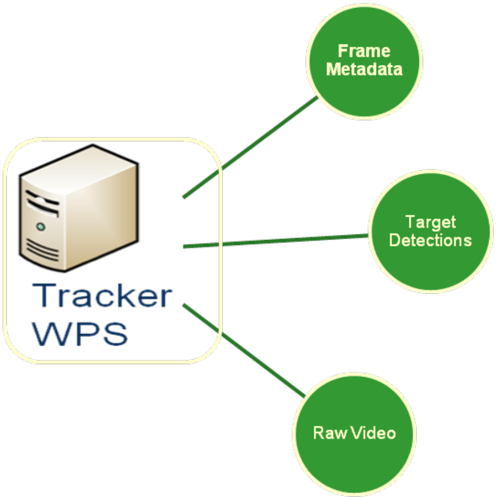


Figure 5 -- Tracker Inputs

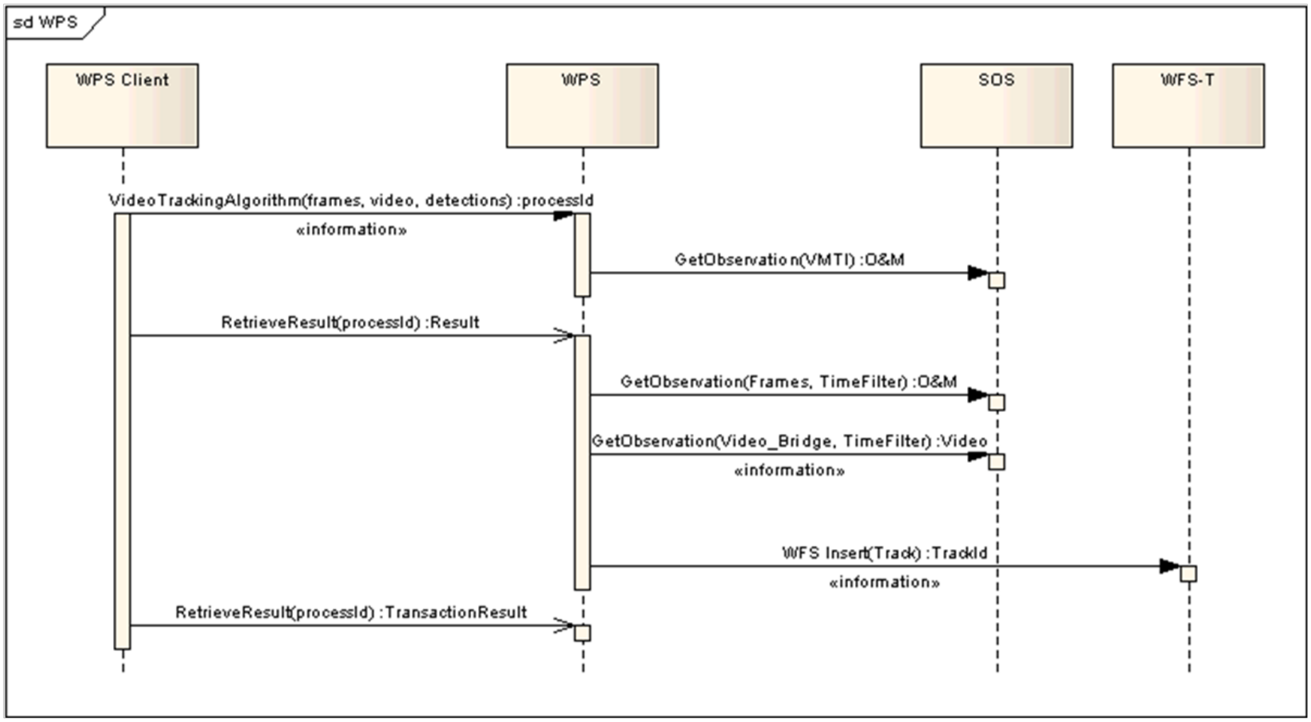

sd WPS
WPS Client
WPS
SOS
WFS-T
VideoTrackingAlgorithm(frames, video, detections) :processId
«information»
GetObservation(VMTI) :O&M
RetrieveResult(processId) :Result
GetObservation(Frames, TimeFilter) :O&M
GetObservation(Video_Bridge, TimeFilter) :Video
«information»
WFS Insert(Track) :TrackId
«information»
RetrieveResult(processId) :TransactionResult

Figure 6 -- WPS Sequence Diagram

The ultimate result of the tracking process as seen in the diagram is insertion of Track Data in a WFS. The nature of the Track features in the WFS is the subject of the next section.

7.3 Lessons learned and discussions

During the development and use of the tracking algorithm, a number of problems arose around the data that was available. What follows are discussions and recommendations based on these issues.

7.3.1 Target Characteristics and Algorithm Traceability

Problems:

Features for extracting the detections in the video frames can be very useful at the stage of forming the tracks from detections. Here, detections are those points or locations where the targets are identified. The detections may be determined by some target-detecting algorithms.

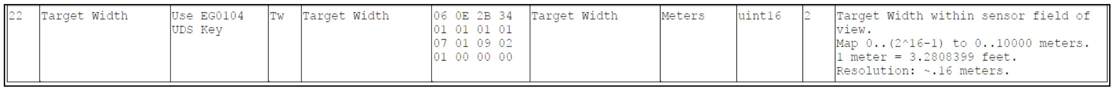
One of the tasks for tracking algorithm is to compute the similarity between detections and therefore determine if they are detections for the same moving target in the same track. Features for computing the similarity between detections may include geometrical (e.g. size, width, height), radiometric (e.g. spectral range at different bands), and/or complex features (e.g. SIFT). These features may be used during the detection stage. It can be useful information for tracking if these features and their deriving algorithm are known to the tracking algorithm.

These may not be an issue if the detection and tracking are done by the same party because they implicitly know what algorithm is used for extracting detection features and the nature of the information passed along. However, if the detection and the tracking are realized by different parties or at a different time, the knowledge of the algorithms and nature of the data is not traceable.

Tracking algorithms need to have some way to compare the similarity between detections across frames. Ideally, the tracking algorithm would require information of geometrical characteristics, radiometric characteristics, and algorithms for extracting these detections (no matter if the features are geometrical, radiometric, or hybrid features).

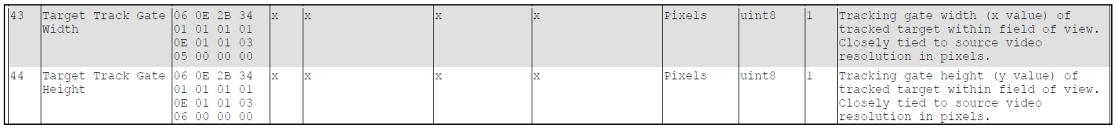
In the GMU Implementation of Detection-to-Tracking algorithm, the dimension of the target can be very a useful feature for KLT algorithm. The use of target dimension can be used in combination of KLT multi-resolution capability to enhance the accuracy of target tracking. However, the dimensions aren't completely available to the algorithm. Therefore, these are exposed as input parameters for the WPS. Ideally, these should be automatically extracted from the detection metadata.

In MISB 0601, there is some target size information as shown in Figure 7 and Figure 8. However, these may not be properly tagged for all detections. Some detection has incomplete information or even none at all. Figure 9 shows one example that has target width.



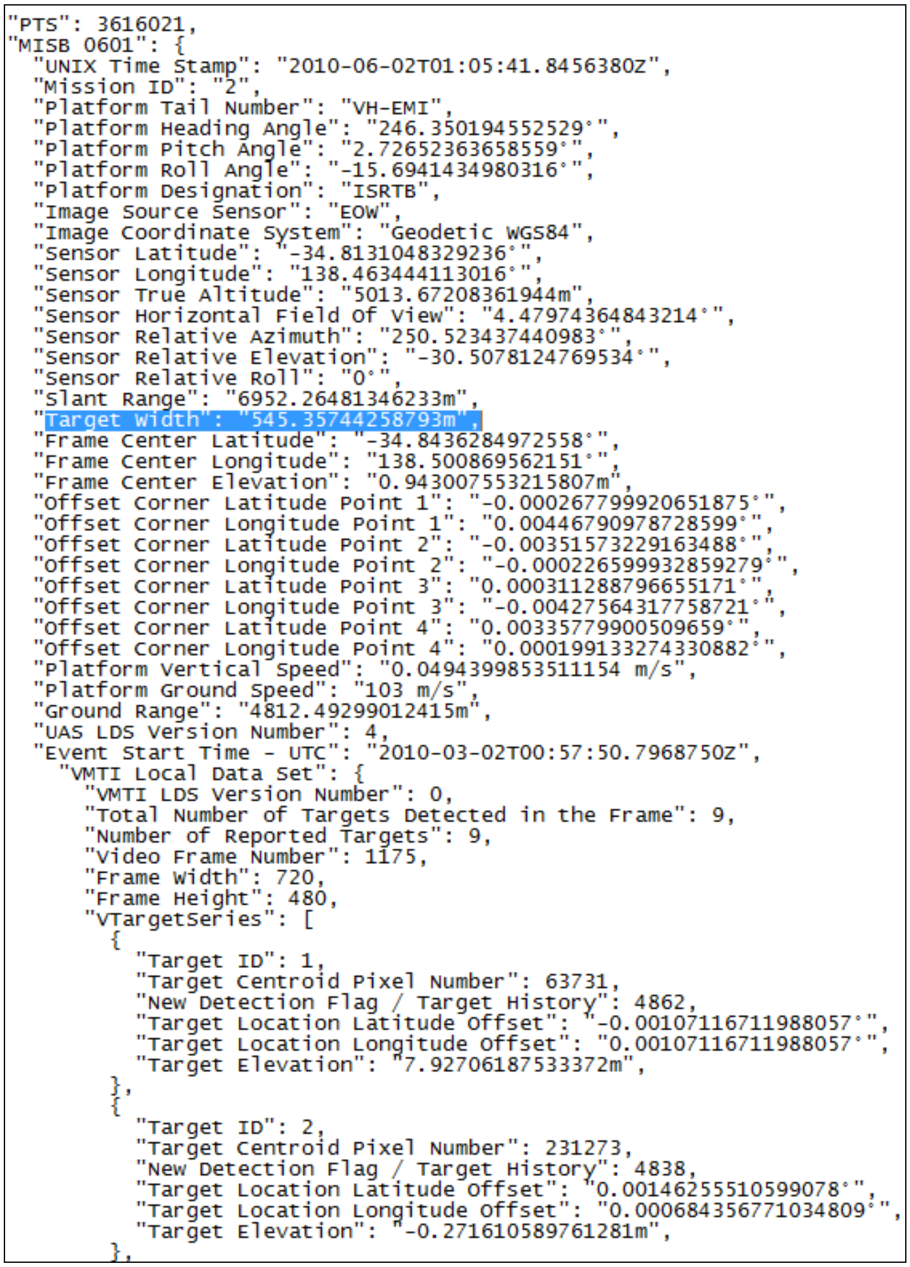
22	Target Width	Use EG0104 UDS Key	Tw	Target Width	06 0E 2B 34 01 01 01 01 07 01 09 02 01 00 00 00	Target Width	Meters	uint16	2	Target Width within sensor field of view. Map 0..(2^16-1) to 0..10000 meters. 1 meter = 3.2808399 feet. Resolution: ~.16 meters.

Figure 7. Target width in MISB 0601



43	Target Track Gate Width	06 0E 2B 34 01 01 01 01 0E 01 01 03 05 00 00 00	x	x	x	x	Pixels	uint8	1	Tracking gate width (x value) of tracked target within field of view. Closely tied to source video resolution in pixels.
44	Target Track Gate Height	06 0E 2B 34 01 01 01 01 0E 01 01 03 06 00 00 00	x	x	x	x	Pixels	uint8	1	Tracking gate height (y value) of tracked target within field of view. Closely tied to source video resolution in pixels.

Figure 8. Target Track Gate width and height in MISB 0601



```
"PTS": 3616021,
"MISB 0601": {
  "UNIX Time Stamp": "2010-06-02T01:05:41.845638OZ",
  "Mission ID": "2",
  "Platform Tail Number": "VH-EMI",
  "Platform Heading Angle": "246.350194552529°",
  "Platform Pitch Angle": "2.72652363658559°",
  "Platform Roll Angle": "-15.6941434980316°",
  "Platform Designation": "ISRTB",
  "Image Source Sensor": "EOW",
  "Image Coordinate System": "Geodetic WGS84",
  "Sensor Latitude": "-34.8131048329236°",
  "Sensor Longitude": "138.463444113016°",
  "Sensor True Altitude": "5013.67208361944m",
  "Sensor Horizontal Field Of View": "4.47974364843214°",
  "Sensor Relative Azimuth": "250.523437440983°",
  "Sensor Relative Elevation": "-30.5078124769534°",
  "Sensor Relative Roll": "0°",
  "Slant Range": "6952.26481346233m",
  "Target Width": "545.35744258793m",
  "Frame Center Latitude": "-34.8436284972558°",
  "Frame Center Longitude": "138.500869562151°",
  "Frame Center Elevation": "0.943007553215807m",
  "Offset Corner Latitude Point 1": "-0.000267799920651875°",
  "Offset Corner Longitude Point 1": "0.00446790978728599°",
  "Offset Corner Latitude Point 2": "-0.00351573229163488°",
  "Offset Corner Longitude Point 2": "-0.000226599932859279°",
  "Offset Corner Latitude Point 3": "0.000311288796655171°",
  "Offset Corner Longitude Point 3": "-0.00427564317758721°",
  "Offset Corner Latitude Point 4": "0.00335779900509659°",
  "Offset Corner Longitude Point 4": "0.000199133274330882°",
  "Platform Vertical Speed": "0.0494399853511154 m/s",
  "Platform Ground Speed": "103 m/s",
  "Ground Range": "4812.49299012415m",
  "UAS LDS Version Number": 4,
  "Event Start Time - UTC": "2010-03-02T00:57:50.796875OZ",
    "VMTI Local Data Set": {
      "VMTI LDS Version Number": 0,
      "Total Number of Targets Detected in the Frame": 9,
      "Number of Reported Targets": 9,
      "Video Frame Number": 1175,
      "Frame Width": 720,
      "Frame Height": 480,
      "VTargetSeries": [
        {
          "Target ID": 1,
          "Target Centroid Pixel Number": 63731,
          "New Detection Flag / Target History": 4862,
          "Target Location Latitude Offset": "-0.00107116711988057°",
          "Target Location Longitude Offset": "0.00107116711988057°",
          "Target Elevation": "7.92706187533372m",
        },
        {
          "Target ID": 2,
          "Target Centroid Pixel Number": 231273,
          "New Detection Flag / Target History": 4838,
          "Target Location Latitude Offset": "0.00146255510599078°",
          "Target Location Longitude Offset": "0.000684356771034809°",
          "Target Elevation": "-0.271610589761281m",
        },
```

Figure 9. Target size information in one detection example

In the case of multiple moving targets occurring in the same frame, the scheme of MISB 0601 may not be sufficient in properly defining every target under tracking. One target may have a significantly different dimension from another in the same frame. So, the dimensions may have to be defined in each VMTI tag, or in MISB EG 0903.

In the current OGC Information Model to model the VMTI data, the dimension of target isn't enabled. This might be due to the inconsistency of detection tagging in the example VMTI dataset. The SOS does not have target width passed along even if there is target width info in the original video under MISB 0601.

Recommendations:

(1) MISB or NATO standards: Detection characteristics should be defined and described for any target under detections.

   a. Add more geometrical/radiometric information model into MISB 0903 VMTI description or MISB 0601. Reference model may refer to ISO 19115.
   b. Add algorithm information model provenance/lineage into MISB, including the detection algorithm used and the model to retrieve the detection features. Reference model may refer to ISO 19115-2.
   c. Geometrical information model may be added to MISB 0903 at LDS for VMTI. This may be more relevant if there are multiple targets with different dimensions occurred in a single frame.

(2) OGC Informational Model: OGC information model should precisely model all the information available in MISB specifications.

   a. Size and other geometrical metadata should be passed along to be modeled in the O&M of the SOS.
   b. If the MISB provide expansion facility, the OGC information model should describe a similar mechanism to accommodate specific profiles or expansions.

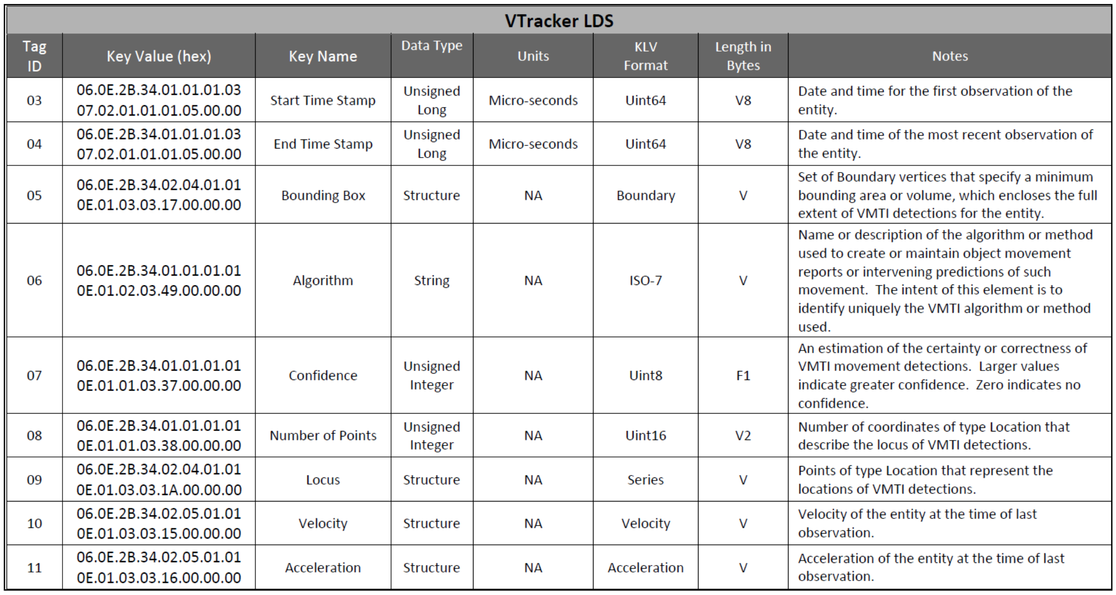
7.3.2 Target Motion Metrics for Detections

Problems:

Dead reckoning may be necessary in continuously moving target tracking. When a target moves along a road or path, it can be occluded by overhead coverage or below similar pavement to the color of target body. To keep track of the target, a momentum may be added to the target to estimate how far and in what direction the target will keep moving if the tracking algorithm loses sight of the target temporarily. In the GMU implementation of Detection-to-Tracking WPS, a momentum is used to keep the target moving in the same direction and speed over the average of certain numbers of previous frames. This has been proved to be very useful in the case of tracking the vehicle moving across the bridge where the vehicle has a very similar tone to its background road surface. The KLT algorithm loses track of the target during these frames but eventually emerges after several frames. The added momentum solved the problem of losing the tracking target in this case.

From this experience, it is obvious that motion metrics of the target under detecting/tracking can be very useful in keeping the correct tracking of target under these cases of occlusion. Any added information about the characteristics of the target under tracking would be very helpful. Explicit intentions or interest would help. For example, tracking a truck may be different from a tracking a special police car. Common sense may be taken into consideration. In an Interstate highway, a vehicle is normally moving at 70mph while special police car may move at a slightly faster speed, e.g. 80 mph on average. The general sense of speed may be useful in correcting the automatically derived speed from tracking algorithm.

