Open Geospatial Consortium Inc.

Date: 2006-02-15

Reference number of this OGC® document: **OGC 05-115**

Version: 0.0.9

Category: Discussion Paper

Editor: Joe Lewis

GeoVideo Web Service

Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC Discussion Paper
Document subtype: Draft Implementation Specification
Document stage: Approved
Document language: English

Contents

Figures Page

Tables Page

Preface

This document defines a Web Services interface for a GeoVideo Web Service. This document is an Interoperability Program Report from the OGC Web Services Phase 3 (OWS-3) interoperability initiative.

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by OGC portal message, 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.

i. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

Intergraph Corporation, Huntsville, Alabama

ii. Document contributor contact points

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

Contact	Company
Joe Lewis	Intergraph Corporation
Gene Grindstaff	Intergraph Corporation
Sheila Whitaker	Intergraph Corporation

iii. Revision history

Date	Release	Editor	Primary clauses modified	Description
25-oct-2005	0.0.0	Grindstaff	7.4	Added closed caption encoding description and database table description
28-Jan-2006	0.0.9	Carl Reed	various	Fix copyright, minor edits, etc.

iv. Changes to the OpenGIS® Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

v. Future work

Improvements in this document are desirable to provide code examples in both C# and Java, of consuming the GVS web service.

Examples of consuming GVS by sending\receiving explicit SOAP XML requests\responses could also be provided.

The textual data (GPS, IMU, camera field of view) in XML format for the video stream needs to be specified.

Foreword

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

— This document was prepared solely by members of the OGC;

— This document does not replace or modify any other document;

— This document references other industrial standards and OGC standards

Introduction

A GeoVideo Web Service (GVS) is a web service that facilitates the viewing of live and/or archived feeds from video cameras. The feeds may be composed of:

- A video stream
- Textual data in a caption stream (e.g. GPS data, camera states and characteristics, custom XML data, such as SensorML\TML)
- A combination of a video stream and associated textual data

The video streams of the feed may be viewed in the Windows Media Player. The textual data is extracted through scripting events that are generated as the caption stream is processed and displayed by the Windows Media Player.

GeoVideo Service

1 Scope

This OpenGIS® document is applicable to anyone who would want to implement a GeoVideo Web Service, or interface with its operations.

This document describes the operations provided by GeoVideo Web Services (GVS), usage and schema of the database required to store information about its logical components (e.g. feeds and cameras). Supporting documentation include a WSDL document describing the Web interface UML diagrams, and examples of HTTP Get Requests\Responses for consuming GVS web.

Included in the technical description of GVS is the generation and usage of GVS client proxy objects, which simplify the client interface to GVS.

This document also describes a methodology for using GVS as an enabler for real-world video-centric security solutions.

2 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent

amendments to, or revisions of, any of these actions do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 03-008r1, OGC Web Services Common Implementation Specification, October 2003

3 N/A

4 Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the OGC Web Services Common Implementation Specification [OGC 03-008r1] shall apply. In addition, the following terms and definitions apply.

4.1 Stream

The output from a video camera, which may be either video (successive images) or textual (e.g. GPS data)

4.2 Feed

A live or archived stream of video and\or textual data, in a format that allows the feed to be displayed in a media player.

4.3 Web Service

A web server application that exposes operations, which a web client can invoke to perform specific tasks

4.4 SOAP (Simple Object Access Protocol)

The format of XML messages that are sent between the web client and web service during the execution of web service methods

4.5 Media File

A file containing a video stream

4.6 Symbols (and abbreviated terms)

Some frequently used abbreviated terms:

GVS **GeoVideo Web Service**

WSDL **Web Service Definition Language**

OGC **Open GIS Consortium**

UML **Unified Modeling Language**

SOAP **Simple Object Access Protocol**

XML Extended Markup Language

4.7 UML notation

Most diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 05-008].

4.8 Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 04-016r2].

5 Overview of GeoVideo Web Service (GVS)

5.1 The Web Service

GVS is a web service that facilitates the viewing of live and\or archived feeds from video cameras. The feeds may be composed of:

- A video stream
- Textual data in a caption stream (e.g. GPS data, camera states and characteristics, custom XML data, such as SensorML\TML)
- A combination of a video stream and associated textual data

The video streams of the feed may be viewed in the Windows Media Player. The textual data is extracted through scripting events that are generated as the caption stream is processed and displayed by the Windows Media Player.

5.2 GVS components

The main components of GVS are:

- Video cameras that provide video streams and textual data
- Microsoft Window Media Encoder, that:
    - Converts camera video format into MPEG-4
    - Encode textual data as scripts commands in caption stream
    - Either stores encoded files as archives, or sends them to Windows Media Server as live streams
    - Creates a stream of textual data only, where the text contains spatial-temporal data for one or more cameras moving within a specified area of interest

- Microsoft Windows media server, which provides clients with “publishing points” or sources of live streaming media files, i.e. real-time video feeds which are being encoded by Microsoft Window Media Encoder

