Open Geospatial Consortium, Inc.

Date: 2010-08-18

Reference number of this document: OGC 10-036r2

Category: OGC Public Engineering Report

Editor: Stan Tillman

OGC® OWS-7 Motion Video Change Detection

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

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

Contents

Page

Figures

Page

OGC® OWS-7 Motion Video Change Detection

1 Introduction

1.1 Scope

This Engineering Report documents the development effort to build a Web Processing Service (WPS) to perform a change detection algorithm on two motion video streams. It will examine the WPS Motion Video Change Detection architecture from various viewpoints in order to describe its purpose, data models, functional decomposition, and interaction between distinct computational components.

1.2 Document contributor contact points

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

Name	Organization
Stan Tillman	Intergraph Corporation
Genong (Eugene) Yu	Center for Spatial Information Sciences & Systems George Mason University
Liping Di	Center for Spatial Information Sciences & Systems George Mason University
Zeqiang Chen	Center for Spatial Information Sciences & Systems George Mason University

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
03/26/2010	Draft	Stan Tillman		Draft Release.
06/01/2010	Initial Release	Stan Tillman	Major Updates	Document to be presented to the OGC TC.
06/25/2010	1.0.0	Stan Tillman	Final Updates	Final Engineering Report
08/02/2010	1.0.1	Stan Tillman	First Revision	To include comments from testbed sponsor review

1.4 Future work

Improvements in this document are desirable to address open issues; to correct errors or enhance existing document content.

Within the OWS-7 testbed, a lack of data sources with appropriate metadata caused some limiting of the functionality in the study. Future work might include:

- The ability to consume near real-time video at higher frame rates to evaluate a more real-world environment.
- Demonstrate more client control over the WPS by allowing rules to drive the change detection algorithm as well as “sampling” to help the user know how to adjust those rules
- Providing the ability to calculate the “certainty” or “uncertainty” of the identified changes
- Better client integration to show the changed frames overlaid with the live video frames
- With the proliferation of UAVs, it would also be interesting to see how this change detection process works from an overhead perspective

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

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OpenGIS® Web Services Common Standard*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this document.

OGC 05-007r7, *OpenGIS® Web Processing Service Standard*

OGC 06-009r6, *OpenGIS® Sensor Observation Service Standard*

OGC 07-006r1, *OpenGIS® Catalogue Service Implementation Specification.*

OGC 10-061, *OWS-7 Dynamic Sensor Notification Engineering Report.*

OGC 10-087, *Motion Imagery Discovery and Retrieval.*

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

3.1
catalog
A mechanism containing interfaces to discover, browse, and query metadata about data, services, and other potential resources.

3.2
integrated client
A software application that has the capability of bringing together offerings from multiple OGC web services. A typical integrated client incorporates a legend, a map view, a service connection manager and a query facility.

3.3
sensor observation service (SOS)
A web service that provides operations for accessing and manipulating various sensor offerings over HTTP.

3.4
web processing service (WPS)
A web service that provides operations for performing various funcitons on given inputs and responding with the results of those operations.

4 Conventions

4.1 Abbreviated terms

AOI	Area Of Interest
CS/W	Catalog Services for the Web
ER	Engineering Report
GIS	Geo Information Systems
HTTP	Hypertext Transfer Protocol
IED	Improvised Explosive Device

OGC	Open Geospatial Consortium
OWS	OGC Web Service
SFE	Sensor Fusion Enablement
SOA	Service Oriented Architecture
SOS	Sensor Observation Service
SOS-T	Sensor Observation Service (Transactional)
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WPS	Web Processing Service
XML	Extensible Markup Language

4.2 UML notation

Most 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 OWS-7 Motion Video Change Detection Overview

The specified Motion Video Change Detection ER documents the findings of the investigation and experiments to deploy a rules-based motion video change detection processing service that identifies changes between two motion video streams over the same geographic area.

The world community faces growing challenges from the proliferation of various types of data sources. This is especially true for motion video. Although the high volume of video data is beneficial, it creates a number of data management and analytical challenges. Detecting changes from imagery over time has been a vital intelligence function for decades. But with the increased use of video, new and more efficient tools are required. Analysts cannot be expected to detect small changes over very large geographical areas using only the naked eye. Thus the need for a processing service to perform an automated comparison of video streams over the same geographical area that identifies small changes is critical to handling these large data volumes. Figure 1 shows a high-level architecture of the Motion Video Change Detection process.

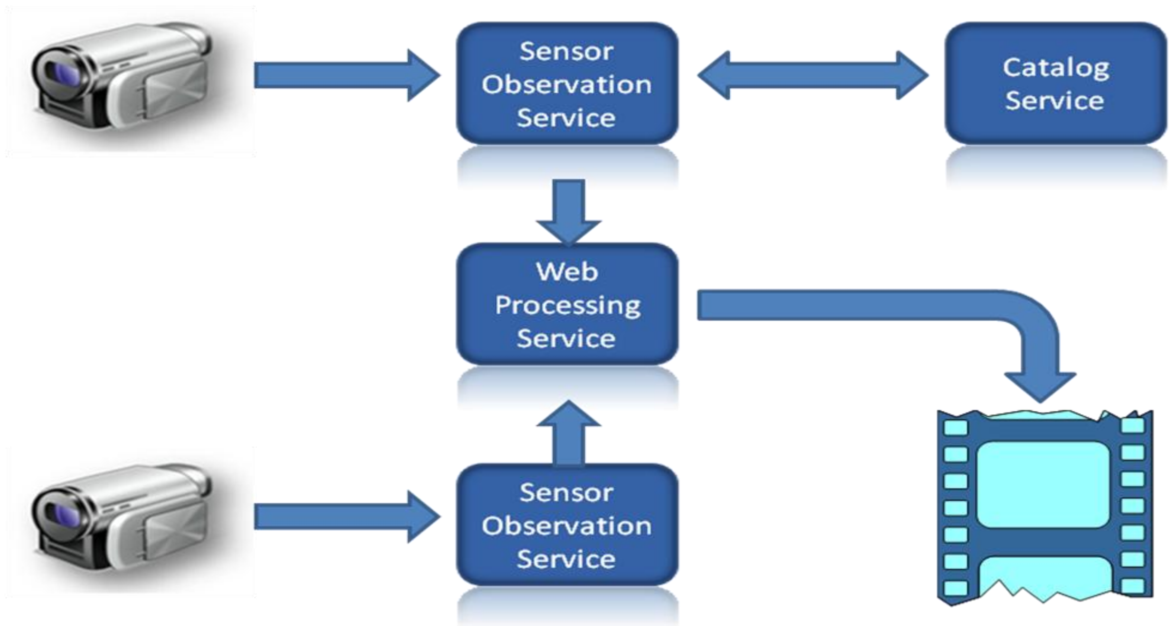


Figure 1 — Motion Video Change Detection WPS Overview

6 OWS-7 Motion Video Change Detection Architecture

6.1 Motion Video Change Detection Enterprise Viewpoint

The Enterprise Viewpoint describes the purpose of the architecture.

The proliferation of video data is a valuable commodity. However, the value of that data lies in the ability to identify changes and intelligible information to improve the capabilities of the decision maker. Within the intelligence, security, and public safety domains, a decision maker's ability to identify changes between video sources over time can be directly responsible for saving lives by allowing potential risks to be seen and thus avoided.

For example, a video of a street view taken from a ground based vehicle or an aerial view taken from a UAV is recorded and cataloged. A few weeks later, another ground based vehicle or UAV is recording the same street view or aerial view. This live video stream can be compared to the one recorded weeks earlier to identify new rock or garbage piles on the side of the road as potential hiding locations for IEDs. This identification will allow convoys to avoid the area until it can be investigated.

A high-level depiction of the Motion Video Change Detection concept is shown in the use case diagram in Figure 1.