In MISB 0903, velocity and acceleration are fields in the VTracker LDS as shown in Figure 10. However, the example dataset provided for use in the OWS-8 does not have any detection containing these fields.



VTracker LDS

Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
03	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	Start Time Stamp	Unsigned Long	Micro-seconds	Uint64	V8	Date and time for the first observation of the entity.
04	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	End Time Stamp	Unsigned Long	Micro-seconds	Uint64	V8	Date and time of the most recent observation of the entity.
05	06.0E.2B.34.02.04.01.01 0E.01.03.03.17.00.00.00	Bounding Box	Structure	NA	Boundary	V	Set of Boundary vertices that specify a minimum bounding area or volume, which encloses the full extent of VMTI detections for the entity.
06	06.0E.2B.34.01.01.01.01 0E.01.02.03.49.00.00.00	Algorithm	String	NA	ISO-7	V	Name or description of the algorithm or method used to create or maintain object movement reports or intervening predictions of such movement. The intent of this element is to identify uniquely the VMTI algorithm or method used.
07	06.0E.2B.34.01.01.01.01 0E.01.01.03.37.00.00.00	Confidence	Unsigned Integer	NA	Uint8	F1	An estimation of the certainty or correctness of VMTI movement detections. Larger values indicate greater confidence. Zero indicates no confidence.
08	06.0E.2B.34.01.01.01.01 0E.01.01.03.38.00.00.00	Number of Points	Unsigned Integer	NA	Uint16	V2	Number of coordinates of type Location that describe the locus of VMTI detections.
09	06.0E.2B.34.02.04.01.01 0E.01.03.03.1A.00.00.00	Locus	Structure	NA	Series	V	Points of type Location that represent the locations of VMTI detections.
10	06.0E.2B.34.02.05.01.01 0E.01.03.03.15.00.00.00	Velocity	Structure	NA	Velocity	V	Velocity of the entity at the time of last observation.
11	06.0E.2B.34.02.05.01.01 0E.01.03.03.16.00.00.00	Acceleration	Structure	NA	Acceleration	V	Acceleration of the entity at the time of last observation.

Figure 10. Speed and acceleration tag in MISB 0903

Recommendations:

MISB standards and dataset for experiment:

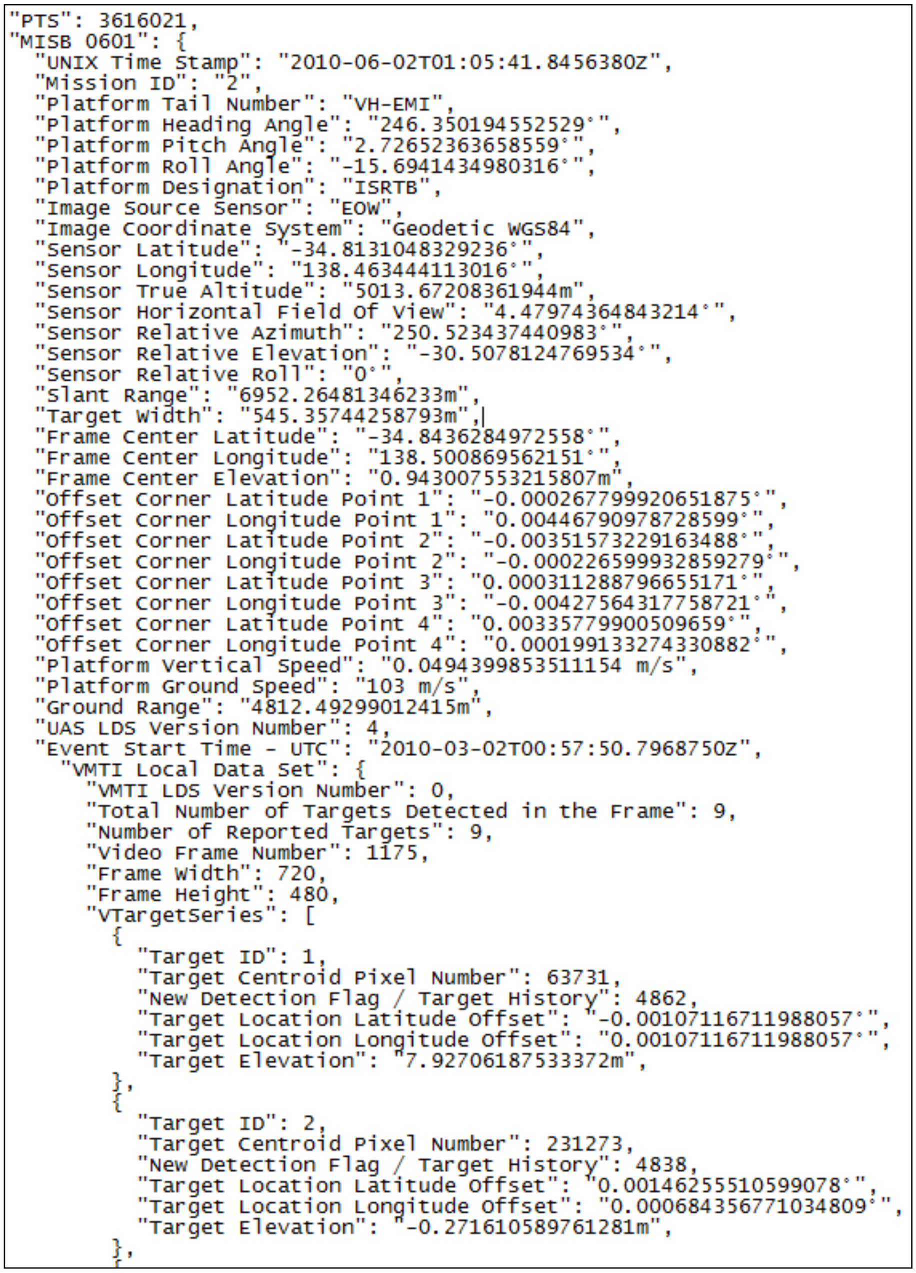
(1) In MISB 0903 or 0601, extension may be added to support the description of the target as different types that have different range of speed/acceleration.
(2) Experiment dataset should have the info.

7.3.3 Sensor Metadata for Frame Co-Registration

Problems:

Frame co-registration is a crucial step for accurate tracking of targets if the video camera position changes drastically during video-recording. Automatic co-registration only works if the two frames do not differ significantly. For example, SIFT can be used to select candidate identical points across consecutive frames. However, the identical points may not be possible using SIFT if the frames have significant difference between the two frames.

In MISB 0601 specification, there are sensor angle and location information, also platform angle and location information. The example dataset does not have complete information for all the detections. Some detection has partial metadata as shown in Figure 11. Some does not have any at all.



```
"PTS": 3616021,
"MISB 0601": {
  "UNIX Time Stamp": "2010-06-02T01:05:41.8456380Z",
  "Mission ID": "2",
  "Platform Tail Number": "VH-EMI",
  "Platform Heading Angle": "246.350194552529°",
  "Platform Pitch Angle": "2.72652363658559°",
  "Platform Roll Angle": "-15.6941434980316°",
  "Platform Designation": "ISRTB",
  "Image Source Sensor": "EOW",
  "Image Coordinate System": "Geodetic WGS84",
  "Sensor Latitude": "-34.8131048329236°",
  "Sensor Longitude": "138.463444113016°",
  "Sensor True Altitude": "5013.67208361944m",
  "Sensor Horizontal Field of View": "4.47974364843214°",
  "Sensor Relative Azimuth": "250.523437440983°",
  "Sensor Relative Elevation": "-30.5078124769534°",
  "Sensor Relative Roll": "0°",
  "Slant Range": "6952.26481346233m",
  "Target Width": "545.35744258793m",
  "Frame Center Latitude": "-34.8436284972558°",
  "Frame Center Longitude": "138.500869562151°",
  "Frame Center Elevation": "0.943007553215807m",
  "Offset Corner Latitude Point 1": "-0.000267799920651875°",
  "Offset Corner Longitude Point 1": "0.00446790978728599°",
  "Offset Corner Latitude Point 2": "-0.00351573229163488°",
  "Offset Corner Longitude Point 2": "-0.000226599932859279°",
  "Offset Corner Latitude Point 3": "0.00031128879665517l°",
  "Offset Corner Longitude Point 3": "-0.00427564317758721°",
  "Offset Corner Latitude Point 4": "0.00335779900509659°",
  "Offset Corner Longitude Point 4": "0.000199133274330882°",
  "Platform Vertical Speed": "0.0494399853511154 m/s",
  "Platform Ground Speed": "103 m/s",
  "Ground Range": "4812.49299012415m",
  "UAS LDS Version Number": 4,
  "Event Start Time - UTC": "2010-03-02T00:57:50.796875OZ",
    "VMTI Local Data Set": {
      "VMTI LDS Version Number": 0,
      "Total Number of Targets Detected in the Frame": 9,
      "Number of Reported Targets": 9,
      "Video Frame Number": 1175,
      "Frame Width": 720,
      "Frame Height": 480,
      "VTargetSeries": [
        {
          "Target ID": 1,
          "Target Centroid Pixel Number": 63731,
          "New Detection Flag / Target History": 4862,
          "Target Location Latitude Offset": "-0.00107116711988057°",
          "Target Location Longitude Offset": "0.00107116711988057°",
          "Target Elevation": "7.92706187533372m",
        },
        {
          "Target ID": 2,
          "Target Centroid Pixel Number": 231273,
          "New Detection Flag / Target History": 4838,
          "Target Location Latitude Offset": "0.0014625551059907 8°",
          "Target Location Longitude Offset": "0.000684356771034809°",
          "Target Elevation": "-0.271610589761281m",
        },
        {
```

Figure 11. Camera metadata of MISB 0601 in the example dataset

In the information model used in the OWS-8 demonstration, the retrieved observations from the SOS do not contain any of these metadata. The incompleteness of camera position metadata might be the reason for the OGC information model to omit these in OWS-8.

Recommendations:

MISB standards and datasets:

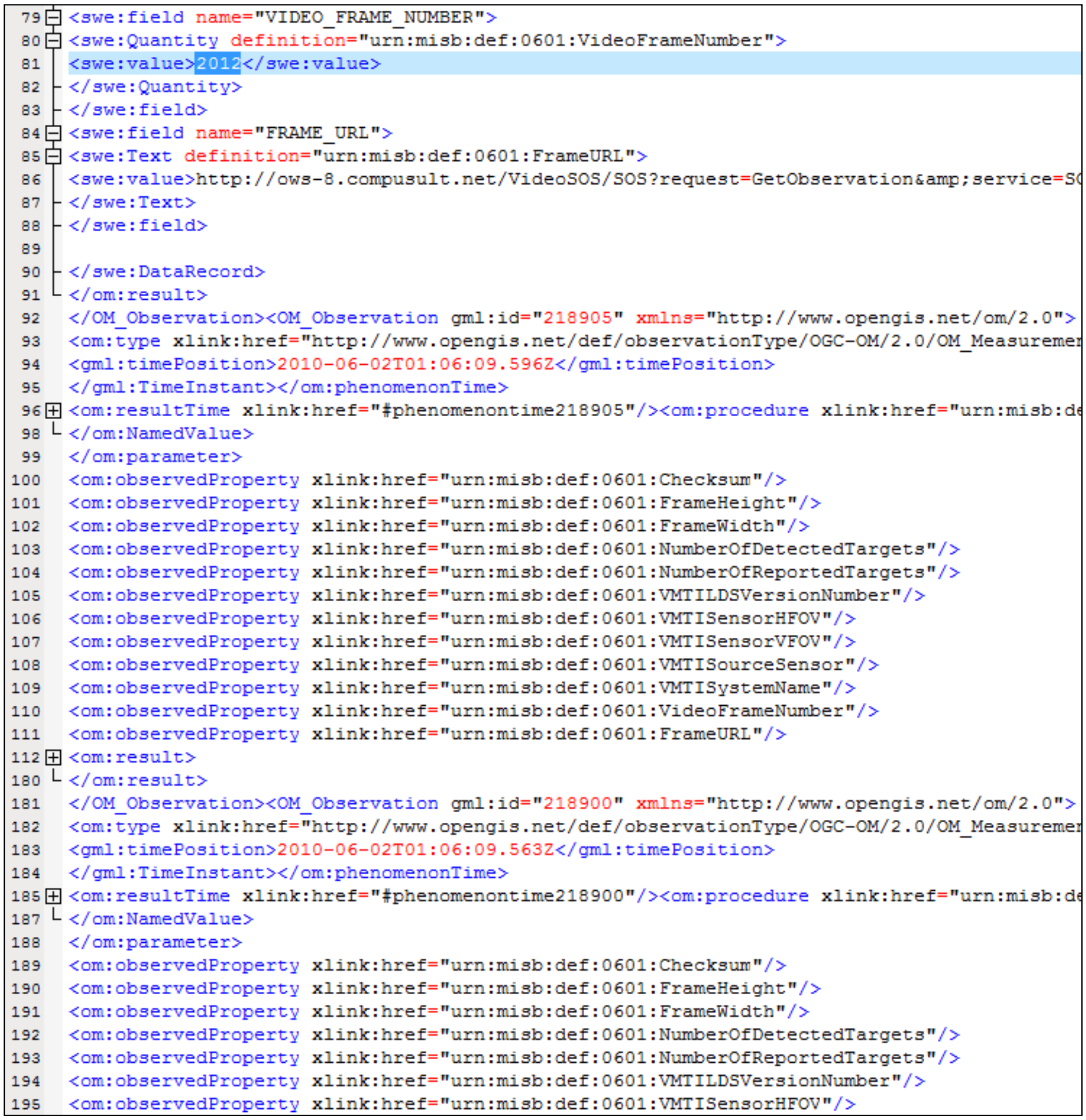
(1) VMTI dataset with complete camera position should be provided for testing.
(2) OGC information model for modeling the MISB 0601 should have these metadata encoded and made accessible through SOS.

7.3.4 Cross-referencing for Performance – Incremental Processing

Problems:

Incremental processing should be enabled with the Detections-to-Tracks WPS. Incremental processing should allow the instant processing of long video streams without delays or waiting until the whole video is available. The current implementation of the tracking WPS does not enable incremental processing. The reason is that exact cross-referencing of frame, metadata, and video is not possible with the current implementation of OGC Information Model in the SOS. The SOS only provides the frames and their metadata when such frames have metadata information. The SOS omits those frames that have no metadata. For example, Figure 13 shows the frame #2007 metadata retrievable from the SOS and Figure 12 shows the frame #2012 retrievable from the SOS. All frames between 2007 and 2012 do not have metadata and these frames cannot be retrieved from the SOS.

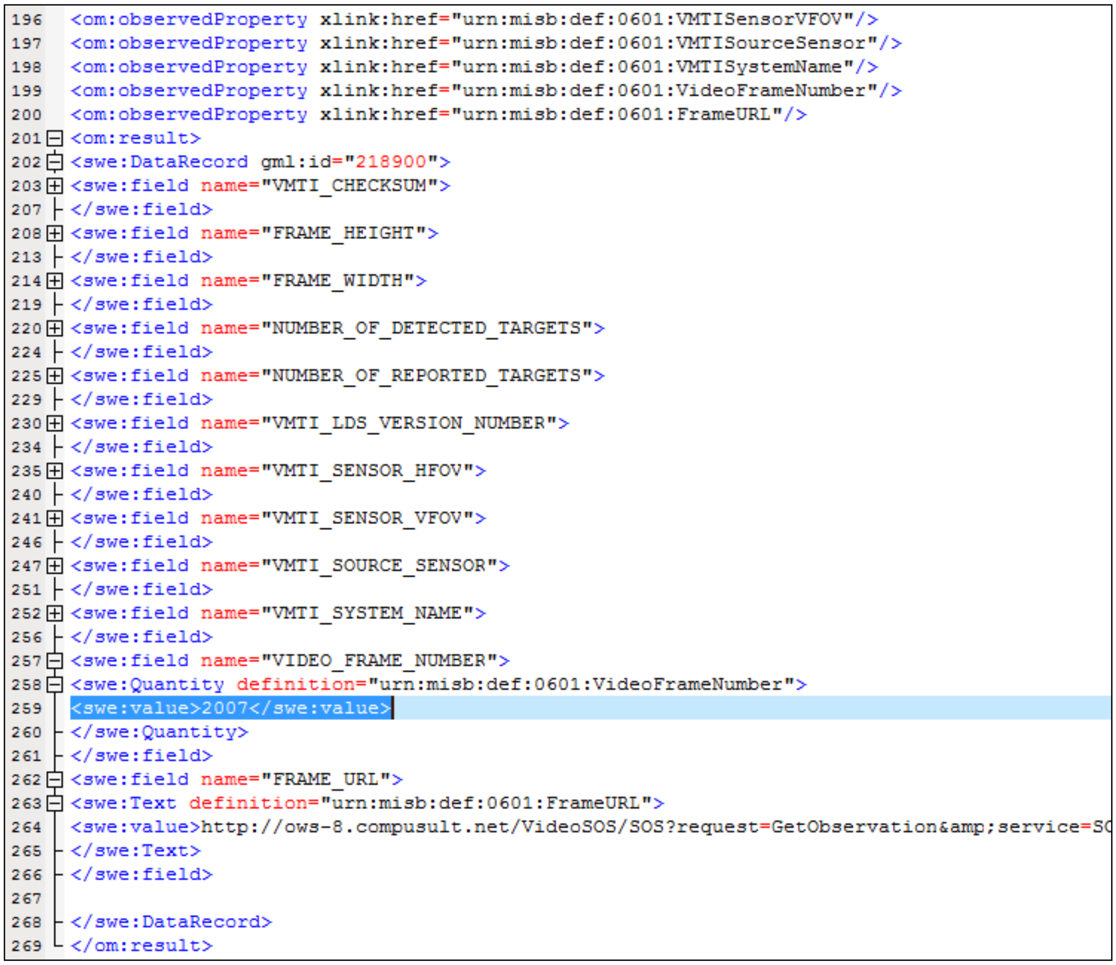
The GMU tracking WPS use KLT as the backbone algorithm that assumes there is little change across frames. Adjacent frames have to be used to keep tracking the target continuously. Ideally, the algorithm needs to retrieve the frame and its corresponding frame metadata (or VMTI metadata). If there is no metadata, the frames should still be retrievable. The current implementation of OGC Information Model modeling the VMTI data does not support such cross-referencing. Two issues can be seen from the example: (1) frames without metadata are not retrievable from the SOS; and (2) frame number is referred to the original video in completeness that has no correspondence to the frame retrieved from the SOS unless the original video is retrieved and used as a whole.



```
<swe:field name="VIDEO_FRAME_NUMBER">
<swe:Quantity definition="urn:misb:def:0601:VideoFrameNumber">
<swe:value>2012</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="FRAME_URL">
<swe:Text definition="urn:misb:def:0601:FrameURL">
<swe:value>http://ows-8.compusult.net/VideoSOS/SOS?request=GetObservation&amp;service=S
</swe:Text>
</swe:field>

</swe:DataRecord>
</om:result>
</OM_Observation><OM_Observation gml:id="218905" xmlns="http://www.opengis.net/om/2.0">
<om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measuremen
<gml:timePosition>2010-06-02T01:06:09.596Z</gml:timePosition>
</gml:TimeInstant></om:phenomenonTime>
<om:resultTime xlink:href="#phenomenontime218905"/><om:procedure xlink:href="urn:misb:de
</om:NamedValue>
</om:parameter>
<om:observedProperty xlink:href="urn:misb:def:0601:Checksum"/>
<om:observedProperty xlink:href="urn:misb:def:0601:FrameHeight"/>
<om:observedProperty xlink:href="urn:misb:def:0601:FrameWidth"/>
<om:observedProperty xlink:href="urn:misb:def:0601:NumberOfDetectedTargets"/>
<om:observedProperty xlink:href="urn:misb:def:0601:NumberOfReportedTargets"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTILDSVersionNumber"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISensorHFOV"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISensorVFOV"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISourceSensor"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISystemName"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VideoFrameNumber"/>
<om:observedProperty xlink:href="urn:misb:def:0601:FrameURL"/>
<om:result>
</om:result>
</OM_Observation><OM_Observation gml:id="218900" xmlns="http://www.opengis.net/om/2.0">
<om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measuremen
<gml:timePosition>2010-06-02T01:06:09.563Z</gml:timePosition>
</gml:TimeInstant></om:phenomenonTime>
<om:resultTime xlink:href="#phenomenontime218900"/><om:procedure xlink:href="urn:misb:de
</om:NamedValue>
</om:parameter>
<om:observedProperty xlink:href="urn:misb:def:0601:Checksum"/>
<om:observedProperty xlink:href="urn:misb:def:0601:FrameHeight"/>
<om:observedProperty xlink:href="urn:misb:def:0601:FrameWidth"/>
<om:observedProperty xlink:href="urn:misb:def:0601:NumberOfDetectedTargets"/>
<om:observedProperty xlink:href="urn:misb:def:0601:NumberOfReportedTargets"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTILDSVersionNumber"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISensorHFOV"/>
```