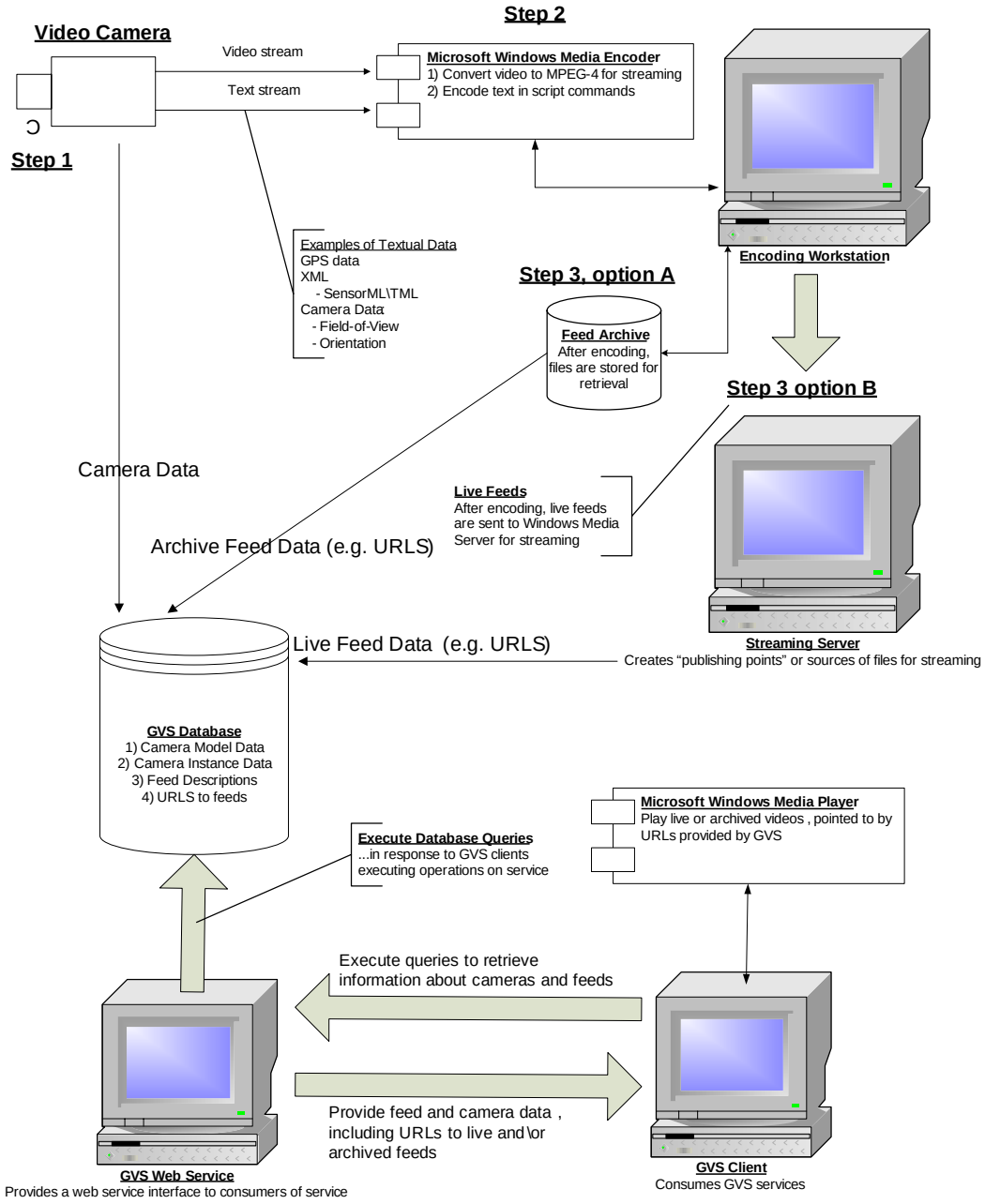
- GVS Database, that contains data about feeds and cameras, that GVS Web Services need to satisfy its clients’ queries, and locate URLs of requested feeds

- GVS Web Services, that provide an interface to its clients for retrieving data from the GVS database

- GVS Client, that queries the GVS Database (via GVS Web Services) to obtain information about cameras and feeds, and URLs for accessing the feeds

- Windows Media Player (embedded in the GVS client) that plays videos and\or displays textual data in the streams from live and\or archived feeds

Figure 1. Overview of GVS



6 The GVS Database

Information about camera types, instances of cameras and their associated feeds are stored in the GVS database. Through the operations exposed by GVS an end-user may retrieve cameras, and\or feeds with properties meeting specific physical, spatial, or temporal data.

An example of such a query would be “return all the archived feeds with the rectangle bounded by 38.01202 N, -71.08812 W and 37.973 N, -71.06 W from cameras with zoom capability, between the 10:30 a.m. and 12:30 p.m. on Aug 3, 2004.