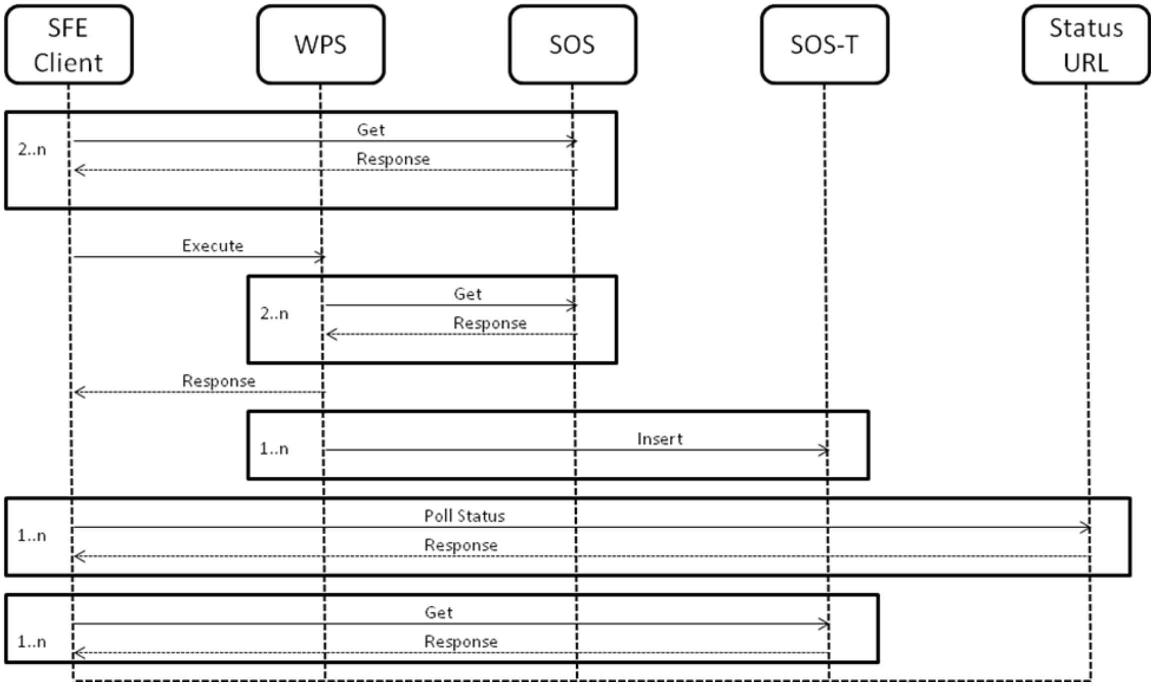


Figure 1 — Motion Video Change Detection Use Case Diagram

The following scenario details a typical flow of actions based on a ground based vehicle:

1. Ground vehicle equipped with video camera enters an area of interest (AOI) and is acquiring motion imagery.
2. As the AOI was previously identified as a potential risk zone, the entry of the vehicle into the AOI triggers an event. The action associated with the event is to begin a change detection process on the motion imagery.
3. Activity of the Service for sensor tracking and notification results in the WPS for Change Detection to begin.
4. WPS begins change detection on live motion imagery stream and previously acquired imagery from the same track.
5. WPS Change Detection is performed again with the new rule parameters. Resulting change detection highlights areas of new debris on the side of a road, significant soil coloration changes, etc. Uncertainty of the second result is less than the first.
6. Crew changes the vehicle route based upon change detection results.

Within the OWS-7 testbed, the Change Detection process was performed on two video streams provided through Sensor Observation Services. As changes are detected, they are highlighted as in the examples below and inserted into a third Sensor Observation

Service. The WPS response contains a status URL that can be polled for processing status as well as the endpoint to the SOS from with the updated video can be obtained.

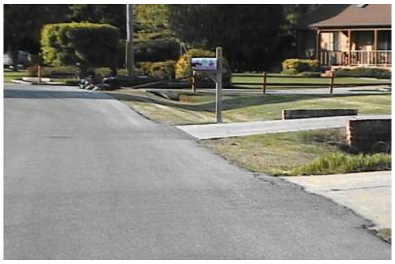
Video1	Video2	Change
		

Figure 3 — Example Change Detection Frames

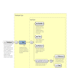
6.2 Motion Video Change Detection Information Viewpoint

The Information Viewpoint presents a description of the architecture data models.

This section presents a UML model for the input and outputs of the Motion Video Change Detection Architecture.

6.2.1 Motion Video Change Detection Inputs

The UML class diagram in Figure 4 represents a standard WPS input schema and depicts the data objects that comprise the inputs of the Motion Video Change Detection concept presented in this document. Within this testbed, all video streams are being read through a Sensor Observation Service. However, the data objects shown in the UML below are defined as a URI for both historical and current video. This allows a WPS to handle video from various types of sources.



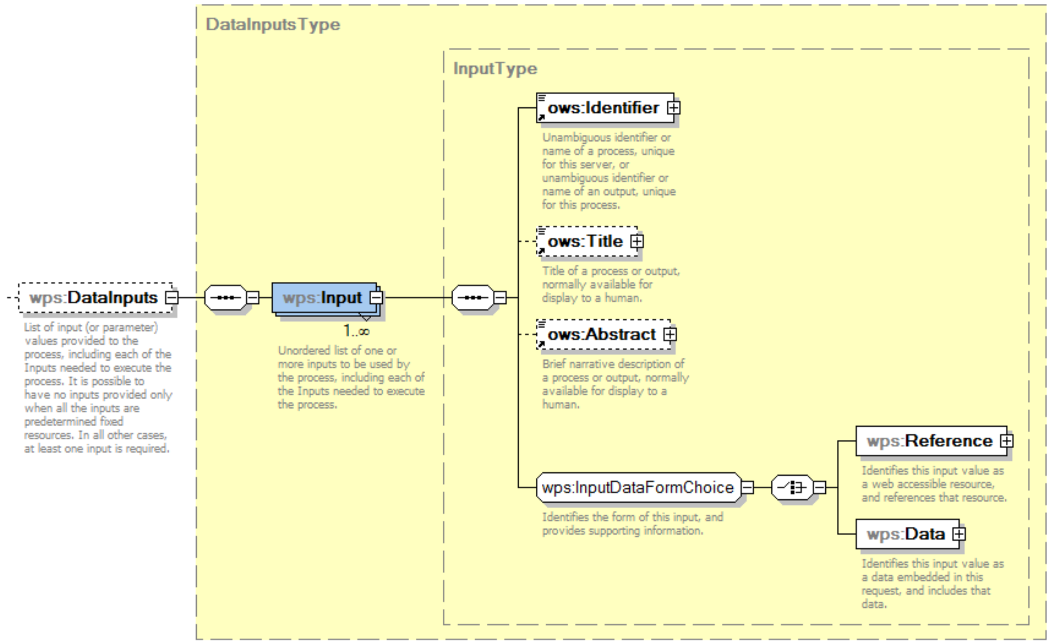


Figure 4 — Standard WPS Input Schema Diagram

Two inputs are required for the invocation of the change detection WPS process. They are (1) a historical video and (2) a current (or live) video. For the OWS-7 testbed, both of these inputs are served by an SOS. The schema is further represented in the example instance below:

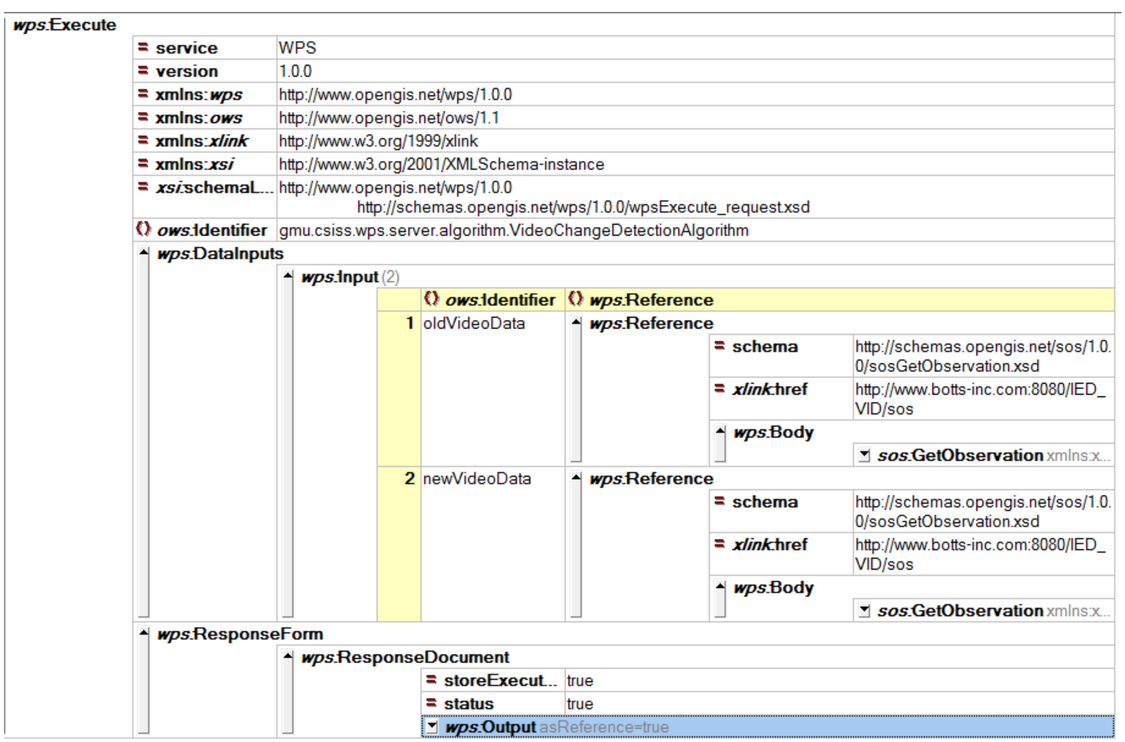


Figure 5 — Motion Video Change Detection Input Instance Example

6.2.2 Motion Video Change Detection Outputs

The UML class diagram in Figure 6 represents a standard WPS output schema and depicts the data objects that comprise the outputs of the Motion Video Change Detection concept presented in this document. The output will simply contain a status along with a URI from which to get a continuing status as well as a URI to an xml document containing the information needed to access the SOS containing the changed frames. The SOS containing the output frames will contain a series of changed frames showing the differences found and associated through a time element to the current (or live) video – this will not be a full video stream.

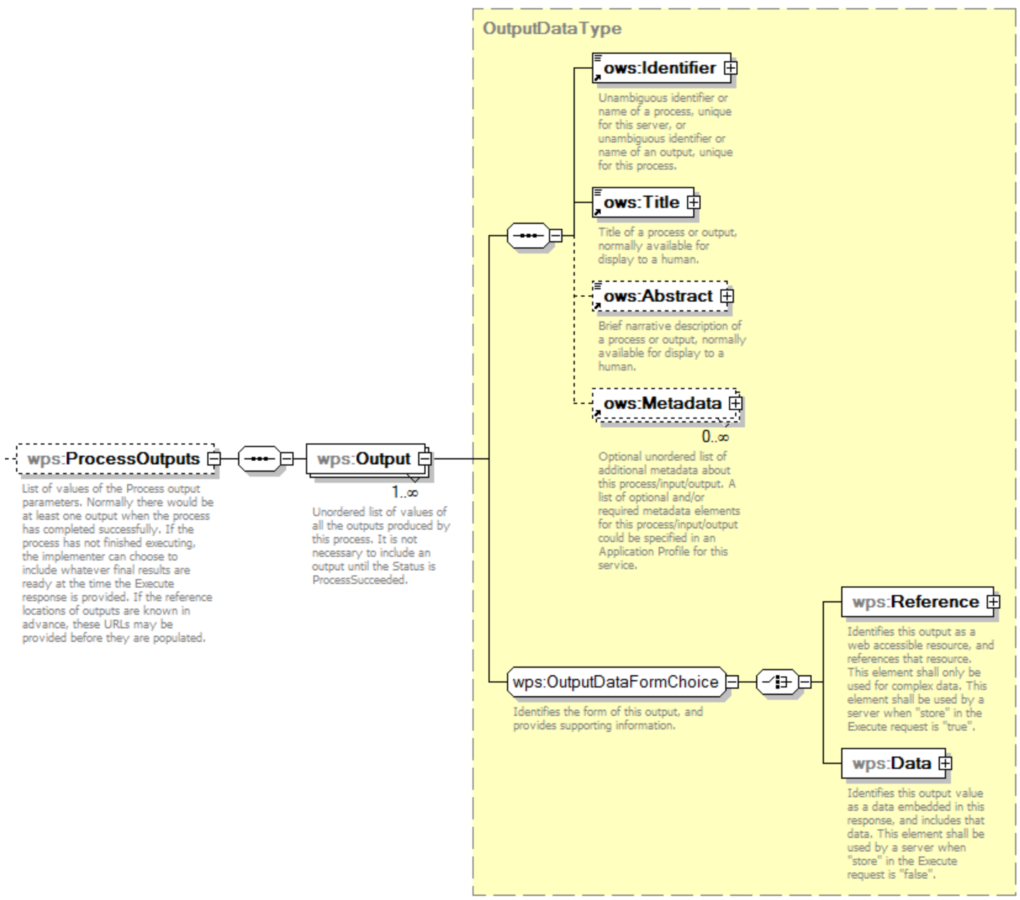


Figure 6 — Motion Video Change Detection Outputs Schema Diagram

The output response of the change detection WPS contains one WPS output element that refers to a URI in which the status can be obtained (in the statusLocation attribute of the ExecuteResponse element) and a URI to an xml document with SOS connection information (in the /ExecuteResponse/ Process/Metadata@href attribute). The schema is further represented in the example instance below:

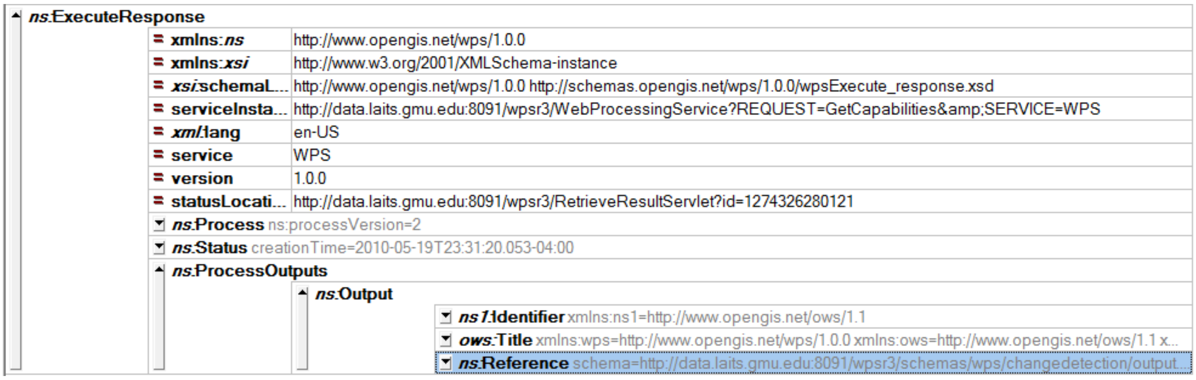


Figure 7 — Motion Video Change Detection Output Instance Example

The reference URI returned by the WPS at the /ExecuteResponse/ Process/Metadata@href attribute provides a location from which the client can retrieve the SOS endpoint and all information needed to retrieve the changed video frames. The schema for the response from this URI is shown in Figure 8 below.