Figure 12. Frame 2012 retrieved from the SOS



```
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISensorVFOV"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISourceSensor"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VMTISystemName"/>
<om:observedProperty xlink:href="urn:misb:def:0601:VideoFrameNumber"/>
<om:observedProperty xlink:href="urn:misb:def:0601:FrameURL"/>
<om:result>
<swe:DataRecord gml:id="218900">
<swe:field name="VMTI_CHECKSUM">
</swe:field>
<swe:field name="FRAME_HEIGHT">
</swe:field>
<swe:field name="FRAME_WIDTH">
</swe:field>
<swe:field name="NUMBER_OF_DETECTED_TARGETS">
</swe:field>
<swe:field name="NUMBER_OF_REPORTED_TARGETS">
</swe:field>
<swe:field name="VMTI_LDS_VERSION_NUMBER">
</swe:field>
<swe:field name="VMTI_SENSOR_HFOV">
</swe:field>
<swe:field name="VMTI_SENSOR_VFOV">
</swe:field>
<swe:field name="VMTI_SOURCE_SENSOR">
</swe:field>
<swe:field name="VMTI_SYSTEM_NAME">
</swe:field>
<swe:field name="VIDEO_FRAME_NUMBER">
<swe:Quantity definition="urn:misb:def:0601:VideoFrameNumber">
<swe:value>2007</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="FRAME_URL">
<swe:Text definition="urn:misb:def:0601:FrameURL">
<swe:value>http://ows-8.compusult.net/VideoSOS/SOS?request=GetObservation&amp;service=S
</swe:Text>
</swe:field>

</swe:DataRecord>
</om:result>
```

Figure 13. Frame 2007 retrieved from the SOS

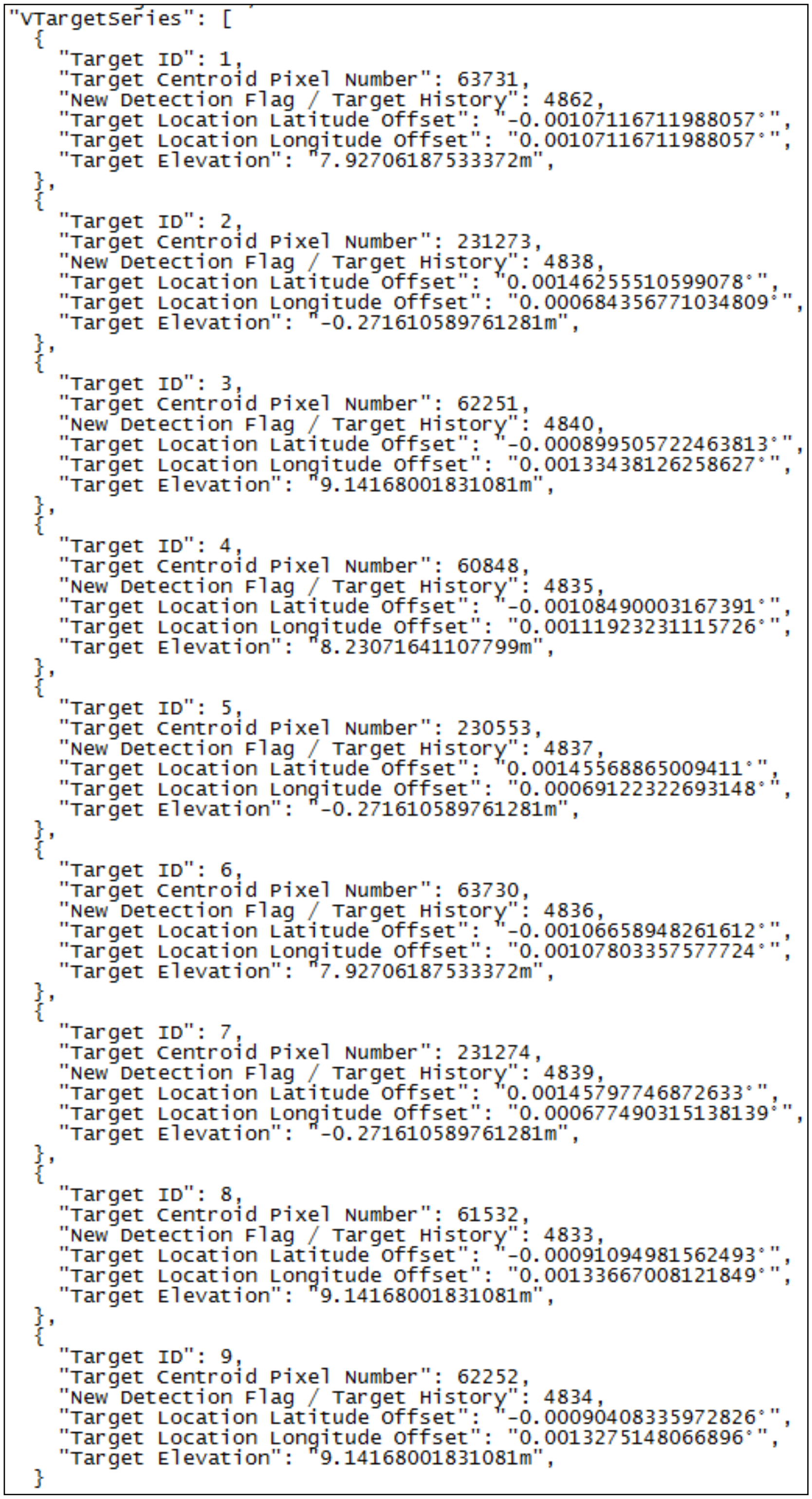
Recommendations:

Cross-referencing of frame, frame metadata, and video segments should be enabled through the OGC Information model for VMTI data. For example, an updated frame number may be assigned. The number should be relevant to the video segment to be retrieved, if the same period is defined to retrieve frames, metadata, and video segment.

7.3.5 Accurate Target Detection for Performance – One Target One Detection

Problems:

Multiple detections are assigned for one single target in one frame. Figure 14 shows one example. Target 1 and Target 6 is so close to each other, only one pixel displacement, to be possibly recognized as two different targets. They are actually one target with different target ID. These situations create confusion for the tracking algorithm to determine which one should be used in the track. In the implementation of GMU Detection-to-Track WPS, a pre-processing to combine targets was adopted. this figure target 1 and target 6 are the same target.



```
"VTargetSeries": [
  {
    "Target ID": 1,
    "Target Centroid Pixel Number": 63731,
    "New Detection Flag / Target History": 4862,
    "Target Location Latitude Offset": "-0.00107116711988057°",
    "Target Location Longitude Offset": "0.00107116711988057°",
    "Target Elevation": "7.92706187533372m",
  },
  {
    "Target ID": 2,
    "Target Centroid Pixel Number": 231273,
    "New Detection Flag / Target History": 4838,
    "Target Location Latitude Offset": "0.00146255510599078°",
    "Target Location Longitude Offset": "0.000684356771034809°",
    "Target Elevation": "-0.271610589761281m",
  },
  {
    "Target ID": 3,
    "Target Centroid Pixel Number": 62251,
    "New Detection Flag / Target History": 4840,
    "Target Location Latitude Offset": "-0.000899505722463813°",
    "Target Location Longitude Offset": "0.00133438126258627°",
    "Target Elevation": "9.14168001831081m",
  },
  {
    "Target ID": 4,
    "Target Centroid Pixel Number": 60848,
    "New Detection Flag / Target History": 4835,
    "Target Location Latitude Offset": "-0.00108490003167391°",
    "Target Location Longitude Offset": "0.00111923231115726°",
    "Target Elevation": "8.23071641107799m",
  },
  {
    "Target ID": 5,
    "Target Centroid Pixel Number": 230553,
    "New Detection Flag / Target History": 4837,
    "Target Location Latitude Offset": "0.00145568865009411°",
    "Target Location Longitude Offset": "0.00069122322693148°",
    "Target Elevation": "-0.271610589761281m",
  },
  {
    "Target ID": 6,
    "Target Centroid Pixel Number": 63730,
    "New Detection Flag / Target History": 4836,
    "Target Location Latitude Offset": "-0.00106658948261612°",
    "Target Location Longitude Offset": "0.00107803357577724°",
    "Target Elevation": "7.92706187533372m",
  },
  {
    "Target ID": 7,
    "Target Centroid Pixel Number": 231274,
    "New Detection Flag / Target History": 4839,
    "Target Location Latitude Offset": "0.00145797746872633°",
    "Target Location Longitude Offset": "0.000677490315138139°",
    "Target Elevation": "-0.271610589761281m",
  },
  {
    "Target ID": 8,
    "Target Centroid Pixel Number": 61532,
    "New Detection Flag / Target History": 4833,
    "Target Location Latitude Offset": "-0.00091094981562493°",
    "Target Location Longitude Offset": "0.00133667008121849°",
    "Target Elevation": "9.14168001831081m",
  },
  {
    "Target ID": 9,
    "Target Centroid Pixel Number": 62252,
    "New Detection Flag / Target History": 4834,
    "Target Location Latitude Offset": "-0.00090408335972826°",
    "Target Location Longitude Offset": "0.0013275148066896°",
    "Target Elevation": "9.14168001831081m",
  }
```

Figure 14. Multiple detections for single target in one frame

Recommendations:

The detection algorithm should generate the VMTI detections with the constraints that one target should have only one detection.

7.3.6 Multiple Target Detections for Performance – Moving Targets in Different Directions

Problems:

The example dataset has another special case that creates problems in tracking objects when two moving targets meet. The case is shown in Figure 15, Figure 16, and Figure 17. In Figure 15, two vehicles are tracked, labeled as vehicle A and B. They have detections described in VMTI. When the two vehicles are crossing each other, some frames have metadata for one vehicle but not both. In the frame shown in Figure 16, vehicle B has metadata but vehicle A does not. In the frame shown in Figure 17Figure 16, vehicle A has metadata but vehicle B does not. This inconsistency creates confusion for the tracking algorithm because both vehicles are so close in these frames. Both detections fall within the threshold of being assigned to either track. The tracking algorithm can easily assign the detection to wrong track.

If both detections are labeled, this problem can be eliminated in combination with the directional arcing adopted in the GMU implementation of Detection-to-Track WPS. The directional arcing is simply a forward looking of frames ahead before finally deciding which track should be associated to the detections.

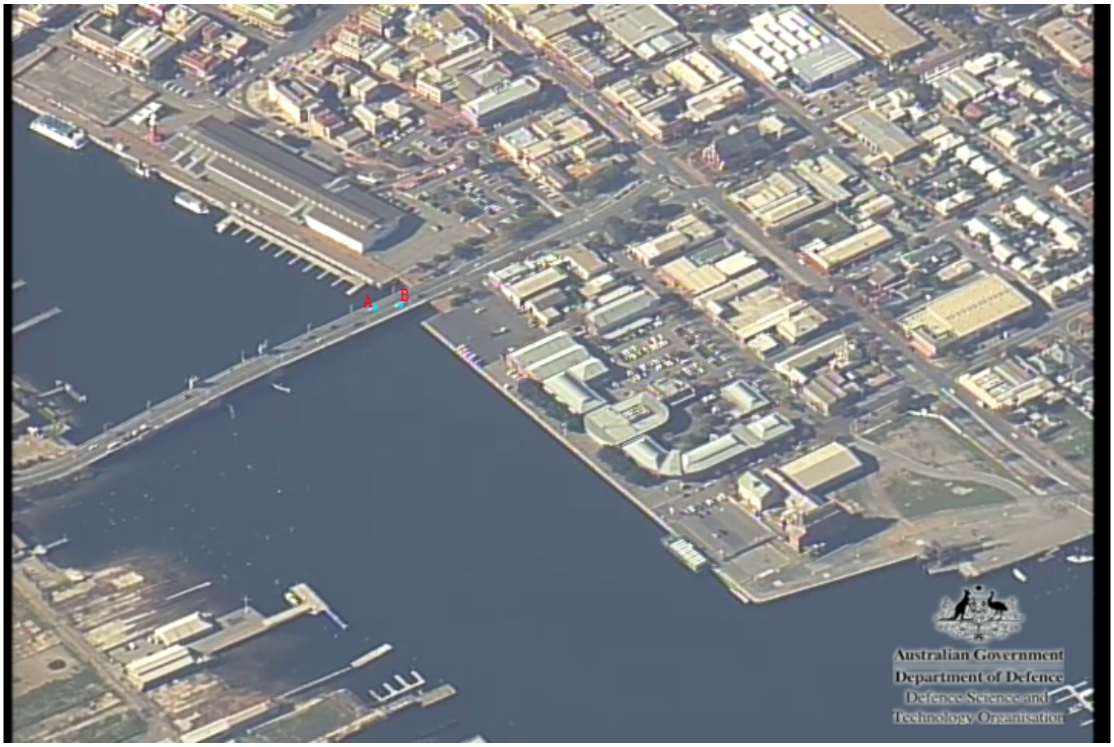


Figure 15. Two vehicles under tracking

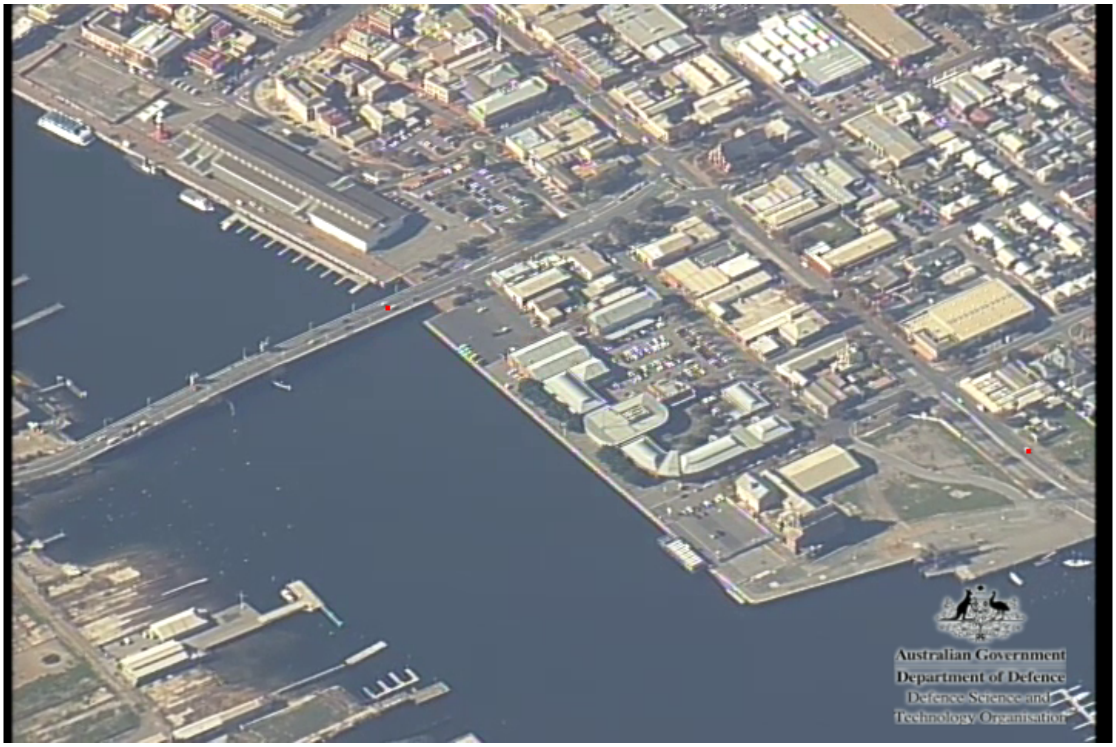


Figure 16. One vehicle is detected and tagged in one adjacent frame although both vehicles present

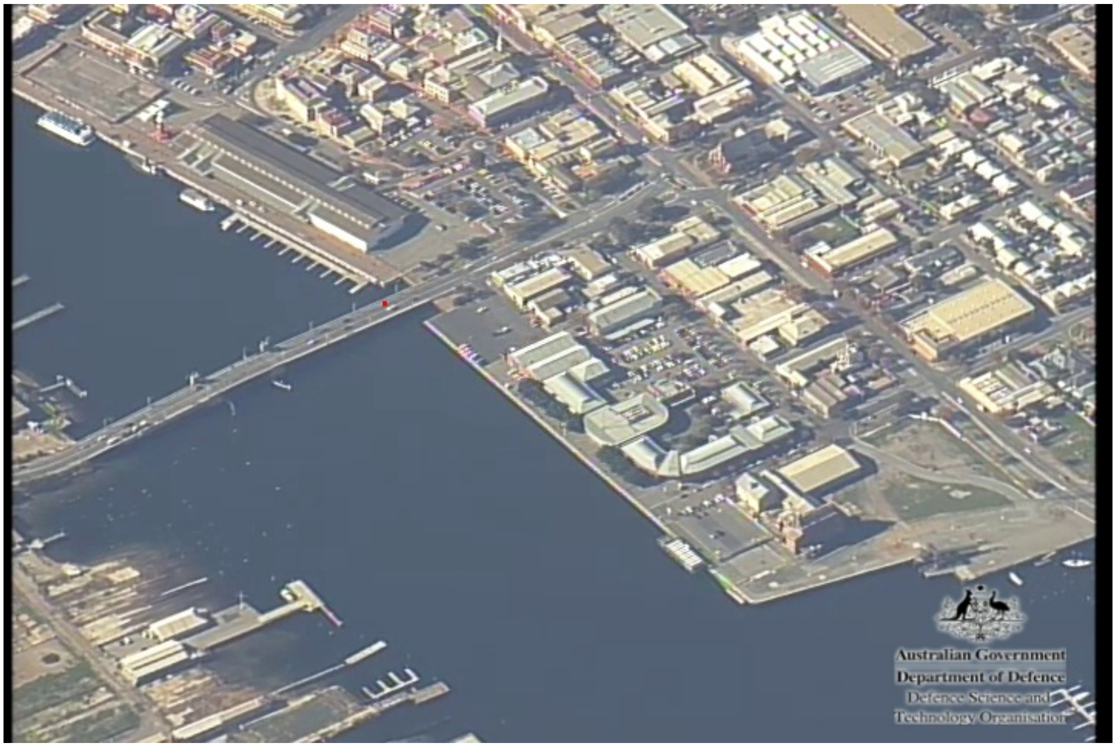


Figure 17. Another vehicle is detected and tagged in another adjacent frame although both vehicles present

Recommendations:

The VMTI dataset should label and provide all detections if multiple targets under tracking occurred in the same frame, Especially when targets occur in the same frame where targets are close together and move in different directions.