6.1 Summary of GVS queries

Through the GVS interface, a consumer of its services can search for videos that

a) start with a given time range
b) covers a given time range
c) starts before a given time
d) starts after a given time
e) starts with a given time range within a given area of interest (AOI)
f) covers a given time range within a given AOI
g) starts before a given time within a given AOI
h) starts after a given time within a given AOI
i) starts with a given time range within a given AOI
j) starts with a given time range within a given AOI, and has specific feed and\or camera characteristics
k) covers a given time range within a given AOI, and has specific feed and\or camera characteristics
l) starts before a given time within a given AOI, and has specific feed and\or camera characteristics
m) starts after a given time within a given AOI, and has specific feed and\or camera characteristics
n) starts with a given time range within a given AOI, and has specific feed and\or camera characteristics
o) for all cameras within a given AOI, that have specific characteristics, find the videos for any\all of the cameras meeting the criteria (these are the “Sensor in Area”, and “Change Sensor Target” use cases

6.1 List of tables that are used by GVS.

Table 1. Description and Usage of GVS tables

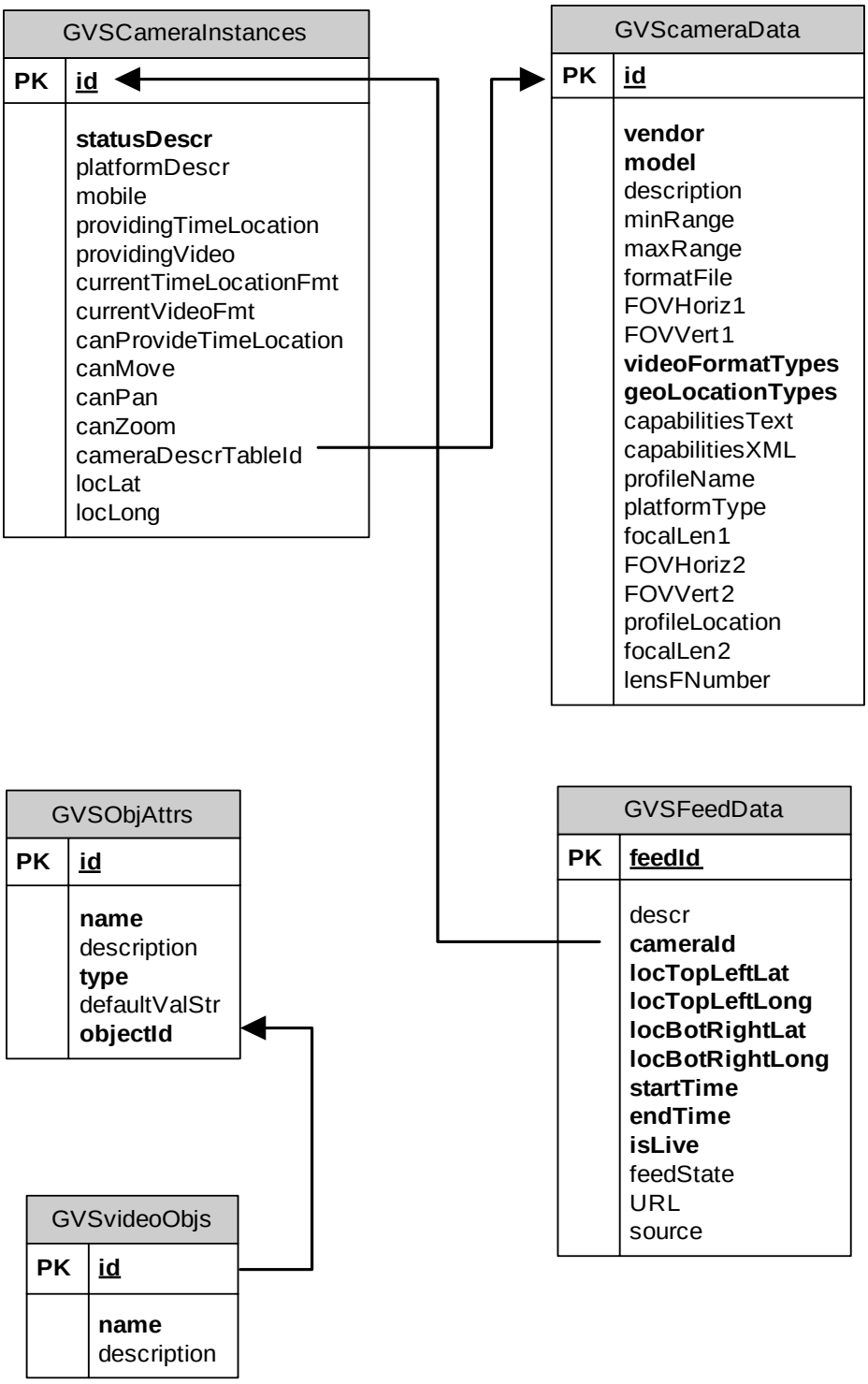
NOTE: table names are in bold-face. **Table.col** represents a column in a table

GVSVideoObjs	The names, descriptions and ID's of all the video objects (e.g. 'cameraInstance', 'feed', 'cameraDescr', stream). The primary key of this table is **id** This table should not be modified, unless new types of objects are added to GVS
GVSAttrs	The queryable attributes of each of the objects in table **GVSVideoObjs** Each object in **GVS VideoObjs** may have 1 or more attributes. All object attributes are in the **GVSAttrs** table, and are related to their objects by **objectid,** which maps to **GVSVideoObjs.id** This table should not be modified, unless new types of objects are added to GVS.
GVSCameraData	Physical characteristics (such as field-of-view) and other data associated with types of cameras. An entry for each type of camera acting as a feed source should be in this table
GVSCameraInstances	Data associated with specific camera instances Each instance of a camera acting as a feed source MUST be added to this table. Each camera instance is related to a record in the **GVSCameraData** table by **cameraDescrTableId,** which maps to **GVSCameraData.id**
GVSFeedData	Data associated with each feed. A record for every feed MUST be added to **GVSFeedData.** Each feed is related to its camera source by **cameraId** which maps to **GVSCameraInstances.id** An exception to the above is where the feed

	contains a caption stream with spatial-temporal data for multiple cameras. In that case, **cameraId** is 0.