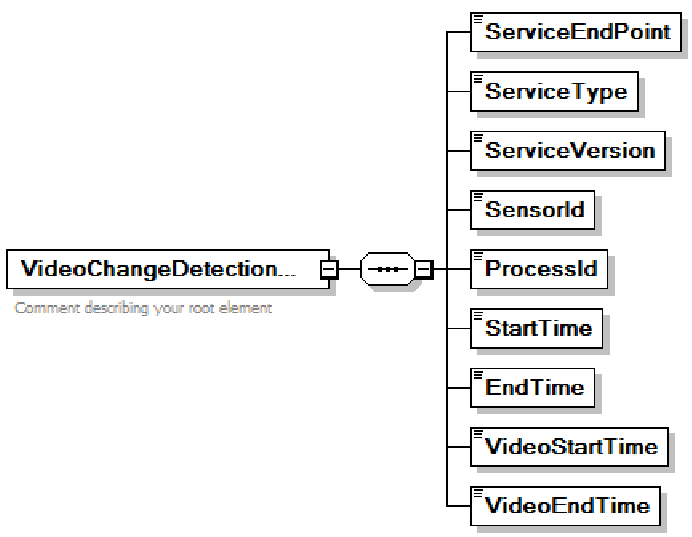


Figure 8 — Motion Video Change Detection Status Schema

Although processing may not yet be complete, a changed frame can be accessed immediately through a GetObservation request after receiving the initial response from the WPS. However, all fields will not be populated in the response. But when change detection has completed the comparison of the streams, all elements will be populated including the end time. As new change frames are inserted into the transactional SOS, the VideoEndTime will be updated. This change in value can be used as a quick method by the client to determine if additional changed frames have been inserted since the last check of the status.

6.3 Motion Video Change Detection Computational Viewpoint

The Computational Viewpoint is concerned with the functional decomposition of the system into a set of distinct components that interact at interfaces.

For OGC, such components are generally realized as web services. The web services of principle concern in this exploration of the Motion Video Change Detection architecture are the Web Processing Service, Sensor Observation Services, and the SFE Integrated Client.

Figure 9 is a wiring diagram showing how the Motion Video Change Detection WPS fits into the overall SFE architecture:

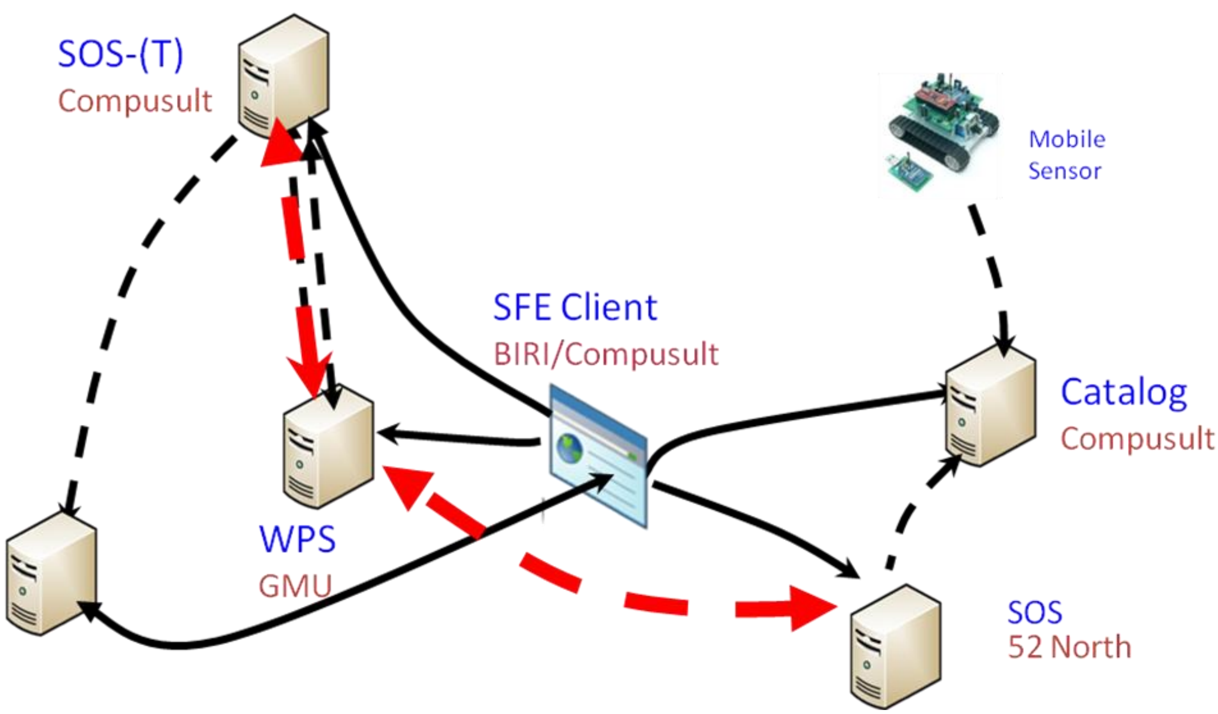


Figure 9 — Wiring Diagram of overall Motion Video Change Detection Activity

6.3.1 Web Processing Service

For the OWS-7 test bed, George Mason University deployed a Web Processing Service (WPS) that takes two input video streams (archived and/or live), performs the change detection algorithm, and returns a status URL to the client through a standard WPS output response. This status URL can then be polled to receive intermediate progress as well as a final status. Each changed frame of the video will be pushed into the SOS-T and can be accessed immediately through a GetObservaion request to endpoint provided in the xml document found at the /ExecuteResponse/ Process/Metadata@href attribute within the WPS status responses. When the streams are completely processed, the final status will be shown in the status response as well as the complete video timeframe will be available in the xml document containing the SOS information.

6.3.2 Sensor Observation Service

BIRI (using 52 North) deployed a Sensor Observation Services to provide video streaming services for the SFE thread of the OWS-7 testbed. This Engineering Report will not detail this effort since the focus is on the Web Processing Service itself. The request to the SOS to retrieve the two video streams serves as the inputs to the Motion Video Change Detection WPS.

Compusult also provided, for the SFE thread of OWS-7, a Sensor Observation Service (Transactional) in which the frames containing changes could be inserted. This service does play an integral part in the Change Detection algorithm in that it allows changed frames to be inserted into the SOS and are immediately made available without the need for the user to wait till the processing of the entire video is completed.

6.3.3 SFE Integrated Client

An Integrated Client was provided by BIRI to be used to drive the Change Detection process for OWS-7. The client plays a vital role in the Change Detection process in that it is responsible for visualizing the process to the user. It is responsible for obtaining the appropriate input video streams and ensuring the two streams overlap in track location and direction. Once the WPS process begins, the client will poll for status and begin retrieving the changed frames from the SOS-T. The client is then responsible for rendering the changed frames such that the user will not only see the changes but visualize them in the context of geographical location and time within the video. More details about the client are provided in section 6.4.6.

6.4 Motion Video Change Detection Engineering Viewpoint

The Engineering Viewpoint describes specific components linked by a communications network and is concerned with the interaction of distinct computational objects.

In the OWS-7 test bed, these are the components that played a role in the Motion Video Change Detection architecture experiment:

- A Sensor Observation Service 1.0.0, provided by Compusult
- A Sensor Observation Service 1.0.0, provided by 52 North

- A Web Processing Service (WPS 1.0.0), provided by George Mason University
- An Integrated Client, provided by BIRA

In the scope of this document, only the interfaces directly related to the WPS will be detailed. **Error! Reference source not found.**10 depicts the high-level interaction between the Integrated Client, the Change Detection WPS, and the Sensor Observation Services.