8 Web Feature Service implementation

NATO proposed a standard for transmitting and sharing track information; this standard is known as STANAG 4676. It defines a track as a set of associated state vector estimates describing the time-sequenced location, velocity, acceleration, and other relevant characteristics of a mobile object or group of objects

For OWS-8, the tracking thread required a way to store and retrieve information modeled after the STANAG 4676 standard. We wanted the ability to store and retrieve both the current location of a detection, as well as the full path (track) of the same object. Compusult implemented a Web Feature Service (WFS) that partially modeled STANAG 4676 data as GML features.

8.1 Feature Type Implementations in the WFS

The STANAG 4676 UML model developed for OWS-8 (Simonis, 2011) features Tracks as a complex type, consisting of one or more TrackItem elements. A TrackItem could then be either a TrackInformation element or a TrackPoint element, each of which had their own complex types as sub-elements. In order to maintain a manageable feature type in the WFS, the data model was flattened as much as possible, leaving just track information and track points, which were associated back to a parent track. Fields that were not relevant to the scenario were left out of the feature type definition, to keep the schema and generated XML at a smaller size and avoid a lot of empty fields that we did not have sample data for.

Figure 18 -- STANAG 4676 Data Model

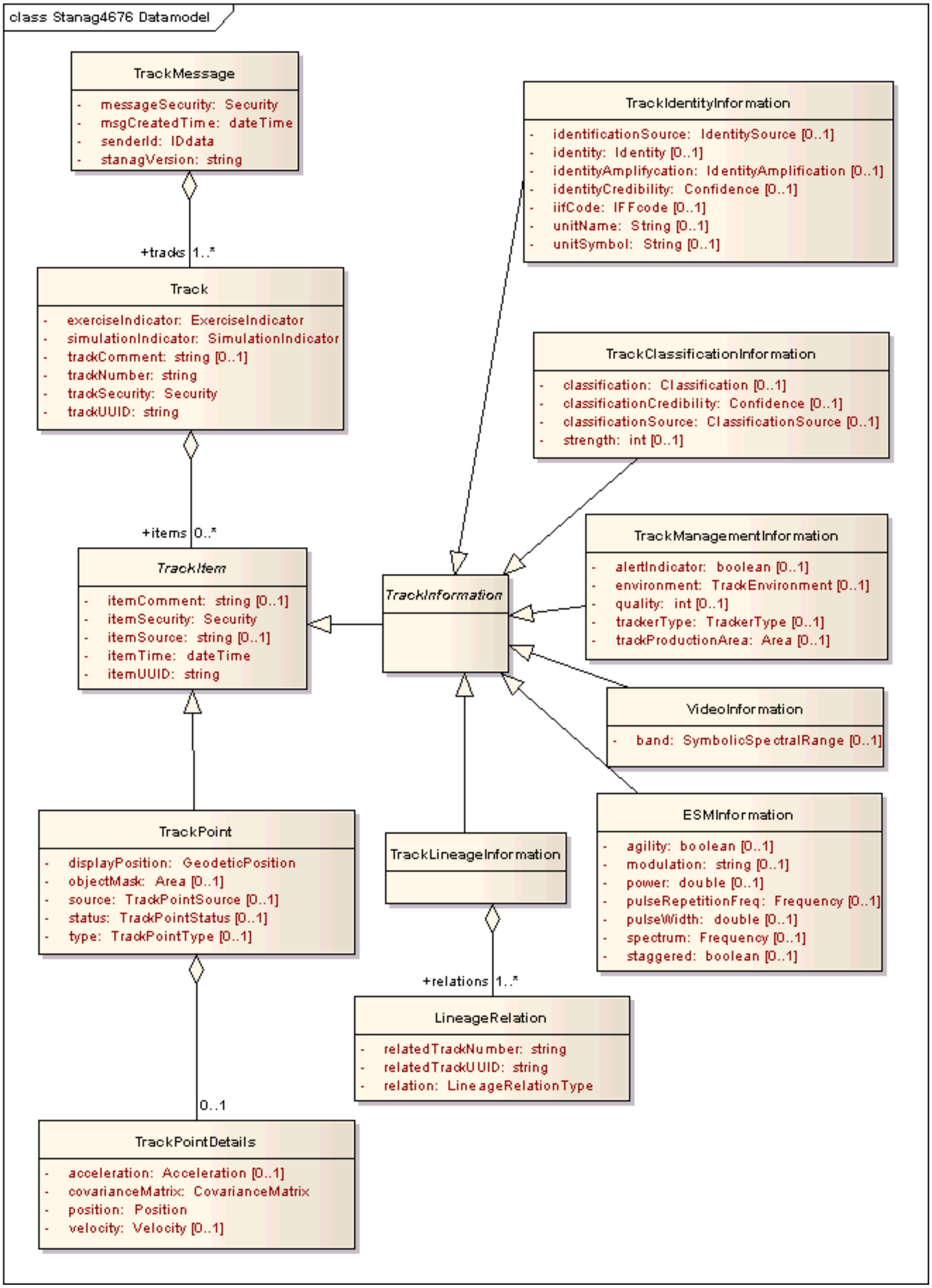
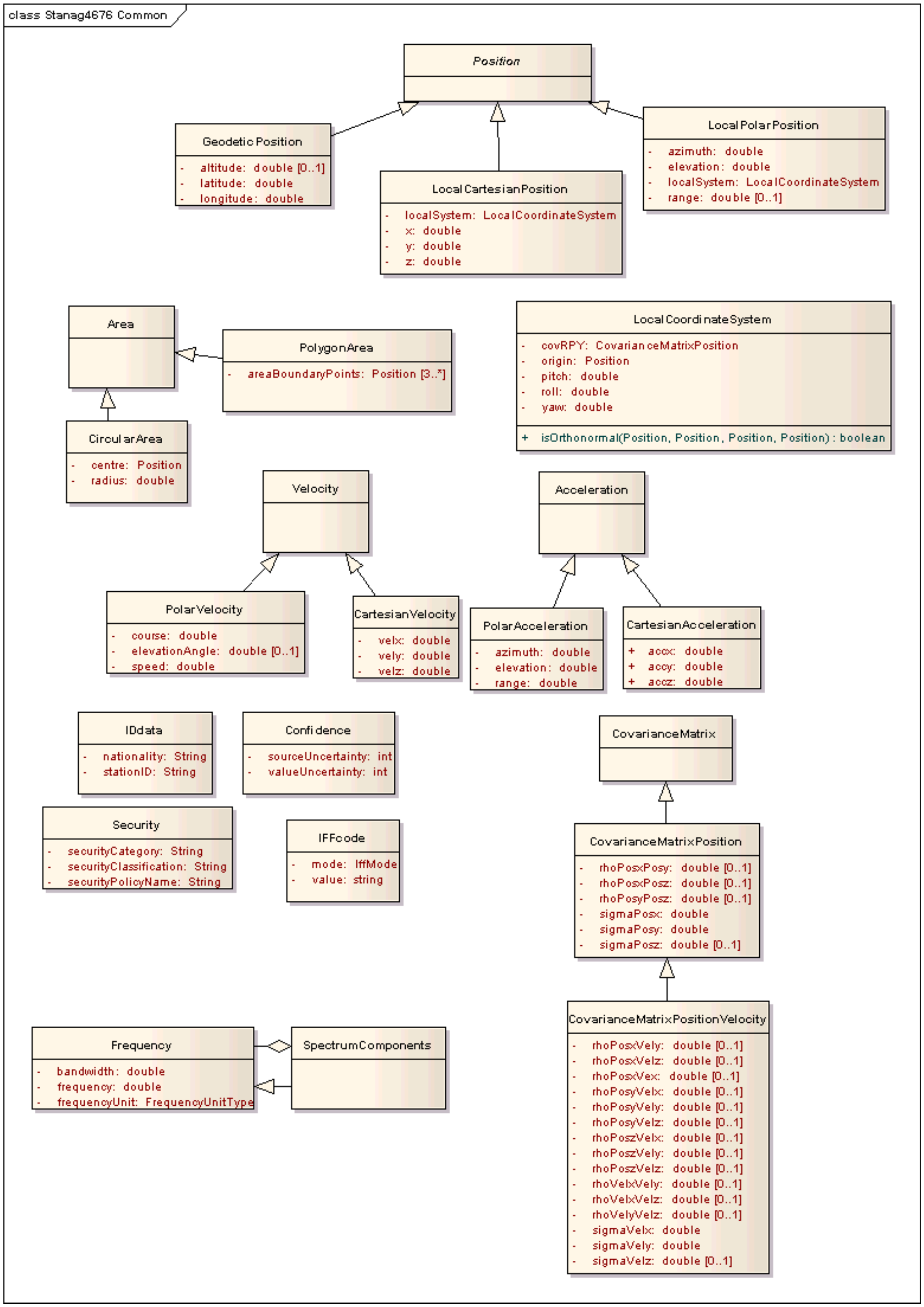

class Stanag4676 Datamodel
TrackMessage
- messageSecurity: Security
- msgCreatedTime: dateTime
- senderId: IDdata
- stanagVersion: string
+tracks 1..*
Track
- exerciseIndicator: ExerciseIndicator
- simulationIndicator: SimulationIndicator
- trackComment: string [0..1]
- trackNumber: string
- trackSecurity: Security
- trackUUID: string
+items 0..*
TrackItem
- itemComment: string [0..1]
- itemSecurity: Security
- itemSource: string [0..1]
- itemTime: dateTime
- itemUUID: string
TrackInformation
TrackIdentityInformation
- identificationSource: IdentitySource [0..1]
- identity: Identity [0..1]
- identityAmplifycation: IdentityAmplification [0..1]
- identityCredibility: Confidence [0..1]
- iifCode: IFFcode [0..1]
- unitName: String [0..1]
- unitSymbol: String [0..1]
TrackClassificationInformation
- classification: Classification [0..1]
- classificationCredibility: Confidence [0..1]
- classificationSource: ClassificationSource [0..1]
- strength: int [0..1]
TrackManagementInformation
- alertIndicator: boolean [0..1]
- environment: TrackEnvironment [0..1]
- quality: int [0..1]
- trackerType: TrackerType [0..1]
- trackProductionArea: Area [0..1]
VideoInformation
- band: SymbolicSpectralRange [0..1]
ESMInformation
- agility: boolean [0..1]
- modulation: string [0..1]
- power: double [0..1]
- pulseRepetitionFreq: Frequency [0..1]
- pulseWidth: double [0..1]
- spectrum: Frequency [0..1]
- staggered: boolean [0..1]
TrackPoint
- displayPosition: GeodeticPosition
- objectMask: Area [0..1]
- source: TrackPointSource [0..1]
- status: TrackPointStatus [0..1]
- type: TrackPointType [0..1]
TrackLineageInformation
+relations 1..*
LineageRelation
- relatedTrackNumber: string
- relatedTrackUUID: string
- relation: LineageRelationType
0..1
TrackPointDetails
- acceleration: Acceleration [0..1]
- covarianceMatrix: CovarianceMatrix
- position: Position
- velocity: Velocity [0..1]

Figure 19 -- STANAG Data Model Cont.



The following table lists the fields available in STANAG 4676, and identifies which ones were included:

Element	Field name	Included (y/n)
TrackMessage	messageSecurity	Y
TrackMessage	msgCreatedTime	N
TrackMessage	senderId	N
TrackMessage	stanagVersion	N
Track	exerciseIndicator	Y
Track	simulationIndicator	Y
Track	trackComment	Y
Track	trackNumber	Y
Track	trackSecurity	Y
Track	trackUUID	Y
TrackItem	itemComment	Y
TrackItem	itemSecurity	Y
TrackItem	itemSource	Y
TrackItem	itemTime	Y
TrackItem	itemUUID	Y
TrackPoint	displayPosition	Y
TrackPoint	objectMask	Y
TrackPoint	Source	Y
TrackPoint	Status	Y
TrackPoint	Type	Y
TrackPointDetails	Acceleration	Y

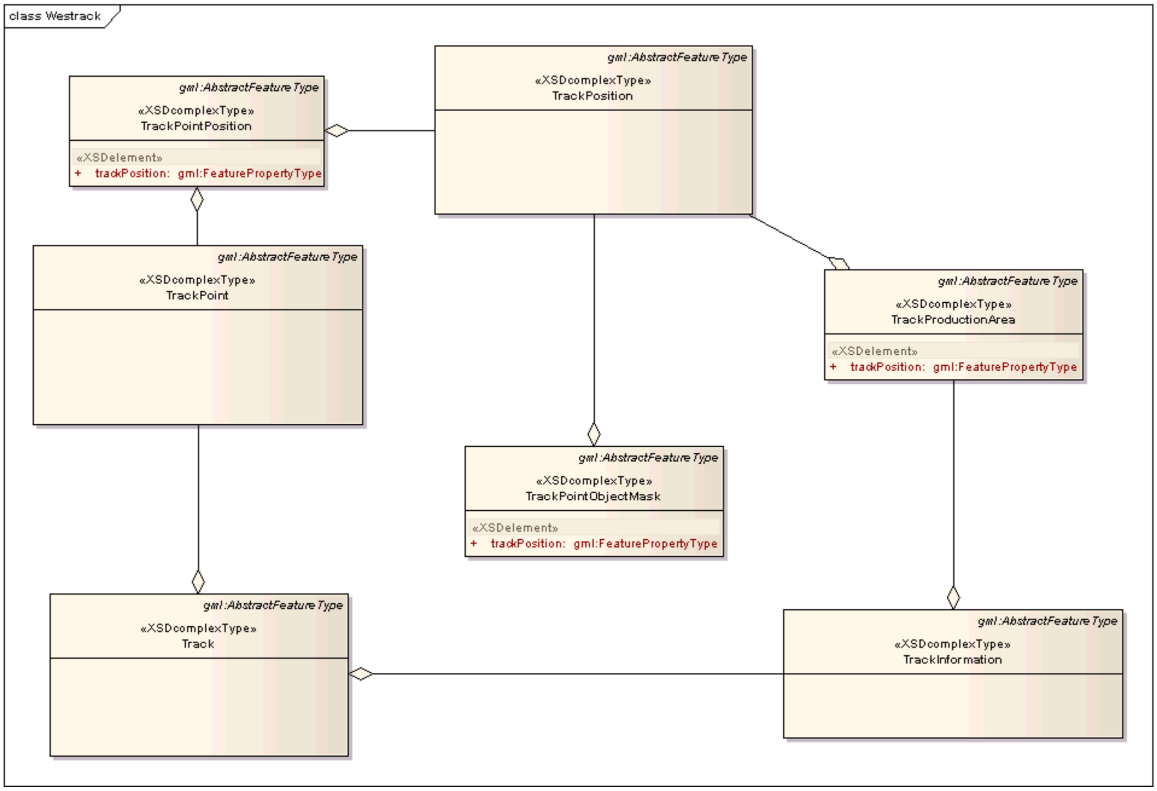
TrackPointDetails	covarianceMatrix	N
TrackPointDetails	Position	Y
TrackPointDetails	Velocity	Y
Element	**Field name**	**Included (y/n)**
TrackIdentityInformation	identificationSource	N
TrackIdentityInformation	Identity	N
TrackIdentityInformation	identityAmplification	N
TrackIdentityInformation	identityCredibilty	N
TrackIdentityInformation	iif_code	N
TrackIdentityInformation	unitName	N
TrackIdentityInformation	unitSymbol	N
TrackClassificationInformation	Classification	Y
TrackClassificationInformation	classificationCredibility	N
TrackClassificationInformation	classificationSource	Y
TrackClassificationInformation	Strength	N
TrackManagementInformation	alertIndicator	N
TrackManagementInformation	Environment	N
TrackManagementInformation	Quality	N
TrackManagementInformation	tracker_type	N
TrackManagementInformation	trackProductionArea	Y
VideoInformation	Band	N
ESMInformation	Agilty	N
ESMInformation	Modulation	N
ESMInformation	Power	N
ESMInformation	pulseRepetitionFreq	N
ESMInformation	pulseWidth	N

ESMInformation	Spectrum	N
ESMInformation	Staggered	N
LineageRelation	relatedTrackNumber	N
LineageRelation	relatedTrackUUID	N
Element	**Field name**	**Included (y/n)**
LineageRelation	Relation	N

Table 3 -- STANAG 4676 Fields

A problem arose in attempting to model the geospatial information associated with a track point. There was a recursive problem with Position types, as a Position could be one of GeodeticPosition, LocalCartesianPosition, or LocalPolarPosition. However, both the LocalCartesianPosition and LocalPolarPosition types had a LocalCoordinateSystem element, which had an origin element, which was of type Position. Thus it was possible for a Position to have an infinite number of sub positions associated with it. To simplify things, we assumed that all Positions were of the non-recursive GeodeticPosition type.

The feature type definition of the Track is provided in Annex B. The following diagram illustrates the model as implemented for OWS-8.



8.1.1 Inserting Track Data

As the tracker was to consume data from the SOS and then insert features into the WFS, it was only necessary for the track to be inserted once, at the point of first detection. From that point on, track points could be inserted, but the parent track was only updated.

8.1.2 Updating Track Data

When additional SOS detections were available to the tracker, new track points were inserted into the WFS. Rather than require the tracker to also update the track with the new geometry information, the WFS service automatically updated with the latest geometry coming from the inserted track points.

A sample track point insert can be seen in A.7.

9 Notification Service implementation

As there was a need in the Tracking thread to be able to send notification messages regarding what was being tracked, Compusult's WFS service had to be extended to provide "PubSub" (Publish/Subscribe) capabilities. As there was already an OGC PubSub working group, we attempted to follow guidelines that they were putting in place, although they are still working toward a final solution and thus we had to make our own implementation decisions.

9.1 OASIS Web Services Notification (WSN)

The OASIS WSN standard was chosen as the basis for our notification service as it provides a framework for subscriptions and notifications, and was under consideration by the OGC PubSub working group.

WSN defines Subscribe, Renew, and Notify requests and responses, which were used by our notification service and clients.

The WSN framework is flexible, allowing "Any" elements inside the XML requests to allow subscription filters to be based on any XML element. Unfortunately, "Any" element also meant that it was easy to have XML that validates against the WSN schema, but that doesn't match the format that the web service is expecting. As such, communication outside of the schema provided by the web service to the client was required so that clients would be sending data in the expected format.

9.2 SubscriptionManager

Ideally, subscription operations would be added to the existing Web Feature Service (WFS). However, any new operations added to the WFS could not be advertised via a WFS GetCapabilities request because a response containing new operations would invalidate the GetCapabilities response against the WFS schema. This could cause problems for clients that use schema-generated code to parse the XML response.

It was easier to create a separate web service to act as the Subscription Manager on behalf of the Web Feature Service, at least until the PubSub operations are officially added into a future version of WFS.