All of the above is summarized in **Diagram 2. GVS tables.**

Figure 2. GVS tables.



7 GVS Web Service Interface

7.1 GVS and Client Proxy Classes

Instead of using HTTP GET with key-value pair encoding, clients consume GVS Web Services via the SOAP protocol, encapsulated by client proxy classes. The classes are generated from the WSDL file, by tools specific to the client's development environment.

The client code does not have to include functionality for creating\sending the XML SOAP requests to the server, or requesting\decoding SOAP responses. Instead, the client creates instances of proxy objects, and executes methods on the GVS Server by calling methods on the objects. The proxy objects already encapsulate all of the logic needed to communicate with the server, via SOAP.

All of the proxy objects are described in detail in section **7.3** of this document.

7.2 Operations provided by GVS Web Service

Following is a list of all the functions in the GVS Web Service Interface. Instead of executing these operations via HTTP Get with key-value pair encoding, a consumer of GVS creates an object that encapsulates - or acts as a proxy for – GVS Web Services, then calls methods on that object.

Example (C#)

```
_GVS = new Service();    //creates a GVS service object

 CObjectDescrs[] _objectDescrs = _GVS.GetGVSObjectDescrs();
```

In the code above, after a GVS service object (GVS) is created in the client code, the operation **GetGVSObjectDescrs()** (described below in section **7.2.1**) is executed and returns a list of GVS objects called **CObjectDescrs** (also described in section **7.2.1**).

The GVS objects become defined and usable to client code, after the GVS Web Services WSDL document is processed by the client development environment. In Microsoft Visual Studio for example, the WSDL document is processed by providing a URL to its location, which is then added to the development environment as a "Web Reference". A GVS namespace is then created, which contains all of the GVS objects described in section **7.2.1**).

7.2.1 Definition of GVS Web Service operations

Name: GetCapabilities
Purpose: Return a capabilities XML document as described in OGC Common Web Services specification

ReturnValue: String – the XML instance document

Arguments:

Name	Data Type	Purpose
acceptVersions	string	
Sections	string	
UpateSequence	string	
AcceptFormats	string	

Name: GetGVSObjectDescrs
Purpose: Returns a list of descriptions for all the objects known to GVS. Each object contains a list of attributes, via a property called '_attributes'.
ReturnValue: CObjectDescr[] – an array of **CObjectDescr** objects (see section 7.3)
Arguments: - none

Name: GetFeedsFromIds
Purpose: Returns a list of **CFeed** objects (see section 7.3) from a list of feed ids
ReturnValue: CFeed[] – an array of **CFeed** objects

Arguments:

Name	Data Type	Description
FeedIds	int []	List of feed ids.

Name: GetFeedsFromQuery
Purpose: Returns the list of **CFeed** objects (see section 7.3 which meet the query criteria specified in the **CQueryObj** object.
ReturnValue: CFeed[] – an array of **CFeed** objects

Arguments:

Name	Data Type	Description
queryObject	**CQueryObj**	Object containing criteria for query

Name: GetCamerasFromQuery
Purpose: Returns the list of **CCamera** objects (see section 7.3 which meet the query criteria specified in the **CQueryObj** object.
ReturnValue: CCamera [] – an array of **CCamera** objects

Arguments:

Name	Data Type	Description
queryObject	**CQueryObj**	Object containing criteria for query

7.3 Description of GVS Client Proxy Classes

In this section is a description of all the GVS client proxy objects.

Table 2. Overview of Client Proxy Classes

Object	Purpose
Service	The main GVS object. Operations in the GVS Web Services interface are executed by calling corresponding methods on this proxy object
CLatLong	A location in decimal latitude and longitude coordinates.
CBoundBoxLatLong	a bounding box defined by two points in latitude and longitude coordinates
CSpaceTimeBounds	extents\contraints of space and time, for executing queries
CQueryObj	query criteria for retrieving cameras and feeds based on their characteristics and\or spatial-temporal constraints in the form of CSpaceTimeBounds objects
CObjectDescr	descriptions of a GVS objects (e.g. a feed or camera)
CAttrs	attributes of each GVS object
CCamera	a video camera
CFeed	a feed