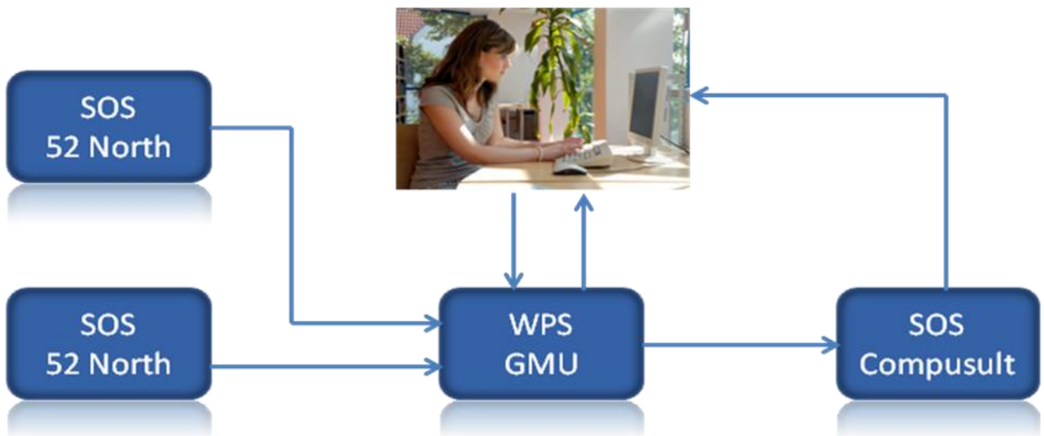


Figure 10 — High-Level Interaction between Major Components

6.4.1 GetCapabilities Sample Request/Response

The Motion Video Change Detection WPS will accept standard GetCapabilities request according to the WPS standard:

http://servername/wps/WebProcessingService?request=GetCapabilities&service=WPS&version=1.0.0

A typical response would be as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-US"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://servername/wpsr3/schemas/wps/1.0.0/wpsGetCapabilities_response.xsd"
updateSequence="1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
+ <ows:ServiceIdentification>
+ <ows:ServiceProvider>
<ows:OperationsMetadata>
<ows:Operation name="GetCapabilities">
<ows:DCP>
<ows:HTTP>
 <ows:Get xlink:href="http://servername/wpsr3/WebProcessingService" />
 </ows:HTTP>
 </ows:DCP>
 </ows:Operation>
<ows:Operation name="DescribeProcess">
<ows:DCP>
<ows:HTTP>
 <ows:Get xlink:href="http:// servername /wpsr3/WebProcessingService" />
 </ows:HTTP>
 </ows:DCP>
 </ows:Operation>
<ows:Operation name="Execute">
<ows:DCP>
<ows:HTTP>
 <ows:Get xlink:href="http:// servername /wpsr3/WebProcessingService" />
 <ows:Post xlink:href="http:// servername /wpsr3/WebProcessingService" />
 </ows:HTTP>
 </ows:DCP>
 </ows:Operation>
 </ows:OperationsMetadata>
<wps:ProcessOfferings>
<wps:Process wps:processVersion="1.0.0">
 <ows:Identifier>changeDetection</ows:Identifier>
 <ows:Title>changeDetection</ows:Title>
 </wps:Process>
 </wps:ProcessOfferings>
<wps:ProcessOfferings>
<wps:Process wps:processVersion="2">

<ows:Identifier>gmu.csiss.wps.server.algorithm.VideoChangeDetectionAlgorithm</ows:
Identifier>
 <ows:Title>detect the change of two videos.</ows:Title>
 </wps:Process>
 </wps:ProcessOfferings>
<wps:Languages>
<wps:Default>
 <ows:Language>en-US</ows:Language>
 </wps:Default>
<wps:Supported>
```

```
 <ows:Language>en-US</ows:Language>
 </wps:Supported>
 </wps:Languages>
 </wps:Capabilities>
```


6.4.2 DescribeProcess Sample Request/Response

The Motion Video Change Detection WPS will accept standard DescribeProcess request according to the WPS standard:

http://servername/wps/WebProcessingService?request=DescribeProcess&service=WPS&version=1.0.0&Identifier=gmu.csiss.wps.server.algorithm.VideoChangeDetectionAlgorithm

A typical response would be as follows:


```
<?xml version="1.0" encoding="UTF-8" ?>
<ns:ProcessDescriptions 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/wpsDescribeProcess_response.xsd" xml:lang="en-
US" service="WPS" version="1.0.0">
<ProcessDescription 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" wps:processVersion="2"
statusSupported="true" storeSupported="true">

<ows:Identifier>gmu.csiss.wps.server.algorithm.VideoChangeDetectionAlgorithm</ows:
Identifier>
 <ows:Title>detect the change of two videos.</ows:Title>
 <ows:Abstract>detect the change of two videos.</ows:Abstract>
 <ows:Metadata xlink:title="video" />
 <ows:Metadata xlink:title="change dectection" />
<DataInputs>
<Input minOccurs="1" maxOccurs="1">
 <ows:Identifier>oldVideoData</ows:Identifier>
 <ows:Title>oldVideoData</ows:Title>
 <ows:Abstract>oldVideoData</ows:Abstract>
<ComplexData>
<Default>
<Format>
 <MimeType>text/XML</MimeType>
 <Schema>http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd</Schema>
 </Format>
 </Default>
<Supported>
<Format>
 <MimeType>text/XML</MimeType>
```

```
 <Schema>http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd</Schema>
 </Format>
 </Supported>
 </ComplexData>
 </Input>
<Input minOccurs="1" maxOccurs="1">
 <ows:Identifier>newVideoData</ows:Identifier>
 <ows:Title>newVideoData</ows:Title>
 <ows:Abstract>newVideoData</ows:Abstract>
<ComplexData>
<Default>
<Format>
 <MimeType>text/XML</MimeType>
 <Schema>http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd</Schema>
 </Format>
 </Default>
<Supported>
<Format>
 <MimeType>text/XML</MimeType>
 <Schema>http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd</Schema>
 </Format>
 </Supported>
 </ComplexData>
 </Input>
 </DataInputs>
<ProcessOutputs>
<Output>
 <ows:Identifier>changeVideo</ows:Identifier>
 <ows:Title>changeVideo</ows:Title>
 <ows:Abstract>changeVideo.</ows:Abstract>
<ComplexOutput>
<Default>
<Format>
 <MimeType>text/XML</MimeType>

<Schema>http://ws.laits.gmu.edu:8080/wpsr3/schemas/wps/changedetection/output.xsd<
/Schema>
 </Format>
 </Default>
<Supported>
<Format>
 <MimeType>text/XML</MimeType>

<Schema>http://ws.laits.gmu.edu:8080/wpsr3/schemas/wps/changedetection/output.xsd<
/Schema>
 </Format>
 </Supported>
```



```
 </ComplexOutput>
 </Output>
 </ProcessOutputs>
 </ProcessDescription>
 </ns:ProcessDescriptions>
```


6.4.3 Execution Sample Request

The following shows a sample request sent to the Motion Video Change Detection WPS by the client application:


```
<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/1.0.0http://schemas.opengis.net/wps/1
.0.0/wpsExecute_request.xsd ">

<ows:Identifier>gmu.csiss.wps.server.algorithm.VideoChangeDetectionAlgorithm</ows:
Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>oldVideoData</ows:Identifier>
      <wps:Reference
schema="http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
xlink:href="http://www.botts-inc.com:8080/IED_VID/sos">
        <wps:Body>
          <sos:GetObservation xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:om="http://www.opengis.net/om/1.0" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml" service="SOS" version="1.0.0">
            <sos:offering>axiscam.video</sos:offering>
            <sos:eventTime>
              <ogc:TM_During>
                <ogc:PropertyName>samplingTime</ogc:PropertyName>
                <gml:TimePeriod>
                  <gml:beginPosition>2010-04-13T17:20:58.000-
05:00</gml:beginPosition>
                  <gml:endPosition>2010-04-13T17:20:59.000-05:00</gml:endPosition>
                </gml:TimePeriod>
              </ogc:TM_During>
            </sos:eventTime>
            <sos:procedure>urn:ogc:object:sensor:BOTTS-
INC:bottsCam0</sos:procedure>
```