9.2.1 Subscribe

A Subscribe request requires 3 pieces of information:

1. Where to send notifications (ConsumerReference)
2. What to subscribe to (Filter)
3. How long to subscribe for (InitialTerminationTime)

A sample Subscribe request can be seen here:

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscribe xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://docs.oasis-open.org/wsn/b-2" xmlns:wfs-
topic="http://www.opengis.net/wfs/topic"
    xmlns:wfs="http://www.opengis.net/wfs"
xmlns:app="http://schemas.compusult.net/tracks"
    xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml"
    xsi:schemaLocation="http://docs.oasis-open.org/wsn/b-2
http://docs.oasis-open.org/wsn/b-2.xsd http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
    <ConsumerReference>
        <Address xmlns="http://www.w3.org/2005/08/addressing"
            >http://ows-8.compusult.net/wes/Notification</Address>
    </ConsumerReference>
    <Filter>
        <TopicExpression Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Concrete">
            wfs-topic:Transaction
        </TopicExpression>
        <wfs:Query typeName="app:Track">
            <ogc:Filter>
                <ogc:Intersects>
                    <ogc:PropertyName>app:geom</ogc:PropertyName>
                    <gml:Envelope>
                        <gml:lowerCorner>-60.02340
45.99196</gml:lowerCorner>
                        <gml:upperCorner>-51.47130
52.51542</gml:upperCorner>
                    </gml:Envelope>
                </ogc:Intersects>
            </ogc:Filter>
        </wfs:Query>
    </Filter>
    <InitialTerminationTime>2011-08-20T18:00:00</InitialTerminationTime>
</Subscribe>
```

The Subscription Manager would respond with the following SubscribeResponse which contains a unique Address identifying the subscription at the SubscriptionManager :

```
<?xml version="1.0" encoding="UTF-8"?>
<SubscribeResponse xmlns="http://docs.oasis-open.org/wsn/b-2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://docs.oasis-open.org/wsn/b-2
http://docs.oasis-open.org/wsn/b-2.xsd">
   <SubscriptionReference>
      <Address xmlns="http://www.w3.org/2005/08/addressing">http://ows-
8.compusult.net/wesgfm/SubscriptionManager/subscription/ 8078c0a0-8093-
11e0-b278-0800200c9a66</Address>
   </SubscriptionReference>
   <CurrentTime>2011-08-19T16:21:00</CurrentTime>
   <TerminationTime>2011-08-20T18:00:00</TerminationTime>
</SubscribeResponse>
```

9.2.2 Renew

Prior to a subscription expiring, a client can request that the subscription be extended by sending the following request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Renew xmlns="http://docs.oasis-open.org/wsn/b-2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://docs.oasis-open.org/wsn/b-2
http://docs.oasis-open.org/wsn/b-2.xsd">
   <TerminationTime>2011-08-21T18:00:00</TerminationTime>
   <add:ReferenceParameters
xmlns:add="http://www.w3.org/2005/08/addressing"
xsi:schemaLocation="http://www.w3.org/2005/08/addressing
http://www.w3.org/2006/03/addressing/ws-addr.xsd">
      <add:RelatesTo>urn:uuid:f14c1c39-3e4e-439b-98a0-
f9bfb9467d36</add:RelatesTo>
   </add:ReferenceParameters>
</Renew>
```

The subscription manager will then return the following response:

```
<?xml version="1.0" encoding="UTF-8"?>
<RenewResponse xmlns="http://docs.oasis-open.org/wsn/b-2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://docs.oasis-open.org/wsn/b-2
http://docs.oasis-open.org/wsn/b-2.xsd">
   <TerminationTime>2011-08-21T18:00:00</TerminationTime>
   <CurrentTime>2011-08-20T16:00:00</CurrentTime>
   <add:ReferenceParameters
xmlns:add="http://www.w3.org/2005/08/addressing"
xsi:schemaLocation="http://www.w3.org/2005/08/addressing
http://www.w3.org/2006/03/addressing/ws-addr.xsd">
      <add:RelatesTo> urn:uuid:f14c1c39-3e4e-439b-98a0-
f9bfb9467d36</add:RelatesTo>
   </add:ReferenceParameters>
</RenewResponse>
```

9.2.3 Topics

The scenario called for objects to be tracked within a defined area of interest (AOI). Thus, two types of notification messages were expected: 1) notification that an object had entered or exited the area of interest; and 2) detections of the object as it moves within the area of interest.

Different clients would be interested in one of these two possible types of notification messages. The GeoSMS notification broker would want to be notified of objects that are entering the specified area of interest, and would then generate a GeoSMS message to their subscribed clients with a reference back to the original detection. A map client, on the other hand, would be interested in receiving all object data so that it can draw and track the object on a map as it moves within the area of interest.

To support different types of notifications, we made use of the concept of Topics. Topics were advertised to the clients by placing them inside the ExtendedCapabilities section of the WFS GetCapabilities response.

The following TopicSet was created:

```
<ows:ExtendedCapabilities>
   <wstop:TopicSet xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://docs.oasis-open.org/wsn/t-1">
      <wfs:Match wstop:topic="true">
         <wfs:EnterArea wstop:topic="true" />
         <wfs:ExitArea wstop:topic="true" />
      </wfs:Match>
      <wfs:Transaction wstop:topic="true">
         <wfs:Insert wstop:topic="true" />
         <wfs:Update wstop:topic="true" />
         <wfs:Delete wstop:topic="true" />
      </wfs:Transaction>
   </wstop:TopicSet>
</ows:ExtendedCapabilities>
```

Clients that were interested in receiving “match” notifications (that is, an object matching the area of interest) could subscribe to the “Match” topic, or two either or both of the “EnterArea” and “ExitArea” subtopics.

Similarly, clients that were interested in receiving all data could subscribe to the “Transaction” topic, or to any or all of the “Insert”, “Update”, and “Delete” subtopics.

9.3 Notifications

When a WFS Transaction comes into the WFS, the web service compares the Transaction to the subscriptions that were registered against it.

Notification messages are generated by the WFS for any subscription that the Transaction satisfies.

9.3.1 Match

If the client specified the Match topic, they will receive a Notify message that includes geospatial information as well as the GetFeature request URL.

A sample match notification can be seen here:

```
<?xml version="1.0" encoding="UTF-8"?>
<Notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://docs.oasis-open.org/wsn/b-2
http://docs.oasis-open.org/wsn/b-2.xsd"
    xmlns="http://docs.oasis-open.org/wsn/b-2"
xmlns:app="http://schemas.compusult.net/tracks"
    xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:gml="http://www.opengis.net/gml" xmlns:msgf="http://ows-
8.compusult.net/MessageFormat">
   <NotificationMessage>
       <SubscriptionReference>
          <Address xmlns="http://www.w3.org/2005/08/addressing"
                >http://ows-
8.compusult.net/wesgfm/SubscriptionManager/8078c0a0-8093-11e0-b278-
0800200c9a66</Address>
       </SubscriptionReference>
       <Topic Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Concrete"
            xmlns:wfs-topic="http://www.opengis.net/wfs/topic">
          wfs-topic:Match/wfs-topic:EnterArea
       </Topic>
       <Message>
          <msgf:MessageData>
             <msgf:URIPath>http://ows-
8.compusult.net/wesgfm/services</msgf:URIPath>
             <msgf:Longitude>139.7970</msgf:Longitude>
             <msgf:Latitude>35.6980</msgf:Latitude>
             <msgf:ExtendedParameters>
                 <![CDATA[?SERVICE=WFS&REQUEST=GetFeature&VERSION=1.1.0

&TYPENAME=app:Track&NAMESPACE=xmlns(app=http://schemas.compusult.net/tr
acks&FEATUREID=Tracks_30]]>
             </msgf:ExtendedParameters>
             <msgf:Description>Feature 'Track_36' was
inserted</msgf:Description>
             <msgf:EventTime>2011-08-17T00:00:00</msgf:EventTime>
          </msgf:MessageData>
       </Message>
   </NotificationMessage>
</Notify>
```

In the case of the GeoSMS client, this information is extracted and converted into a GeoSMS message, containing the point information and a reference to the GetFeature

WFS request, which consists of the msgf:URIPath URL appended with the msgf:ExtendedParameters content.

9.3.2 Transaction

If the client specified the Transaction topic, they will receive a Notify message that includes full FeatureCollection information from the WFS.

A sample transaction notification can be seen here illustrating delete and update notifications:

```
<?xml version="1.0" encoding="UTF-8"?>
<Notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://docs.oasis-open.org/wsn/b-2
http://docs.oasis-open.org/wsn/b-2.xsd"
 xmlns="http://docs.oasis-open.org/wsn/b-2"
 xmlns:app="http://schemas.compusult.net/tracks"
 xmlns:wfs="http://www.opengis.net/wfs"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:gml="http://www.opengis.net/gml">
   <NotificationMessage>
       <SubscriptionReference>
          <Address xmlns="http://www.w3.org/2005/08/addressing"
                >http://ows-
8.compusult.net/wesgfm/SubscriptionManager/c14f8b5f1f4b4afbac16b0a3a5e4
03e2</Address>
       </SubscriptionReference>
       <Topic Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Concrete"
            xmlns:wfs-topic="http://www.opengis.net/wfs/topic">
          wfs-topic:Transaction/wfs-topic:Delete
       </Topic>
       <Message>
          <wfs:FeatureCollection numberOfFeatures="1"
                xmlns:app="http://schemas.compusult.net/tracks"
                xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xlink="http://www.w3.org/1999/xlink"
                xmlns:gml="http://www.opengis.net/gml">
             <gml:featureMember>
                 <app:Track gml:id="Tracks_34"/>
             </gml:featureMember>
          </wfs:FeatureCollection>
       </Message>
   </NotificationMessage>
   <NotificationMessage>
       <SubscriptionReference>
          <Address xmlns="http://www.w3.org/2005/08/addressing"
                >http://ows-8-
devel.compusult.net/wesgfm/SubscriptionManager/subscription/f4484c7d-
56ad-4e73-9754-4377b801d8fd</Address>
       </SubscriptionReference>
       <Topic Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Concrete"
            xmlns:wfs-topic="http://www.opengis.net/wfs/topic">
          wfs-topic:Transaction/wfs-topic:Update
      </Topic>
      <Message>
          <wfs:FeatureCollection numberOfFeatures="1"
                xmlns:app="http://schemas.compusult.net/tracks"
                xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xlink="http://www.w3.org/1999/xlink"
                xmlns:gml="http://www.opengis.net/gml">

<!—content removed for brevity --->

          </wfs:FeatureCollection>
      </Message>
   </NotificationMessage>
</Notify>
```

A map client can translate this FeatureCollection information into data that can be rendered as points and tracks on a map.

10 Client Implementation

10.1 Client Data Flows

For OWS-8, the scenario revolved around the concepts of subscription to track information. The analyst user would subscribe for notifications of new track data. When these notifications were received, the analyst would be able to access the track data and visualize the data on the map. Once the track data was accessible, the analyst could use metadata to retrieve the imagery evidence provided in the track data and display track information directly on the imagery. This data flow was proven in concept. The following table provides the use case used for the OWS-8 demo.

Table 4 -- Use case for tracking and notification

Title	Analyst Use Case for viewing track information
Description	This use case illustrates the subscription for track data, display of track data in a mapping client, accessing the track metadata and retrieval of imagery evidence to illustrate and confirm the track information.
Actors:	ISR analyst, UAV controller
Trigger:	Earthquake occurs and tracking data is required for a bridge area to confirm or deny traffic flow.
Preconditions:	□ WPS and client have access to WFS and SOS □ WFS is available that has STANAG 4676 Track data model □ SOS is available that can support storage and retrieval

	of VMTI frame and detection information □ WPS is available to analyze data for tracks □ Notifications can be sent from WFS to client
Post conditions:	□ Track data is displayed to the user □ Track evidence is retrieved from the SOS and is displayed to the user
Basic Flow (Steps):	1. Analyst logs in to portal 2. Analyst requests UAV to survey area. 3. UAV is tasked to survey area. 4. UAV film is uploaded to SOS 5. WPS is invoked to create track data 6. Analyst discovers Track WFS using catalog. 7. Analyst subscribes to Track WFS by drawing area of interest on map. 8. WPS inserts discovered tracks into WFS 9. Client receives notification from WFS. 10. Client retrieves track data from WFS. 11. Client displays track data on the map. 12. Analyst invokes dialog to display track metadata for selected track. 13. Client displays imagery evidence retrieved from SOS. 14. Imagery evidence shows tracked object which is highlighted using metadata retrieved from SOS.
Notes	None

Figure 20 -- Client use of services illustrates how the client software uses the WFS and the SOS to visualize tracking data. The user is provided with a dialog that displays track data from a WFS track point feature. The track point contains relevant metadata that can be used by an analyst. Additionally, the feature contains a link to the video evidence for that derived track point. It is important to note that the point has been developed from the detection and video information and will not have a direct correspondence to the detections available from the SOS.

The evidence does contain a link to the frame image that corresponds to the point in time for the track point. Using this time stamp information and the frame information itself, the client retrieves frame metadata such as footprint quadrilateral, frame height and width. These are used with track point geographic position to plot the track point on the frame image itself.

Because all track points are available from the WFS, the analyst can flip back and forth between frames and see the plotting of track points over frame imagery to determine the accuracy of the track and the visual properties of the tracked feature.

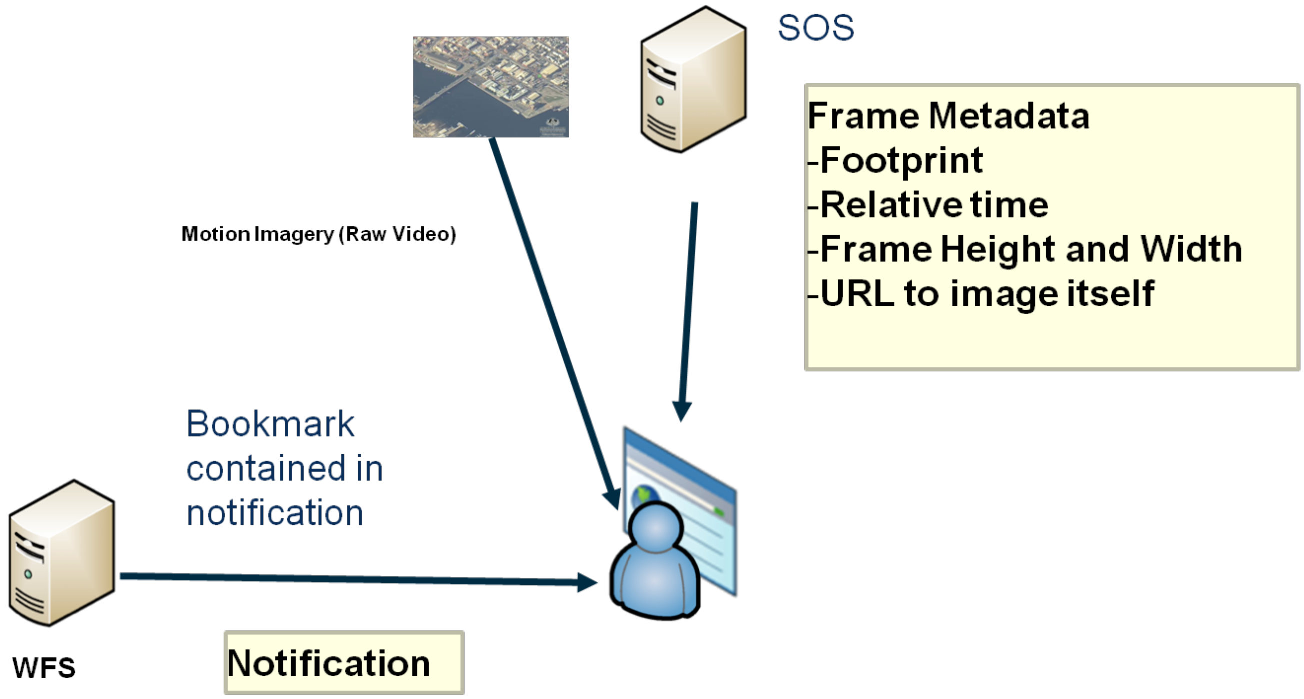


Figure 20 -- Client use of services

Figure 21 -- Track Display Using a WMS provides a screenshot of the analyst interface showing a track developing along a road. The dot indicates the latest known track point and the brown line indicates the known geometry of the track. Clicking on the track will invoke the dialog illustrated in **Figure 22 -- Track Point Interface.** In that figure, we can see frame imagery, metadata for the latest point and Next and Previous buttons that allow the Analyst to view track points, their metadata and associated frame imagery. The track point itself is plotted on the frame imagery. This is illustrated **in Figure 23 -- Frame Image with highlighted Detection**.

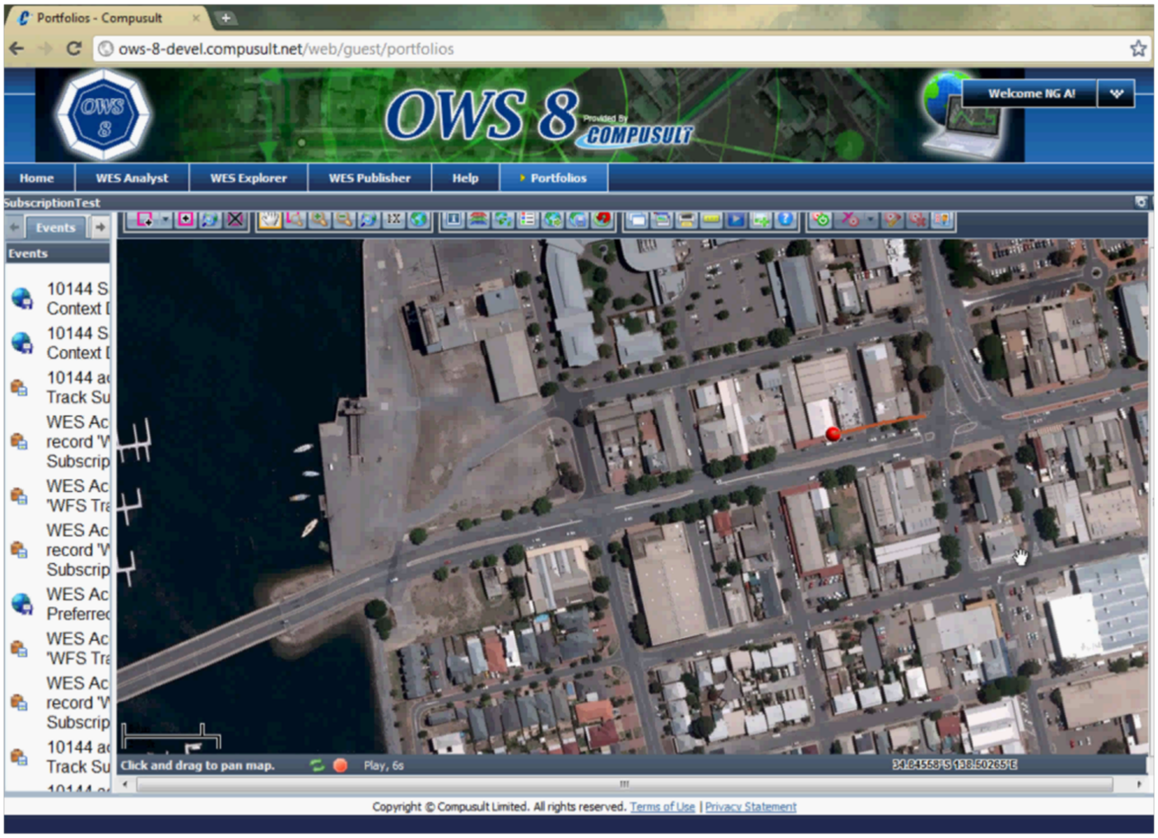
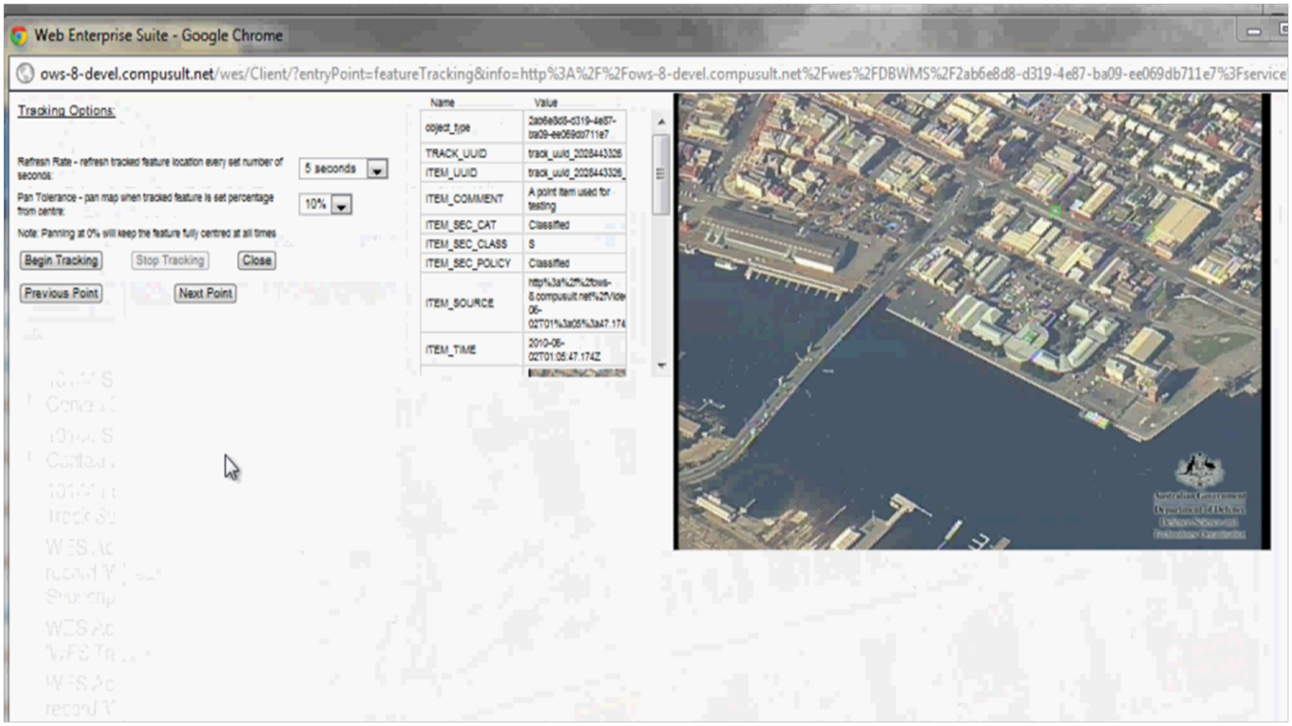