Figure 3- UML diagram for main GVS Objects

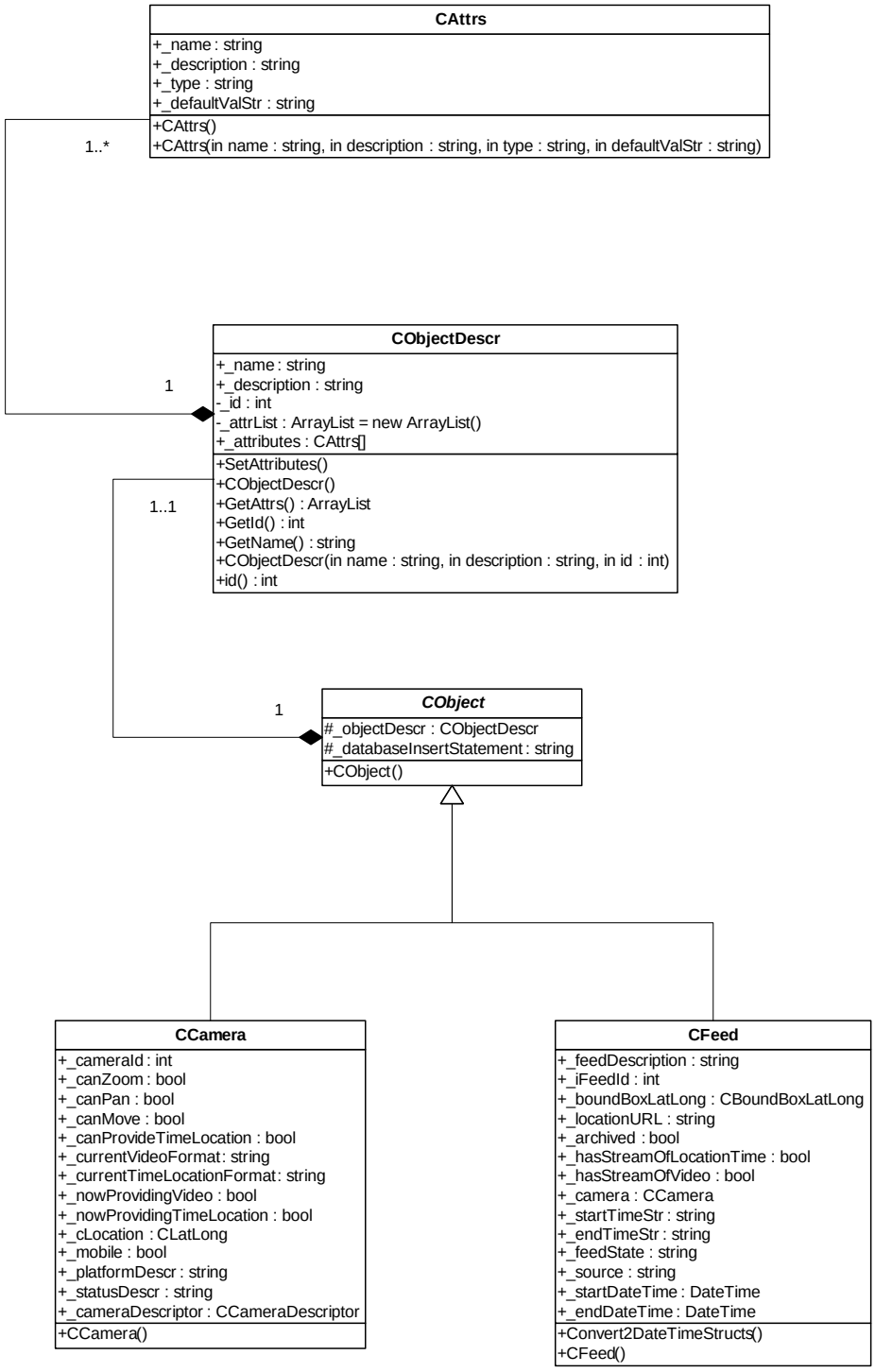


Figure 4 UML Diagram for GVS Query Objects

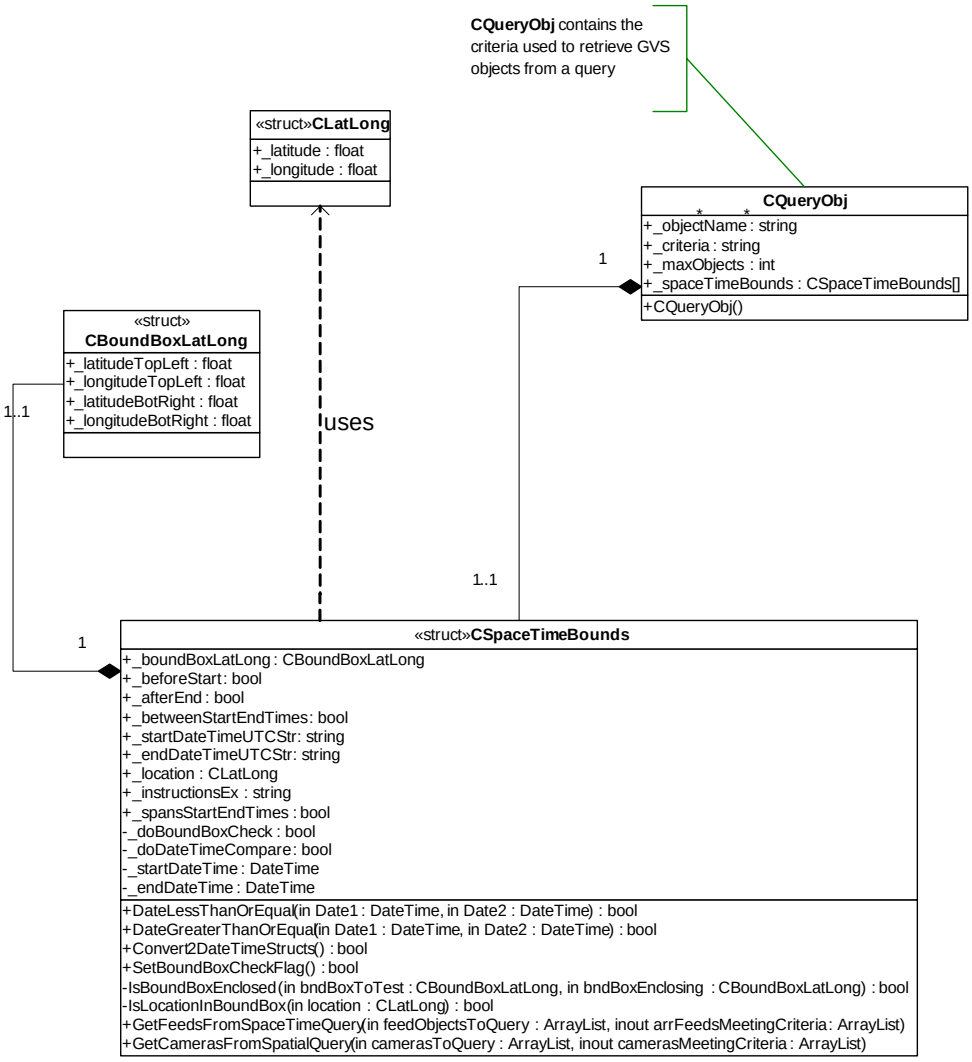
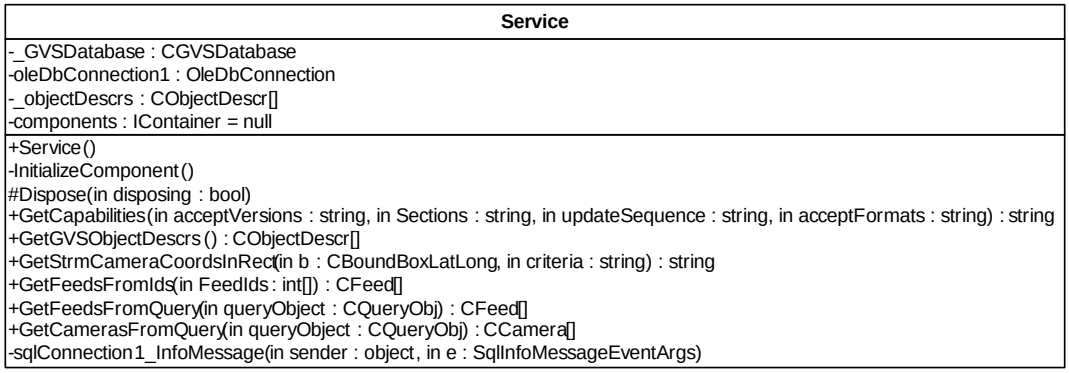


Figure 5 UML Object for GVS Service Object



7.3.1 GVS Client Proxy Class Data Members

Following is a list of all the GVS Client Proxy Classes, along with their data types and descriptions.

Class: **CFeed**

Data Member	Type	Description
_feedDescription:	String	Description of feed
_iFeedId:	int	Feed id
_boundBoxLatLong:	CBoundBoxLatLong (GVS object)	Bounding box object containing spatial extents of feed
_locationURL:	string	URL of feed
_archived:	Boolean	Whether or not feed is archived
_hasStreamOfLocationTime:	Boolean	Feed has textual spatial\temporal data
_hasStreamOfVideo:	Boolean	Feed has video

_camera:	CCamera (GVS object)	The camera object that is the feed source
_startTimeStr:	String	Date\time that feed started
_endTimeStr:	String	Date\time that feed ended
_feedState:	String	Not used
_source:	String	If "boundingBox" then this is a feed containing spatial-temporal data of cameras moving within the area defined by _boundBoxLatLong

Class: **CCamera –** represents a GVS camera

Data Member	**Type**	**Description**
_cameraId	int	Camera id
_canZoom	Boolean	Has zoom capability
_canPan	Boolean	Has pan capability
_canMove	Boolean	Can move (pitch, roll, yaw)
_canProvideTimeLocationt	Boolean	Can provide textual spatial-temporal data
_currentVideoFormat	String	Format of video (e.g. MPEG-2)
_currentTimeLocationFormat	String	Format of spatial-temporal data (e.g. GPS
_nowProvidingVideo	Boolean	Currently providing video feed
_nowProvidingTimeLocation	Boolean	Currently providing spatial-temporal data
_cLocation	**CLatLong**	A location object, containing location of

		camera
_mobile	Boolean	Camera is moving
_platformDescr	String	Description of platform
_statusDescr	String	Status
_cameraDescriptor	**CCameraDescriptor**	Object containing a description of camera