```
<sos:observedProperty>urn:ogc:def:property:OGC:radiance</sos:observedProperty>
            <sos:responseFormat>text/xml; subtype="om/1.0.0"</sos:responseFormat>
          </sos:GetObservation>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>newVideoData</ows:Identifier>
      <wps:Reference
schema="http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
xlink:href="http://www.botts-inc.com:8080/IED_VID/sos">
        <wps:Body>
          <sos:GetObservation xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:om="http://www.opengis.net/om/1.0" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml" service="SOS" version="1.0.0">
            <sos:offering>axiscam.video</sos:offering>
            <sos:eventTime>
              <ogc:TM_During>
                <ogc:PropertyName>samplingTime</ogc:PropertyName>
                <gml:TimePeriod>
                  <gml:beginPosition>2010-04-14T17:12:26.300-
05:00</gml:beginPosition>
                  <gml:endPosition>2010-04-14T17:12:28.000-05:00</gml:endPosition>
                </gml:TimePeriod>
              </ogc:TM_During>
            </sos:eventTime>
            <sos:procedure>urn:ogc:object:sensor:BOTTS-
INC:bottsCam0</sos:procedure>

<sos:observedProperty>urn:ogc:def:property:OGC:radiance</sos:observedProperty>
            <sos:responseFormat>text/xml; subtype="om/1.0.0"</sos:responseFormat>
          </sos:GetObservation>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="true" status="true">
      <wps:Output asReference="true">
        <ows:Identifier>changeVideo</ows:Identifier>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```


6.4.4 Execution Sample Response

The following shows a sample response from to the Motion Video Change Detection WPS. This response is returned immediately after the process execution is requested.


```xml
<?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://servername/wpsr3/WebProcessingService?REQUEST=GetCapab
ilities&amp;SERVICE=WPS" xml:lang="en-US" service="WPS" version="1.0.0"
statusLocation="http://servername/wpsr3/RetrieveResultServlet?id=1274841638790">
  <ns:Process ns:processVersion="2">
    <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">gmu.csiss.wps.server.algorithm.VideoCha
ngeDetectionAlgorithm</ns1:Identifier>
    <ows:Title 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">detect the change of two
videos.</ows:Title>
    <ns1:Metadata xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:xlin="http://www.w3.org/1999/xlink"
xlin:type="http://servername/wpsr3/schemas/wps/changedetection/output.xsd"
xlin:role="urn:ogc:def:procedure:GMU::WPSRESULT"
xlin:href="http://servername/wpsr3/Databases/FlatFile/1274841638790changeVideoresul
t.XML"/>
  </ns:Process>
  <ns:Status creationTime="2010-05-25T22:40:38.783-04:00">
    <ns:ProcessStarted percentCompleted="33"/>
  </ns:Status>
</ns:ExecuteResponse>
```


6.4.5 Status Response

Once the above response is received, the client can continually request status through the link provided in the statusLocation attribute from the initial Execute response. This status response will also contain the same statusLocation attribute for continuing to obtain status. In addition, it will include a reference to an xml file from which information can be obtained to retrieve the changed frames from the SOS. The following status requests will be responded to with a response similar to the following:


```xml
<?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://servername/wpsr3/WebProcessingService?REQUEST=GetCapab
```



```
ilities&amp;SERVICE=WPS" xml:lang="en-US" service="WPS" version="1.0.0"
statusLocation="http://servername/wpsr3/RetrieveResultServlet?id=1274841638790">
  <ns:Process ns:processVersion="2">
    <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">gmu.csiss.wps.server.algorithm.VideoCha
ngeDetectionAlgorithm</ns1:Identifier>
    <ows:Title 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">detect the change of two
videos.</ows:Title>
    <ns1:Metadata xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:xlin="http://www.w3.org/1999/xlink"
xlin:type="http://servername/wpsr3/schemas/wps/changedetection/output.xsd"
xlin:role="urn:ogc:def:procedure:GMU::WPSRESULT"
xlin:href="http://servername/wpsr3/Databases/FlatFile/1274841638790changeVideoresul
t.XML"/>
  </ns:Process>
  <ns:Status creationTime="2010-05-25T22:40:38.783-04:00">
    <ns:ProcessSucceeded>Process successful</ns:ProcessSucceeded>
  </ns:Status>
  <ns:ProcessOutputs>
    <ns:Output>
      <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">changeVideo</ns1:Identifier>
      <ows:Title 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">changeVideo</ows:Title>
      <ns:Reference
schema="http://servername/wpsr3/schemas/wps/changedetection/output.xsd"
mimeType="text/XML"
href="http://servername/wpsr3/RetrieveResultServlet?id=1274841638790changeVideo"/
>
    </ns:Output>
  </ns:ProcessOutputs>
</ns:ExecuteResponse>
```


6.4.6 SOS Content Response

When the initial WPS response or the Status response is received, a URL will be provided in the /ExecuteResponse/ Process/Metadata@href attribute that can be used to poll the service for information as to how to retrieve the updated video frames from the SOS. Prior to any frames being processed, the referenced xml document will appear as follows:


```
<?xml version="1.0" encoding="UTF-8"?>
<VideoChangeDetectionOutput xmlns="http://servername/wps/changedetection/output"
xsi:schemaLocation="http://servername/wps/changedetection/output
```



```
http://servername/wpsr3/schemas/wps/changedetection/output.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<ServiceEndPoint>http://sos_servername/SOS_WPS/GetObservation</ServiceEndPoint
>
  <ServiceType>SOS</ServiceType>
  <ServiceVersion>1.0.1</ServiceVersion>
  <SensorId>urn:ogc:def:procedure:GMU::WPS</SensorId>
  <ProcessId>1274841638790</ProcessId>
  <StartTime/>
  <EndTime/>
  <VideoStartTime/>
  <VideoEndTime/>
</VideoChangeDetectionOutput>
```

Note that several of the elements have yet to be populated. After at least one frame has been processed, the SOS Content document will return a response similar to the following xml segment:

```
<?xml version="1.0" encoding="UTF-8"?>
<VideoChangeDetectionOutput xmlns="http://servername/wps/changedetection/output"
xsi:schemaLocation="http://servername/wps/changedetection/output
http://servername/wpsr3/schemas/wps/changedetection/output.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<ServiceEndPoint>http://sos_servername/SOS_WPS/GetObservation</ServiceEndPoint
>
  <ServiceType>SOS</ServiceType>
  <ServiceVersion>1.0.1</ServiceVersion>
  <SensorId>urn:ogc:def:procedure:GMU::WPS</SensorId>
  <ProcessId>1274841638790</ProcessId>
  <StartTime>2010-05-26 00:28:01.0</StartTime>
  <EndTime/>
  <VideoStartTime>2010-04-14T17:12:27.000-05:00</VideoStartTime>
  <VideoEndTime>2010-04-14T17:12:28.000-05:00</VideoEndTime>
</VideoChangeDetectionOutput>
```

Again, note that additional information such as the StartTime, VideoStartTime, and VideoEndTime have now been populated. At this point, any changes can be accessed through the provided SOS. As changed frames are inserted into the SOS-T, the VideoEndTime will be updated. This allows the client to easily determine if additional changed frames have been inserted since the last status check. After all frames have been processed, the final SOS Content document will be returned as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<VideoChangeDetectionOutput
xmlns="http://csiss.gmu.edu/wps/changedetection/output"
```