Figure 21 -- Track Display Using a WMS



ows-8-devel.compusult.net/wes/Client/?entryPoint=featureTracking&info=http%3A%2F%2Fows-8-devel.compusult.net%2Fwes%2FDBWMS%2F2ab6e8d8-d319-4e87-ba09-ee069db711e7%3Fservice

Tracking Options:

Refresh Rate - refresh tracked feature location every set number of seconds: 5 seconds

Pan Tolerance - pan map when tracked feature is set percentage from centre: 10%

Note: Panning at 0% will keep the feature fully centred at all times

Name	Value
object_type	2ab6e8d8-d319-4e87-ba09-ee069db711e7
TRACK_UUID	track_uuid_2028443326
ITEM_UUID	track_uuid_2028443326_
ITEM_COMMENT	A point item used for testing
ITEM_SEC_CAT	Classified
ITEM_SEC_CLASS	S
ITEM_SEC_POLICY	Classified
ITEM_SOURCE	http%3a%2f%2fows-8.compusult.net%2fVide 06- 02T01%3a05%3a47.174
ITEM_TIME	2010-06-02T01:05:47.174Z

Figure 22 -- Track Point Interface

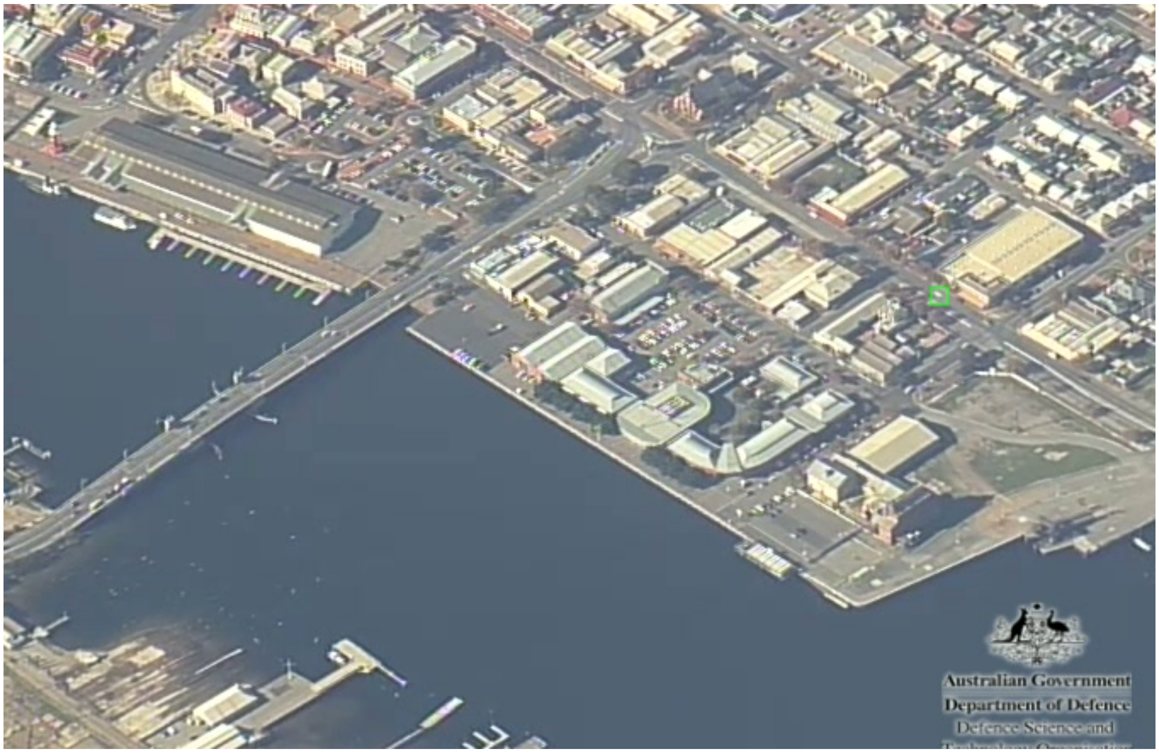
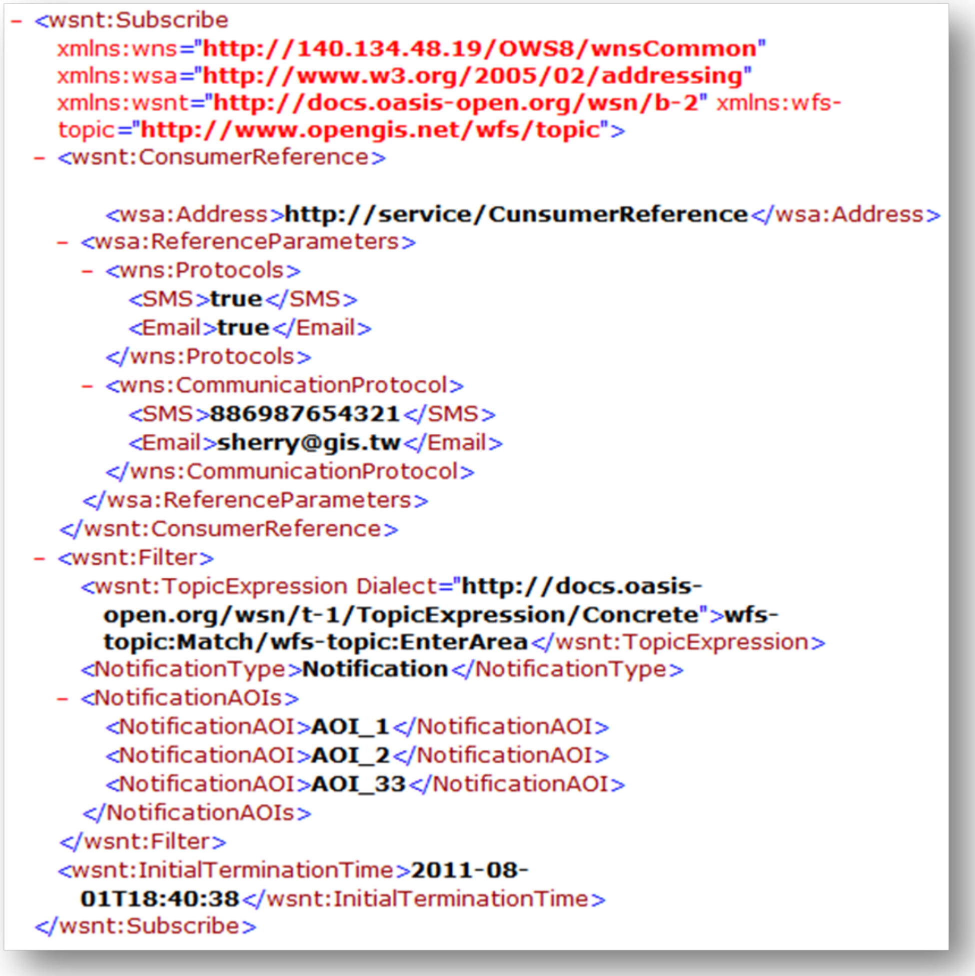


Figure 23 -- Frame Image with highlighted Detection

10.2 GeoSMS notification

As part of OWS-8, notification was supported using a brokering system that would relay WFS notifications to mobile clients as GeoSMS messages. These messages would contain enough information to orient the receiver geographically and enable the receiver to retrieve track information directly from the WFS. In order to accomplish this, there are 2 stages in the notification process. The first is the registration of the client with the notification broker. The broker sits between the client (mobile device) and the notification producer (WFS) and specializes in subscribing to the producer in a generic way, but relaying any notifications from the producer in a client-specific manner. In the OWS-8 scenario, the broker received data from the WFS in XML SOAP messages, but relayed the data as GeoSMS messages.

Clients register with the broker using the WSN-T SOAP protocol. This allows the client to register their event subscription information and an address for notifications to be relayed to the client in this case SMS or email endpoints.

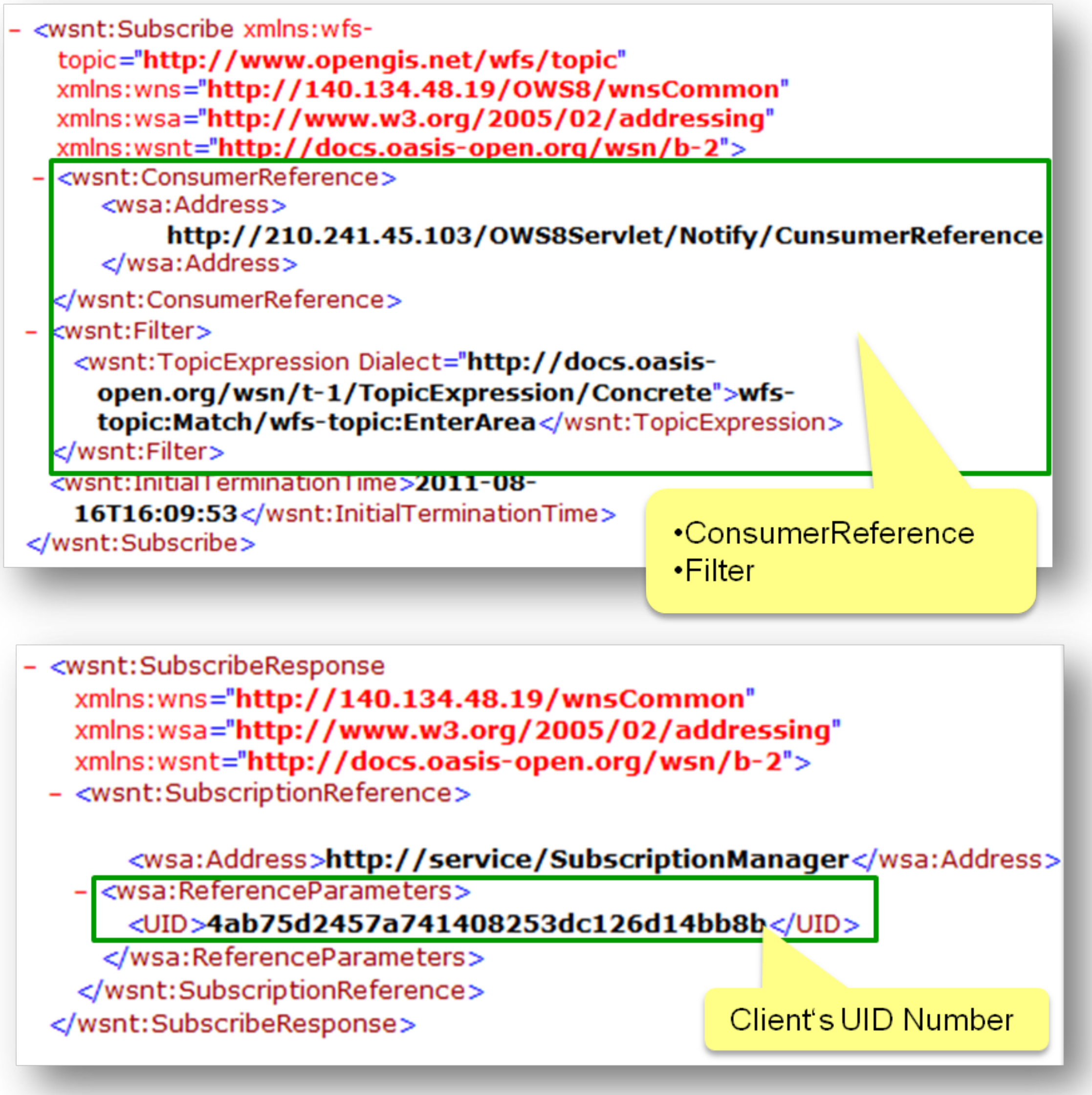


```
- <wsnt:Subscribe
    xmlns:wns="http://140.134.48.19/OWS8/wnsCommon"
    xmlns:wsa="http://www.w3.org/2005/02/addressing"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2" xmlns:wfs-
    topic="http://www.opengis.net/wfs/topic">
  - <wsnt:ConsumerReference>

        <wsa:Address>http://service/CunsumerReference</wsa:Address>
    - <wsa:ReferenceParameters>
      - <wns:Protocols>
          <SMS>true</SMS>
          <Email>true</Email>
        </wns:Protocols>
      - <wns:CommunicationProtocol>
          <SMS>886987654321</SMS>
          <Email>sherry@gis.tw</Email>
        </wns:CommunicationProtocol>
      </wsa:ReferenceParameters>
    </wsnt:ConsumerReference>
  - <wsnt:Filter>
      <wsnt:TopicExpression Dialect="http://docs.oasis-
        open.org/wsn/t-1/TopicExpression/Concrete">wfs-
        topic:Match/wfs-topic:EnterArea</wsnt:TopicExpression>
      <NotificationType>Notification</NotificationType>
    - <NotificationAOIs>
        <NotificationAOI>AOI_1</NotificationAOI>
        <NotificationAOI>AOI_2</NotificationAOI>
        <NotificationAOI>AOI_33</NotificationAOI>
      </NotificationAOIs>
    </wsnt:Filter>
    <wsnt:InitialTerminationTime>2011-08-
      01T18:40:38</wsnt:InitialTerminationTime>
  </wsnt:Subscribe>
```

Figure 24 - Subscribing to a Broker

The broker takes this information and retains it for communication back to the client when notifications arrive. Clients are provided a unique identifier which is used for reference when the notification arrives. The broker then forwards the subscription information to the Subscription Manager in a format that the WFS can make use of. A subscription to a broker might be a high level representation of a filter by AOI and feature type, but the broker has the responsibility to encode this subscription as a WSN Filter which is usable by the WFS.



```
- <wsnt:Subscribe xmlns:wfs-
   topic="http://www.opengis.net/wfs/topic"
   xmlns:wns="http://140.134.48.19/OWS8/wnsCommon"
   xmlns:wsa="http://www.w3.org/2005/02/addressing"
   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
  - <wsnt:ConsumerReference>
      <wsa:Address>
         http://210.241.45.103/OWS8Servlet/Notify/CunsumerReference
      </wsa:Address>
    </wsnt:ConsumerReference>
  - <wsnt:Filter>
     <wsnt:TopicExpression Dialect="http://docs.oasis-
       open.org/wsn/t-1/TopicExpression/Concrete">wfs-
       topic:Match/wfs-topic:EnterArea</wsnt:TopicExpression>
    </wsnt:Filter>
    <wsnt:InitialTerminationTime>2011-08-
     16T16:09:53</wsnt:InitialTerminationTime>
  </wsnt:Subscribe>
```

•ConsumerReference
•Filter

```
- <wsnt:SubscribeResponse
   xmlns:wns="http://140.134.48.19/wnsCommon"
   xmlns:wsa="http://www.w3.org/2005/02/addressing"
   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
 - <wsnt:SubscriptionReference>

     <wsa:Address>http://service/SubscriptionManager</wsa:Address>
   - <wsa:ReferenceParameters>
       <UID>4ab75d2457a741408253dc126d14bb8b</UID>
     </wsa:ReferenceParameters>
   </wsnt:SubscriptionReference>
 </wsnt:SubscribeResponse>
```

Client's UID Number

Figure 25 - Subscription from Broker to Subscription Manager

When notifications match a client subscription at the WFS, the broker will match the notification to a particular client. This will allow the broker to identify the endpoint, protocol and format used to forward the notification to the client e.g. a phone number , SMS and an appropriate GeoSMS message.