```
xsi:schemaLocation="http://csiss.gmu.edu/wps/changedetection/output
http://data.laits.gmu.edu:8091/wpsr3/schemas/wps/changedetection/output.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ServiceEndPoint>http://ows-
7.compusult.net/SOS_WPS/GetObservation</ServiceEndPoint>
  <ServiceType>SOS</ServiceType>
  <ServiceVersion>1.0.1</ServiceVersion>
  <SensorId>urn:ogc:def:procedure:GMU::WPS</SensorId>
  <ProcessId>1274841638790</ProcessId>
  <StartTime>2010-05-26 00:28:01.0</StartTime>
  <EndTime>2010-05-26 00:53:04.0</EndTime>
  <VideoStartTime>2010-04-14T17:12:27.000-05:00</VideoStartTime>
  <VideoEndTime>2010-04-14T17:12:31.000-05:00</VideoEndTime>
</VideoChangeDetectionOutput>
```


6.4.7 Presentation of Motion Video Change Detection in the SFE Client

This section presents the handling of Motion Video Change Detection in the Integrated Client provided by BIRI.

Once the client has been alerted by a Notification Service to the fact that an existing video track exists over the current path, it retrieves the two video streams from the appropriate SOS services as seen below:

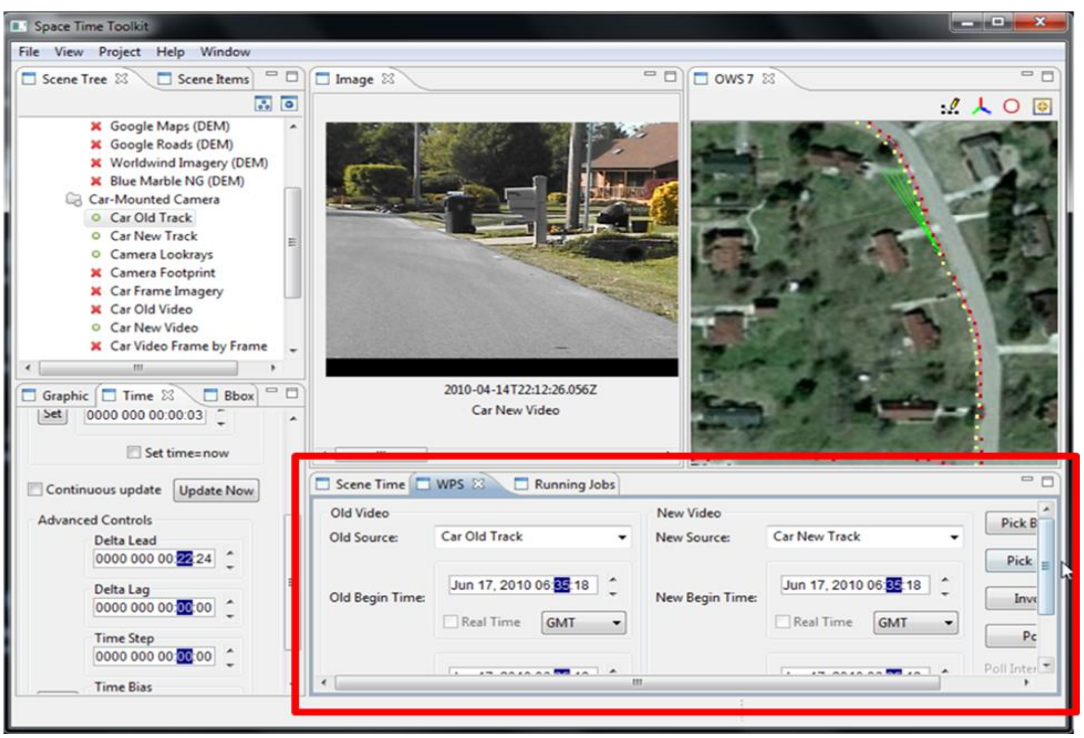


Figure 11 — Client Interface for Retrieving Video Streams

After selecting the two video streams, the user will select the Pick Begin Point button and then select the appropriate location on the track within the map view. This process is then repeated for the ending point.

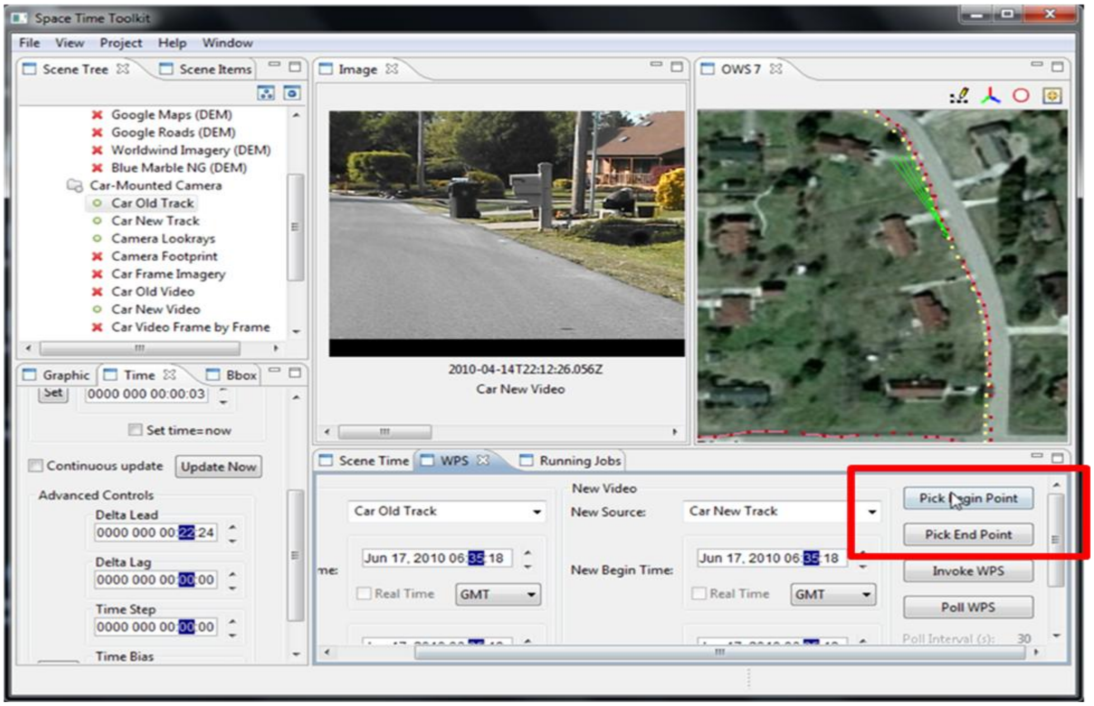


Figure 12 — Client Interface for Picking Start and End Points

After the begin and end points are selected, the user selects the Invoke WPS button to initiate the Change Detection WPS. An immediate response from the WPS is received and a dialog displays the location of the WPS where the status can be checked.

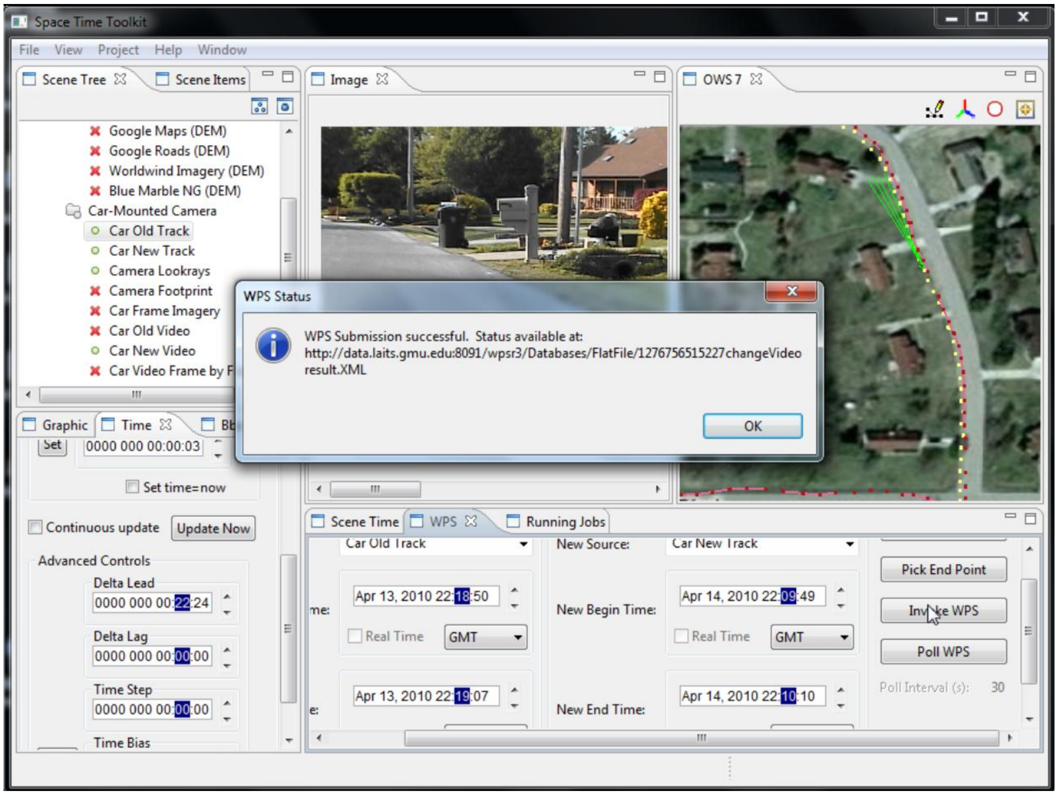


Figure 13 — Client Dialog Showing WPS Submission

The user will then poll the WPS for status. Once complete, another dialog will be displayed show the completion of the process and the Video Start and End Times.

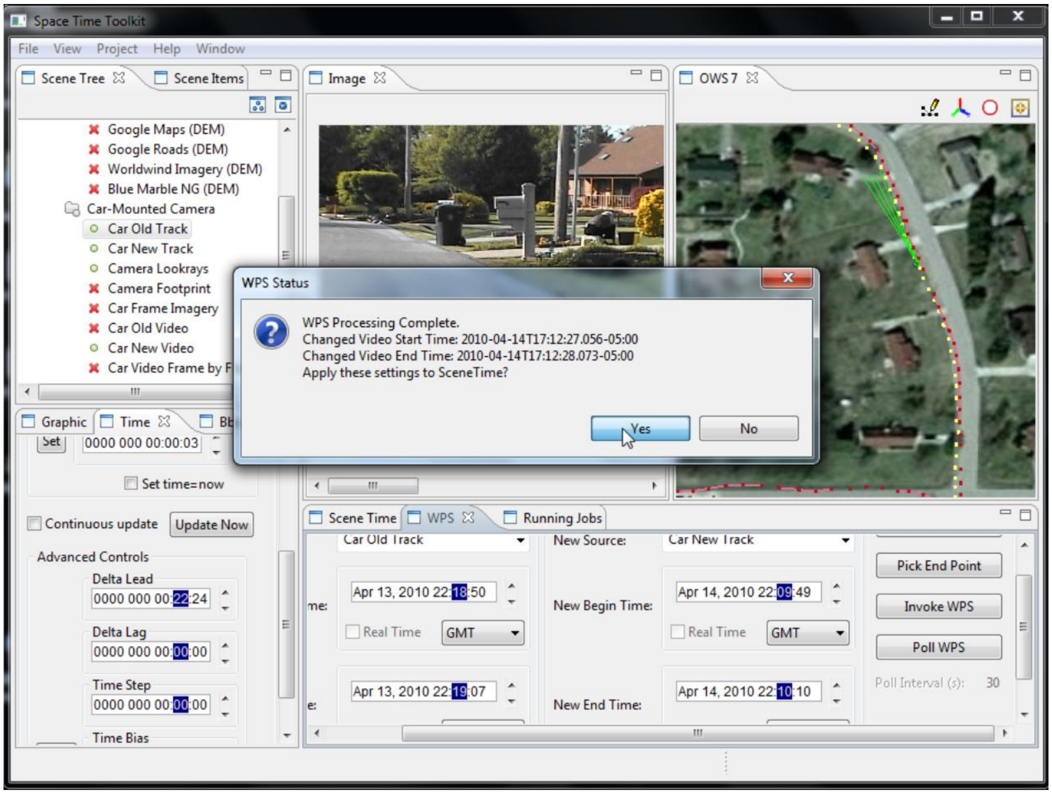


Figure 14 — Client Dialog Showing WPS Completion

Once the Change Detection WPS finds changes, it inserts changed frames that highlight the changed pixels into an SOS. The client will then pull those changed frames for display.

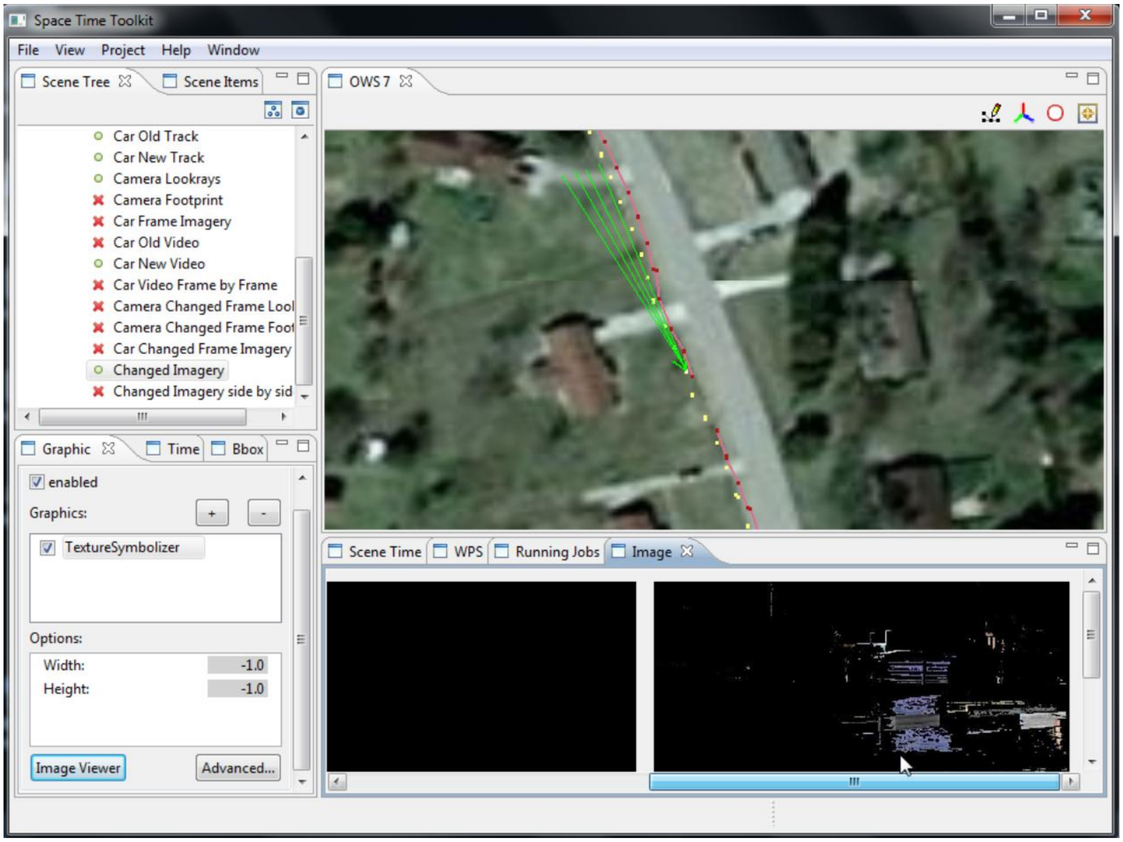


Figure 15 — Client Display Showing a Single Changed Frame

In the example above, this changed frame is shown as a separate image but ultimately we would like to see the changed frame overlaid on the live video stream.