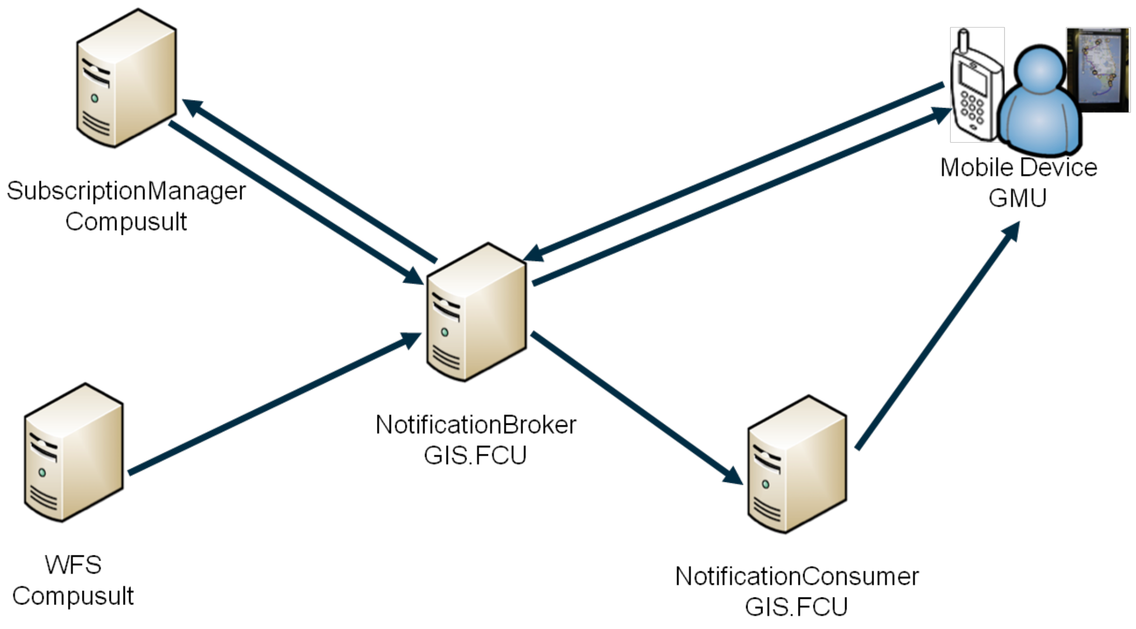
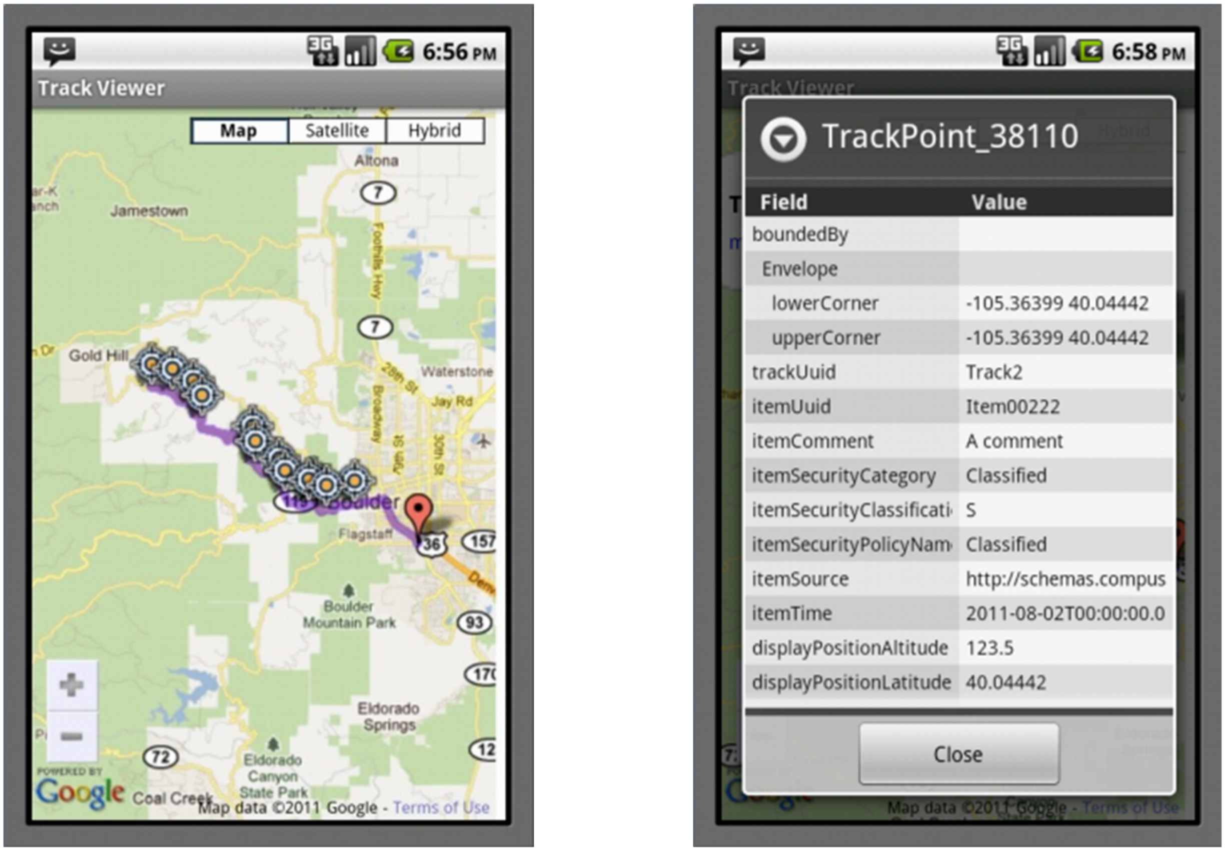


Figure 26 -- Notification Process

The process for registering with the broker and the subsequent WFS notifications is illustrated in Figure 26 -- Notification Process.

10.3 Mobile Client Implementation

The Mobile Client is an Android-based application which supports the OGC GeoSMS 1.0 standard and is capable of handling WFS features and KML files. Once the mobile client receives a GeoSMS, the URL is decoded to retrieve the WFS track feature. The url to the WFS feature is an encoded tinyURL url to the GetFeature request. The application uses the Google Maps interface to display the track.



TrackPoint_38110

Field	Value
boundedBy	
Envelope	
lowerCorner	-105.36399 40.04442
upperCorner	-105.36399 40.04442
trackUuid	Track2
itemUuid	Item00222
itemComment	A comment
itemSecurityCategory	Classified
itemSecurityClassificati	S
itemSecurityPolicyNam	Classified
itemSource	http://schemas.compus
itemTime	2011-08-02T00:00:00.0
displayPositionAltitude	123.5
displayPositionLatitude	40.04442

Figure 27—Screen shot of Mobile Client

Annex A

Example Requests and Responses

This Annex is designed to illustrate requests and responses used by components of the OWS-8 Observation Fusion Tracking sub-thread. Each request and response is labeled for cross-reference in the main body of this Engineering Report.

A.1 Sample SOS Frame GetObservation Request and Response

```
http://ows-
8.compusult.net/VMTI_SOS_2_0/SOS?request=GetObservation&service=SOS&version
=2.0.0&responseFormat=text/xml;subtype=%22om/2.0.0%22&offering=Frame_Bridge&
observedProperty=urn:misb:def:0601:Checksum,urn:misb:def:eg0903:FrameHeight,urn:
misb:def:eg0903:FrameWidth,urn:misb:def:eg0903:NumberOfDetectedTargets,urn:misb:
def:eg0903:NumberOfReportedTargets,urn:misb:def:eg0903:VMTILDSVersionNumber,
urn:misb:def:eg0903:VMTISensorHFOV,urn:misb:def:eg0903:VMTISensorVFOV,urn:
misb:def:eg0903:VMTISourceSensor,urn:misb:def:eg0903:VMTISystemName,urn:misb:
def:eg0903:VideoFrameNumber,urn:misb:def:eg0903:FrameURL&temporalFilter=2010-
06-02T01:05:50.000Z/2010-06-02T01:05:53.000Z
```

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservationResponse xmlns="http://www.opengis.net/sos/2.0"
xmlns:sos="http://www.opengis.net/sos/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:om="http://www.opengis.net/om/2.0" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:swes="http://www.opengis.net/swes/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><observationData>
<OM_Observation gml:id="226691" xmlns="http://www.opengis.net/om/2.0">
<om:type xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement"/><om:phenomenonTime><gml:TimeInstant
gml:id="phenomenontime226691">
<gml:timePosition>2010-06-02T01:05:52.613Z</gml:timePosition>
</gml:TimeInstant></om:phenomenonTime>
<om:resultTime xlink:href="#phenomenontime226691"/><om:procedure
xlink:href="urn:misb:def:0601:VideoSource4"/><om:parameter><om:NamedValue><om:name
xlink:href="http://www.opengis.net/req/omxml/2.0/data/samplingGeometry"/><om:value>
<gml:Polygon srsName="EPSG:4326"
xmlns:gml="http://www.opengis.net/gml"><gml:exterior><gml:LinearRing><gml:posList
srsDimension="2">138.50555538051267 -34.84308962838208 138.50178558162963 -
34.847340093789356 138.49644331654588 -34.84412191696511 138.49990869329687 -
34.84007516260859 138.50555538051267 -34.84308962838208
</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></om:value></om:NamedValue>
</om:parameter>
<om:observedProperty xlink:href="urn:misb:def:0601:Checksum"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:FrameHeight"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:FrameWidth"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:NumberOfDetectedTargets"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:NumberOfReportedTargets"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:VMTILDSVersionNumber"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:VMTISensorHFOV"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:VMTISensorVFOV"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:VMTISourceSensor"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:VMTISystemName"/>
<om:observedProperty xlink:href="urn:misb:def:eg0903:VideoFrameNumber"/>
```

```
<om:observedProperty xlink:href="urn:misb:def:eg0903:FrameURL"/>
<om:result>
<swe:DataRecord gml:id="226691">
<swe:field name="VMTI_CHECKSUM">
<swe:Quantity definition="urn:misb:def:0601:Checksum">
<swe:value>63576</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="FRAME_HEIGHT">
<swe:Quantity definition="urn:misb:def:eg0903:FrameHeight">
<swe:value>480</swe:value>
<swe:uom code="pixels"/>
</swe:Quantity>
</swe:field>
<swe:field name="FRAME_WIDTH">
<swe:Quantity definition="urn:misb:def:eg0903:FrameWidth">
<swe:value>720</swe:value>
<swe:uom code="pixels"/>
</swe:Quantity>
</swe:field>
<swe:field name="NUMBER_OF_DETECTED_TARGETS">
<swe:Quantity definition="urn:misb:def:eg0903:NumberOfDetectedTargets">
<swe:value>1</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="NUMBER_OF_REPORTED_TARGETS">
<swe:Quantity definition="urn:misb:def:eg0903:NumberOfReportedTargets">
<swe:value>1</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="VMTI_LDS_VERSION_NUMBER">
<swe:Quantity definition="urn:misb:def:eg0903:VMTILDSVersionNumber">
<swe:value>0</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="VMTI_SENSOR_HFOV">
<swe:Quantity definition="urn:misb:def:eg0903:VMTISensorHFOV">
<swe:value>4.47974</swe:value>
<swe:uom code="degrees"/>
</swe:Quantity>
</swe:field>
<swe:field name="VMTI_SENSOR_VFOV">
<swe:Quantity definition="urn:misb:def:eg0903:VMTISensorVFOV">
<swe:value>null</swe:value>
<swe:uom code="degrees"/>
</swe:Quantity>
</swe:field>
<swe:field name="VMTI_SOURCE_SENSOR">
<swe:Text definition="urn:misb:def:eg0903:VMTISourceSensor">
<swe:value>EOW</swe:value>
</swe:Text>
</swe:field>
<swe:field name="VMTI_SYSTEM_NAME">
<swe:Text definition="urn:misb:def:eg0903:VMTISystemName">
<swe:value>null</swe:value>
</swe:Text>
</swe:field>
<swe:field name="VIDEO_FRAME_NUMBER">
<swe:Quantity definition="urn:misb:def:eg0903:VideoFrameNumber">
<swe:value>1499</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="FRAME_URL">
<swe:Text definition="urn:misb:def:eg0903:FrameURL">
<swe:value>http://ows-
8.compusult.net/VideoSOS/SOS?request=GetObservation&amp;service=SOS&amp;version=2.0.0&amp
;responseFormat=image/jpeg&amp;offering=Frame_Bridge&amp;temporalFilter=2010-06-
02T01:05:52.613Z</swe:value>
</swe:Text>
</swe:field>
```

```
</swe:DataRecord>
</om:result></OM_Observation>…
```

A.2 Sample SOS Target GetObservation Request and Response

```
http://ows-8.compusult.net/VMTI_SOS_2_0/SOS?request=GetObservation&service=SOS&version=2.0.0&responseFormat=text/xml;subtype=%22om/2.0.0%22&offering=TargetResult_Bridge&observedProperty=urn:misb:def:eg0903:Elevation,urn:misb:def:eg0903:Latitude,urn:misb:def:eg0903:Longitude,urn:misb:def:eg0903:NewDetectionTargetHistoryFlag,urn:misb:def:eg0903:PercentTargetPixels,urn:misb:def:eg0903:PixelNumberBottomRight,urn:misb:def:eg0903:PixelNumberCentroid,urn:misb:def:eg0903:PixelNumberTopLeft,urn:misb:def:eg0903:TargetColor,urn:misb:def:eg0903:TargetConfidenceLevel,urn:misb:def:eg0903:TargetID,urn:misb:def:eg0903:TargetIntensity,urn:misb:def:eg0903:TargetPriority&temporalFilter=2010-06-02T01:05:50.000Z/2010-06-02T01:05:53.000Z
```

```xml
<?xml version="1.0" encoding="UTF-8"?><sos:GetObservationResponse
xmlns="http://www.opengis.net/sos/2.0" xmlns:sos="http://www.opengis.net/sos/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:om="http://www.opengis.net/om/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:swes="http://www.opengis.net/swes/2.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><observationData>
<OM_Observation gml:id="226692" xmlns="http://www.opengis.net/om/2.0" >
<om:type xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement"/><om:phenomenonTime><gml:TimeInstant
gml:id="phenomenontime226692">
<gml:timePosition>2010-06-02T01:05:52.613Z</gml:timePosition>
</gml:TimeInstant></om:phenomenonTime>
<om:resultTime xlink:href="#phenomenontime226692" /><om:procedure
xlink:href="urn:misb:def:eg0903:VMTIDetector3" /><om:parameter><om:NamedValue><om:name
xlink:href="http://www.opengis.net/req/omxml/2.0/data/samplingGeometry"/><om:value>
<gml:Polygon srsName="EPSG:4326"
xmlns:gml="http://www.opengis.net/gml"><gml:exterior><gml:LinearRing><gml:posList
srsDimension="2">138.50075870634086 -34.845127318294544 138.50075870634086 -
34.845127318294544 138.50075870634086 -34.845127318294544 138.50075870634086 -
34.845127318294544
</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></om:value>
</om:NamedValue>
</om:parameter>
<om:observedProperty xlink:href="urn:misb:def:eg0903:Elevation" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:Latitude" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:Longitude" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:NewDetectionTargetHistoryFlag" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:PercentTargetPixels" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:PixelNumberBottomRight" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:PixelNumberCentroid" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:PixelNumberTopLeft" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:TargetColor" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:TargetConfidenceLevel" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:TargetID" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:TargetIntensity" />
<om:observedProperty xlink:href="urn:misb:def:eg0903:TargetPriority" />
<om:result>
<swe:DataRecord gml:id="226692">
<swe:field name="ELEVATION">
<swe:Quantity definition="urn:misb:def:eg0903:Elevation">
<swe:value>null</swe:value>
<swe:uom code="meters" />
</swe:Quantity>
</swe:field>
<swe:field name="LATITUDE">
<swe:Quantity definition="urn:misb:def:eg0903:Latitude">
<swe:value>-34.845127</swe:value>
<swe:uom code="degrees" />
```

```
</swe:Quantity>
</swe:field>
<swe:field name="LONGITUDE">
<swe:Quantity definition="urn:misb:def:eg0903:Longitude">
<swe:value>138.500759</swe:value>
<swe:uom code="degrees" />
</swe:Quantity>
</swe:field>
<swe:field name="NEW_DETECTION_TARGET_HISTORY">
<swe:Quantity definition="urn:misb:def:eg0903:NewDetectionTargetHistoryFlag">
<swe:value>5307</swe:value>
<swe:uom code="frames" />
</swe:Quantity>
</swe:field>
<swe:field name="PERCENT_TARGET_PIXELS">
<swe:Quantity definition="urn:misb:def:eg0903:PercentTargetPixels">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="PIXEL_NUMBER_BOTTOM_RIGHT">
<swe:Quantity definition="urn:misb:def:eg0903:PixelNumberBottomRight">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="PIXEL_NUMBER_CENTROID">
<swe:Quantity definition="urn:misb:def:eg0903:PixelNumberCentroid">
<swe:value>120778</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="PIXEL_NUMBER_TOP_LEFT">
<swe:Quantity definition="urn:misb:def:eg0903:PixelNumberTopLeft">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="TARGET_COLOR">
<swe:Quantity definition="urn:misb:def:eg0903:TargetColor">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="TARGET_CONFIDENCE_LEVEL">
<swe:Quantity definition="urn:misb:def:eg0903:TargetConfidenceLevel">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="TARGET_ID">
<swe:Quantity definition="urn:misb:def:eg0903:TargetID">
<swe:value>1</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="TARGET_INTENSITY">
<swe:Quantity definition="urn:misb:def:eg0903:TargetIntensity">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>
<swe:field name="TARGET_PRIORITY">
<swe:Quantity definition="urn:misb:def:eg0903:TargetPriority">
<swe:value>null</swe:value>
</swe:Quantity>
</swe:field>

</swe:DataRecord>
</om:result>

</OM_Observation>…
```

A.3 Sample WPS Execute request

This example initiates an Execute process in the tracker WPS. Inputs are requests for Frame metadata, Raw Video and TargetResult metadata requested from an SOS.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd ">
    <ows:Identifier>edu.gmu.csiss.wps.server.algorithm.VideoTrackingAlgorithm</ows:Identi
fier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>frames</ows:Identifier>
            <wps:Reference mimeType="application/xml"
                  schema="http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
xlink:href="http://ows-
8.compusult.net/VMTI_SOS_2_0/SOS?request=GetObservation&amp;service=SOS&amp;version=2.0.0
&amp;responseFormat=text/xml;subtype=%22om/2.0.0%22&amp;offering=Frame_Bridge&amp;observe
dProperty=urn:misb:def:0601:Checksum,urn:misb:def:eg0903:FrameHeight,urn:misb:def:eg0903:
FrameWidth,urn:misb:def:eg0903:NumberOfDetectedTargets,urn:misb:def:eg0903:NumberOfReport
edTargets,urn:misb:def:eg0903:VMTILDSVersionNumber,urn:misb:def:eg0903:VMTISensorHFOV,urn
:misb:def:eg0903:VMTISensorVFOV,urn:misb:def:eg0903:VMTISourceSensor,urn:misb:def:eg0903:
VMTISystemName,urn:misb:def:eg0903:VideoFrameNumber,urn:misb:def:eg0903:FrameURL&amp;temp
oralFilter=2010-06-02T01:05:41.635Z/2010-06-02T01:06:09.746Z">
            </wps:Reference>
        </wps:Input>
        <wps:Input>
            <ows:Identifier>detections</ows:Identifier>
            <wps:Reference mimeType="application/xml"
schema=http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd
xlink:href="http://ows-
8.compusult.net/VMTI_SOS_2_0/SOS?request=GetObservation&amp;service=SOS&amp;version=2.0.0
&amp;responseFormat=text/xml;subtype=%22om/2.0.0%22&amp;offering=TargetResult_Bridge&amp;
observedProperty=urn:misb:def:eg0903:Elevation,urn:misb:def:eg0903:Latitude,urn:misb:def:
eg0903:Longitude,urn:misb:def:eg0903:NewDetectionTargetHistoryFlag,urn:misb:def:eg0903:Pe
rcentTargetPixels,urn:misb:def:eg0903:PixelNumberBottomRight,urn:misb:def:eg0903:PixelNum
berCentroid,urn:misb:def:eg0903:PixelNumberTopLeft,urn:misb:def:eg0903:TargetColor,urn:mi
sb:def:eg0903:TargetConfidenceLevel,urn:misb:def:eg0903:TargetID,urn:misb:def:eg0903:Targ
etIntensity,urn:misb:def:eg0903:TargetPriority&amp;temporalFilter=2010-06-
02T01:05:41.635Z/2010-06-02T01:06:09.746Z">
            </wps:Reference>
        </wps:Input>
        <wps:Input>
            <ows:Identifier>video</ows:Identifier>
            <wps:Reference mimeType="video/avi"
schema="http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
xlink:href="http://ows-
8.compusult.net/VideoSOS/SOS?request=GetObservation&amp;service=SOS&amp;version=2.0.0&amp
;offering=Video_Bridge&amp;responseFormat=video/mpeg&amp;temporalFilter=2010-06-
02T01:05:00.010Z/2010-06-02T01:06:21.000Z">
            </wps:Reference>
        </wps:Input>
        <wps:Input>
            <ows:Identifier>parameters</ows:Identifier>
            <wps:Data>
                <wps:ComplexData mimeType="application/xml"
schema="http://ws.laits.gmu.edu/tracksparameters.xsd">
                    <tp:TracksParameters
xmlns:tp="http://ws.laits.gmu.edu/tracksparameters"

    xsi:schemaLocation="http://ws.laits.gmu.edu/tracksparameters
http://ws.laits.gmu.edu/tracksparameters.xsd">
                        <tp:TrackParameter>
                            <tp:TrackNumber>1</tp:TrackNumber>
                            <tp:XStopThreshold>0.07</tp:XStopThreshold>
                            <tp:YStopThreshold>0.07</tp:YStopThreshold>
                            <tp:XMotivationRatio>1.18</tp:XMotivationRatio>
                            <tp:YMotivationRatio>1.18</tp:YMotivationRatio>
                            <tp:WindowWidth>5</tp:WindowWidth>
                            <tp:WindowHeight>5</tp:WindowHeight>
                            <tp:SearchRange>15</tp:SearchRange>
                            <tp:BackTrackNumber>7</tp:BackTrackNumber>
                        </tp:TrackParameter>
                        <tp:TrackParameter>
                            <tp:TrackNumber>2</tp:TrackNumber>
                            <tp:XStopThreshold>0.09</tp:XStopThreshold>
                            <tp:YStopThreshold>0.09</tp:YStopThreshold>
                            <tp:XMotivationRatio>1.15</tp:XMotivationRatio>
                            <tp:YMotivationRatio>1.15</tp:YMotivationRatio>
                            <tp:WindowWidth>5</tp:WindowWidth>
                            <tp:WindowHeight>5</tp:WindowHeight>
                            <tp:SearchRange>5</tp:SearchRange>
                            <tp:BackTrackNumber>7</tp:BackTrackNumber>
                        </tp:TrackParameter>
                    </tp:TracksParameters>
                </wps:ComplexData>
            </wps:Data>
        </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="true" status="true">
        <wps:Output asReference="true">
            <ows:Identifier>TracksResult</ows:Identifier>
        </wps:Output>
    </wps:ResponseDocument>
    </wps:ResponseForm>

</wps:Execute>
```

A.4 Sample WPS ExecuteResponse

The response contains a URL to a status servlet that clients can use to determine the status of the executing process.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ExecuteResponse xmlns:ns="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd"
serviceInstance="http://ws.laits.gmu.edu:/trackingwpsr3/WebProcessingService?REQUEST=GetC
apabilities&amp;SERVICE=WPS"
xml:lang="en-US" service="WPS" version="1.0.0"
statusLocation="http://ws.laits.gmu.edu:/trackingwpsr3/RetrieveResultServlet?id=131509287
9217">
    <ns:Process ns:processVersion="1.0.0">
        <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">edu.gmu.csiss.wps.server.algorithm.VideoTracki
ngAlgorithm</ns1:Identifier>
        <ns1:Title
xmlns:ns1="http://www.opengis.net/ows/1.1">edu.gmu.csiss.wps.server.algorithm.VideoTracki
ngAlgorithm</ns1:Title>
    </ns:Process>
    <ns:Status creationTime="2011-09-03T19:34:39.182-04:00">
        <ns:ProcessStarted percentCompleted="0"/>
    </ns:Status>
</ns:ExecuteResponse>
```

A.5 Sample WPS Status Response

A response to a status request to the RetrieveResultsServlet provides a successful status result and a link to retrieve the result information.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ExecuteResponse xmlns:ns="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd"
```

```
serviceInstance="http://ws.laits.gmu.edu:/trackingwpsr3/WebProcessingService?REQUEST=GetC
apabilities&amp;SERVICE=WPS"
xml:lang="en-US" service="WPS" version="1.0.0"
statusLocation="http://ws.laits.gmu.edu:/trackingwpsr3/RetrieveResultServlet?id=131509287
9217">
    <ns:Process ns:processVersion="1.0.0">
        <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">edu.gmu.csiss.wps.server.algorithm.VideoTracki
ngAlgorithm</ns1:Identifier>
        <ns1:Title
xmlns:ns1="http://www.opengis.net/ows/1.1">edu.gmu.csiss.wps.server.algorithm.VideoTracki
ngAlgorithm</ns1:Title>
    </ns:Process>
    <ns:Status creationTime="2011-09-03T19:34:39.182-04:00">
        <ns:ProcessSucceeded>Process successful</ns:ProcessSucceeded>
    </ns:Status>
    <ns:ProcessOutputs>
        <ns:Output>
            <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">TracksResult</ns1:Identifier>
            <ns1:Title
xmlns:ns1="http://www.opengis.net/ows/1.1">TracksResult</ns1:Title>
            <ns:Reference
schema="http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd" encoding="UTF-8"
mimeType="text/xml"

    href="http://ws.laits.gmu.edu:/trackingwpsr3/RetrieveResultServlet?id=1315092879217Tr
acksResultresult-58ea62b5-a202-4760-a5b5-51707f9053b9"/>
        </ns:Output>
    </ns:ProcessOutputs>
</ns:ExecuteResponse>
```

A.6 Sample Status Response showing WFS Insert Results

The result of following the link highlighted in A.5 is a result showing the insert of features into the WFS.

```
<?xml version="1.0" encoding="UTF-8"?>
<TracksResult xmlns="http://ws.laits.gmu.edu/tracksresult"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ws.laits.gmu.edu/tracksresult
http://ws.laits.gmu.edu/tracksresult.xsd">
    <wfs:TransactionResponse xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:wfs="http://www.opengis.net/wfs" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    version="1.1.0" xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
        <TransactionSummary xmlns="http://www.opengis.net/wfs">
            <totalInserted>2</totalInserted>
            <totalUpdated>0</totalUpdated>
            <totalDeleted>0</totalDeleted>
        </TransactionSummary>
        <InsertResults xmlns="http://www.opengis.net/wfs">
            <Feature handle="WPS_insert_WFS_Track_858507493">
                <ogc:FeatureId fid="Track_637"/>
            </Feature>
            <Feature handle="WPS_insert_WFS_Track_1354468546">
                <ogc:FeatureId fid="Track_638"/>
            </Feature>
        </InsertResults>
    </wfs:TransactionResponse>
</TracksResult>
```

A.7 Sample Track Point Insert

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml"
    xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:wes="http://schemas.compusult.net/tracks"
    xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd"
    service="WFS" version="1.1.0">
    <wfs:Insert handle="Insert_Track_1" idgen="GenerateNew">
        <wfs:FeatureCollection>
            <gml:featureMember>
                <wes:TrackPoint>
                    <wes:trackUuid>Test001</wes:trackUuid>
                    <wes:itemUuid>Item005</wes:itemUuid>
                    <wes:itemComment>A point item used for testing</wes:itemComment>
                    <wes:itemSecurityCategory>Classified</wes:itemSecurityCategory>
                    <wes:itemSecurityClassification>S</wes:itemSecurityClassification>
                    <wes:itemSecurityPolicyName>Classified</wes:itemSecurityPolicyName>
                    <wes:itemSource>http://schemas.compusult.net/tracks</wes:itemSource>
                    <wes:itemTime>2011-06-19T13:46:39</wes:itemTime>
                    <wes:displayPositionAltitude>123.5</wes:displayPositionAltitude>
                    <wes:displayPositionLatitude>123.5</wes:displayPositionLatitude>
                    <wes:displayPositionLongitude>44.5</wes:displayPositionLongitude>
                    <wes:trackPointObjectMask>
                        <wes:TrackPointObjectMask>
                            <wes:trackPosition>
                                <wes:TrackPosition>
                                    <wes:altitude>135.6</wes:altitude>
                                    <wes:latitude>89.2</wes:latitude>
                                    <wes:longitude>125.5</wes:longitude>
                                    <wes:radius>6</wes:radius>
                                </wes:TrackPosition>
                            </wes:trackPosition>
                        </wes:TrackPointObjectMask>
                    </wes:trackPointObjectMask>
                    <wes:source>ESM</wes:source>
                    <wes:status>DROPPING</wes:status>
                    <wes:type>AUTOMATIC_ESTIMATED</wes:type>
                    <wes:accelerationAzimuth>10.1</wes:accelerationAzimuth>
                    <wes:accelerationElevation>11.2</wes:accelerationElevation>
                    <wes:accelerationRange>1.14</wes:accelerationRange>
                    <wes:accelerationX>32.1</wes:accelerationX>
                    <wes:accelerationY>43.2</wes:accelerationY>
                    <wes:accelerationZ>123.987</wes:accelerationZ>
                    <wes:trackPointPosition>
                        <wes:TrackPointPosition>
                            <wes:trackPosition>
                                <wes:TrackPosition>
                                    <wes:altitude>135.6</wes:altitude>
                                    <wes:latitude>89.2</wes:latitude>
                                    <wes:longitude>125.5</wes:longitude>
                                    <wes:radius>6</wes:radius>
                                </wes:TrackPosition>
                            </wes:trackPosition>
                        </wes:TrackPointPosition>
                    </wes:trackPointPosition>
                    <wes:velocityCourse>12.8</wes:velocityCourse>
                    <wes:velocityElevationAngle>1.54</wes:velocityElevationAngle>
                    <wes:velocitySpeed>142.6</wes:velocitySpeed>
                    <wes:velocityX>10.1</wes:velocityX>
                    <wes:velocityY>11.2</wes:velocityY>
                    <wes:velocityZ>13.3</wes:velocityZ>
                    <wes:rhoPosxPosy>12.3</wes:rhoPosxPosy>
                    <wes:rhoPosxPosz>14.4</wes:rhoPosxPosz>
                    <wes:rhoPosyPosz>15.2</wes:rhoPosyPosz>
                    <wes:sigmaPosx>18.1</wes:sigmaPosx>
                    <wes:sigmaPosy>17.2</wes:sigmaPosy>
                    <wes:sigmaPosz>14.2</wes:sigmaPosz>
                    <wes:rhoPosxVelx>16.2</wes:rhoPosxVelx>
                    <wes:rhoPosxVely>18.3</wes:rhoPosxVely>
                    <wes:rhoPosxVelz>19.1</wes:rhoPosxVelz>
                    <wes:rhoPosyVelx>16.7</wes:rhoPosyVelx>
                    <wes:rhoPosyVely>15.3</wes:rhoPosyVely>
                    <wes:rhoPosyVelz>18.9</wes:rhoPosyVelz>
                    <wes:rhoPoszVelx>13.4</wes:rhoPoszVelx>
                    <wes:rhoPoszVely>11.6</wes:rhoPoszVely>
                    <wes:rhoPoszVelz>16.6</wes:rhoPoszVelz>
                    <wes:rhoVelxVely>15.5</wes:rhoVelxVely>
                    <wes:rhoVelxVelz>14.4</wes:rhoVelxVelz>
                    <wes:rhoVelyVelz>16.7</wes:rhoVelyVelz>
                    <wes:sigmaVelx>14.7</wes:sigmaVelx>
                    <wes:sigmaVely>12.1</wes:sigmaVely>
                    <wes:sigmaVelz>10.2</wes:sigmaVelz>
                </wes:TrackPoint>
            </gml:featureMember>
        </wfs:FeatureCollection>
    </wfs:Insert>
</wfs:Transaction>
```

Annex B

XML Schema Documents

In addition to this document, this report includes a non-normative XML Schema Documents. This XML Schema Document is bundled in a zip file with the present document.

The Track features derived from the STANAG 4676 track specification are defined by the schema document below. This schema document is included in the zip file with this document. This XML Schema Document matches the UML diagram provided in section A.8, and is named Track.xsd :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wes="http://schemas.compusult.net/tracks" xmlns:wfs="http://www.opengis.net/wfs"
attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.compusult.net/tracks">
    <xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"/>
    <xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/geometryAggregates.xsd"/>
    <xsd:import namespace="http://www.opengis.net/wfs"
schemaLocation="http://schemas.opengis.net/wfs/1.1.0/wfs.xsd"/>

  <!-- ====================================================================== -->
    <xsd:element name="Track" substitutionGroup="gml:_Feature" type="wes:TrackType">

    </xsd:element>
    <!-- ====================================================================== -->
    <xsd:complexType name="TrackType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element name="trackNumber" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="trackUuid" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="trackComment" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="trackSecurityCategory" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="trackSecurityClassification" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="trackSecurityPolicyName" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="exerciseIndicator" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="simulationIndicator" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="messageSecurityCategory" type="xsd:string">
                    </xsd:element>
                    <xsd:element name="messageSecurityClassification" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="messageSecurityPolicyName" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="messageCreationTime" type="xsd:dateTime">

                    </xsd:element>
                    <xsd:element name="senderIdNationality" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="senderIdStationId" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="stanagVersion" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="trackGeometry"
type="gml:GeometryPropertyType">

                    </xsd:element>
                    <xsd:element maxOccurs="unbounded" minOccurs="0"
name="trackInformation" type="gml:FeaturePropertyType">

                    </xsd:element>
                    <xsd:element maxOccurs="unbounded" minOccurs="0" name="trackPoint"
type="gml:FeaturePropertyType">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- =================================================================== -->
    <xsd:element name="TrackInformation" substitutionGroup="gml:_Feature"
type="wes:TrackInformationType">

    </xsd:element>
    <!-- =================================================================== -->
    <xsd:complexType name="TrackInformationType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element name="trackUuid" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemUuid" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="itemComment" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemSecurityCategory" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemSecurityClassification" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemSecurityPolicyName" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="itemSource" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemTime" type="xsd:dateTime">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="identificationSource" 
type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="identity" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="identityAmplification" 
type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" 
name="identityCredibilitySourceUncertainty" type="xsd:integer">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="identityCredibilityValueUncertainty" 
type="xsd:integer">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="iifCodeMode" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="iifCodeValue" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="unitName" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="unitSymbol" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="classification" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" 
name="classificationCredibilitySourceUncertainty" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" 
name="classificationCredibilityValueUncertainty" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="classificationSource" 
type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="strength" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="alertIndicator" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="environment" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="quality" type="xsd:integer">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="trackerType" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="trackProductionArea" 
type="gml:FeaturePropertyType">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="band" type="xsd:string">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
```

```
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ====================================================================== -->
    <xsd:element name="TrackPoint" substitutionGroup="gml:_Feature"
type="wes:TrackPointType">

    </xsd:element>
    <!-- ====================================================================== -->
    <xsd:complexType name="TrackPointType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element name="trackUuid" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemUuid" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="itemComment" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemSecurityCategory" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemSecurityClassification" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemSecurityPolicyName" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="itemSource" type="xsd:string">

                    </xsd:element>
                    <xsd:element name="itemTime" type="xsd:dateTime">

                    </xsd:element>
                    <xsd:element name="displayPositionAltitude" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="displayPositionLatitude" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="displayPositionLongitude" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="trackPointObjectMask"
type="gml:FeaturePropertyType">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="source" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="status" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="type" type="xsd:string">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="accelerationAzimuth"
type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="accelerationElevation"
type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="accelerationRange"
type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="accelerationX" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="accelerationY" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="accelerationZ" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="trackPointPosition"
type="gml:FeaturePropertyType">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="velocityCourse" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="velocityElevationAngle"
type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="velocitySpeed" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="velocityX" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="velocityY" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="velocityZ" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosxPosy" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosxPosz" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosyPosz" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="sigmaPosx" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="sigmaPosy" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="sigmaPosz" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosxVelx" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosxVely" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosxVelz" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosyVelx" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosyVely" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPosyVelz" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPoszVelx" type="xsd:double">
                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPoszVely" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoPoszVelz" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoVelxVely" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoVelxVelz" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="rhoVelyVelz" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="sigmaVelx" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="sigmaVely" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="sigmaVelz" type="xsd:double">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ====================================================================== -->
    <xsd:element name="TrackProductionArea" substitutionGroup="gml:_Feature"
type="wes:TrackProductionAreaType">

    </xsd:element>
    <!-- ====================================================================== -->
    <xsd:complexType name="TrackProductionAreaType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" name="trackPosition"
type="gml:FeaturePropertyType">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ====================================================================== -->
    <xsd:element name="TrackPointPosition" substitutionGroup="gml:_Feature"
type="wes:TrackPointPositionType">

    </xsd:element>
    <!-- ====================================================================== -->
    <xsd:complexType name="TrackPointPositionType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" name="trackPosition"
type="gml:FeaturePropertyType">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ====================================================================== -->
    <xsd:element name="TrackPointObjectMask" substitutionGroup="gml:_Feature"
type="wes:TrackPointObjectMaskType">

    </xsd:element>
    <!-- ====================================================================== -->
    <xsd:complexType name="TrackPointObjectMaskType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" name="trackPosition"
type="gml:FeaturePropertyType">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ====================================================================== -->
    <xsd:element name="TrackPosition" substitutionGroup="gml:_Feature"
type="wes:TrackPositionType">

    </xsd:element>
    <!-- ====================================================================== -->
    <xsd:complexType name="TrackPositionType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element minOccurs="0" name="altitude" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="latitude" type="xsd:double">

                    </xsd:element>
                    <xsd:element name="longitude" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="radius" type="xsd:double">

                    </xsd:element>
                    <xsd:element minOccurs="0" name="positionGeometry"
type="gml:GeometryPropertyType">

                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>
```

Bibliography

11 Bibliography

Motion Imagery Standards Board. (2009 йил 03-September). Engineering Guideline MISB EG 0903.0. *Engineering Guideline MISB EG 0903.0* . Motion Imagery Standards Board.

Motion Imagery Standards Board. (2011 йил 24-February). Recommended Practice Video Moving Target Indicator Local Data Set RP 0903.2. Motion Imagery Standards Board.

Motion Imagery Standards Board. (2010 йил 4-March). UAS Datalink Local Metadata Set STANDARD 0601.4. *UAS Datalink Local Metadata Set STANDARD 0601.4* . Motion Imagery Standards Board.

NATO. (2009 йил 13-October). STANAG 4609 JAIS (Edition 3) - NATO Digital Motion Imagery Standard. *STANAG 4609 JAIS (Edition 3) - NATO Digital Motion Imagery Standard* . NATO.

NATO Standardization Agency. (2011 йил 17-February). AEDP-12v3. *NATO INTELLIGENCE, SURVEILLANCE AND RECONNAISSANCE (ISR) (Draft Version 3.0)* . NATO Standardization Agency.

OGC® SWE Common Data Model Encoding Standard. (2011 йил 04-January). *OGC 08-094r1* . Open Geospatial Consortium Inc.

Open Geospatial Consortium Inc. (2010 йил 02-September). OGC SOS 2.0 Interface Standard. *OGC 10-037* . Open Geospatial Consortium Inc.

Open Geospatial Consortium. (2011 йил 22-March). Observations and Measurements - XML Implementation. *OGC 10-025r1* . Open Geospatial Consortium.

Simonis, I. (2011). *OWS-8 Information Model for Moving Target Indicators and Moving Object Bookmarks.* OpenGeospatial Consortium. OpenGeospatial Consortium.

Yu, G., & Chen, Z. (2011). *Tracking WPS Manual version 0.9.* George Mason University.