Class: **CAttrs –** attributes of a GVS Object

Data Member	**Type**	**Description**
_name	string	Name of object
_description	string	Description of object
_type	string	Type of object
_defaultValStr	string	Default Value

Class: **CObjectDescr –** describes GVS Objects

Data Member	Type	Description
_name	string	descr
_description	string	descr
_id		descr
_attributes	CAttrs[]	descr

Class: **CCameraDescriptor –** describes GVS Camera Objects

Data Member	Type	Description
_vendor	String	vendor
_model	String	model
_id	int	Unique row id
_description	String	Description of camera
_minRange	double	Minimum range
_maxRange	double	Maximum range
_formatFile	String	descr
_FOVHoriz1	double	Horizontal field-of-view 1
_FOVVert1	double	Vertical field-of-view 1
_videoFormatTypes	String	Video formats supported (colon-delimited)
_geoLocationTypes	String	Textual spatial-temporal data format (e.g. GPS) colon-delimited
_capabilitiesText	String	Additional capabilities text
_capabilitiesXML	String	Capabilities XML

_profileName	String	Name of encoding profile file
_platformType	String	Type of platform
_focalLen1	double	Focal length 1
_FOVHoriz2	double	Horizontal field-of-view 2
_FOVVert2	double	Vertical field-of-view 2
_profileLocation	String	Location of profile file
_focalLen2	double	Focal length 2
_lensFNumber	double	Lens f/number

Struct: **CLatLong –** describes a location in decimal latitude and longitude

Data Member	**Type**	**Description**
_latitude	float	
_longitude	float	

Struct: **CBoundBoxLatLong** describes a bounding-box location in decimal latitude and longitude.

Data Member	**Type**	**Description**
_latitudeTopLeft	float	Latitude at top left corner of bounding box
_longitudeTopLeft	float	Longitude at top left corner of bounding box
_latitudeBotRight	float	Latitude at bottom right corner of bounding box
_longitudeBotRight	float	Longitude at bottom right corner of bounding box

Class: **CQueryObj -** stores query criteria for retrieving data and feeds from GVS database

Data Member	**Type**	**Description**
_objectName	string	Name of object –currently ignore
_criteria	string	String containing 'where clause' used in query, e.g "isLive = 1" to query for live feeds
_maxObjects	int	Currently unused
_spaceTimeBounds	**CSpaceTimeBounds** (object)	Stores spatial-temporal extents of query (see below)

Class: CSpaceTimeBounds – stores spatial-temporal extents for queries

_boundBoxLatLong:	**CBoundBoxLatLong** (object)	Bounding box that query is based on
_beforeStart:	Boolean	Query for date\time before _startDateTimeUTCStr
_afterEnd:	Boolean	Query for date\time after _endDateTimeUTCStr
_betweenStartEndTimes:	Boolean	Query for times between _startDateTimeUTCStr and_endDateTimeUTCStr
_startDateTimeUTCStr:	String	Start date\time that query is based on
_endDateTimeUTCStr:	String	End date\time that query is based on
_location:	**CLatLong** (object)	Store a location
_instructionsEx:	String	Currently unused
_spansStartEndTimes:	Boolean	Query for date\time extent that span

		_startDateTimeUTCStr and_endDateTimeUTCStr
_doBoundBoxCheck:	Boolean	Use bounding box in query
_doDateTimeCompare:	Boolean	Use date\time criteria in query

7.4 GVS Data Encapsulation

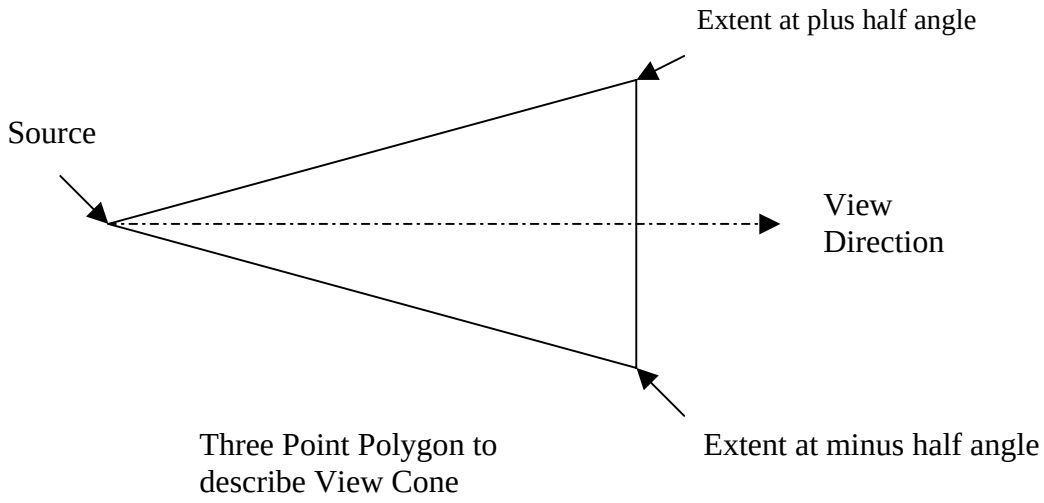
7.4.1 GVS Date Time Location String

The following information must be encoded in the Closed Caption field of the video stream for the GeoVideo Service:

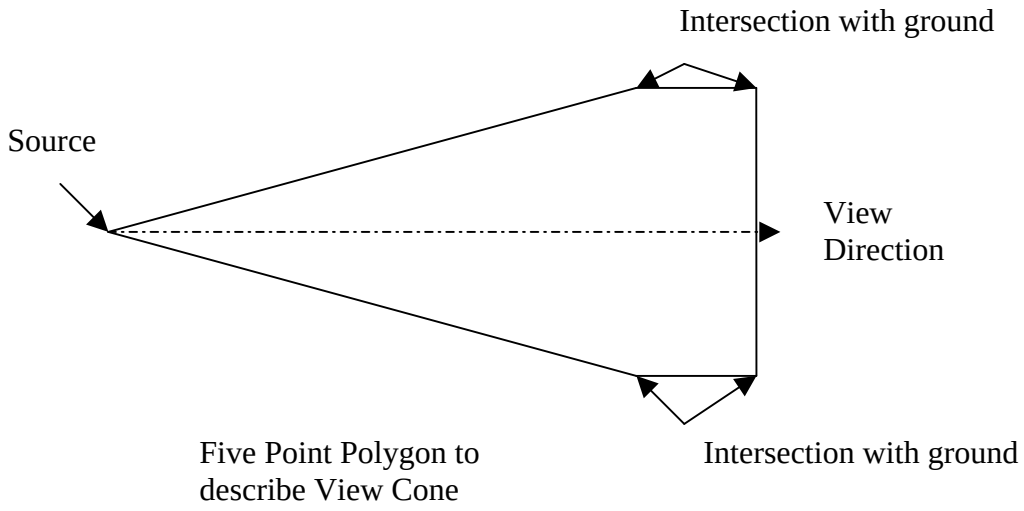
1. Date
2. Time
3. A closed polygon describing either the camera view cone when looking over the horizon, or quadrangle that shows the area of interest when looking down on the ground.

The date shall be encoded in ISO date/time format, and should take the ISO format of year-month-day hour:minute:seconds, or 2005-07-05 13:00:00 zulu.

The view cone shall be a three point or five point closed polygon. The first point shall be the source location of the camera expressed in LAT LON decimal degrees. For a three point polygon where the camera's view cone does not intersect with the ground, the second and third point should represent a point at the camera's view extent with an angle equal to plus or minus half of the camera's view angle from the viewing direction. All points shall be expressed in LAT LON decimal degrees.



If the camera's view cone intersects with the ground, then the first point shall be the source location of the camera expressed in LAT LON decimal degrees. The remaining five points will describe the intersection of the view cone in LAT LON decimal degrees.



If the view extent of the camera is not specified, the default value of 30 meters will be used to calculate the points by the service. If the horizontal view angle is not specified a

default value of 60 degrees will be used. If the vertical view angle is not specified, a default value of 45 degrees will be used. If the camera is mobile, and the viewing direction is not specified, then successive points will be used to calculate the viewing direction. If the camera is not mobile and the viewing direction is not specified, it will default to zero degrees. If the roll and pitch of the camera are not specified, they will default to zero degrees. If the camera's altitude is not specified, then a default value of sea level will be assumed. Since the terrain elevation is not available to the GVS, all terrain is assumed to be at sea level.

Encoding Format for Date Time Location:

The encoding of the data shall be the following simple line of text:

$GVDTL 0, 2005-07-05T13:55:07**,** 3, 34.6996866,-86.6883722**,** 34.6996602,-86.6880433, 34.6994416,-86.6882358, \n

Where the above is:

GeoVideo Date Tile Location Identifier and version, Year-month-day hour:minute:seconds ZULU, number of points, lat, lon, lat, lon, lat, lon…, newline

7.4.2 GVS Raw String

The second line of the encoded string shall be the raw of the GPS, IMU, and camera parameters. It is the information used to calculate the first string and is provided so that the client can do more detailed mapping of the camera to terrain for such purposes as roto-scoping video.

Encoding Format for RAW string:

The encoding of the data shall be the following simple line of text:

$GVRAW 0, 2005-07-05T13:55:07,
34.6996866,-86.6883722,673.5,0.0,-15.0,270.0,60.0,40.0,30.0

This string consists of the following fields:

$GVRAW 0 - A GeoVideo identifier and version number

2005-07-05T13:55:07 - Date and Zulu Time in ISO format

34.6996866,-86.6883722,673.5, - Camera location in lat, lon, elevation in decimal degrees and meters

0.0,-15.0,270.0, - Roll, Pitch, Yaw (Heading) in decimal degrees

60.0,45.0,30.0 - camera horizontal view angle, camera vertical view angle, and field of view in meters.

7.5 GVS Database Schema

The following is a description of the tables used in the GVS database.

7.5.1 GVS GVScameraData Table

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[gvsuser].[GVScameraData]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [gvsuser].[GVScameraData]

GO

CREATE TABLE [gvsuser].[GVScameraData] (
 [vendor] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
 [model] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
 [id] [int] NOT NULL ,
 [description] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [minRange] [float] NULL ,
 [maxRange] [float] NULL ,
 [formatFile] [varchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [FOVHoriz1] [float] NULL ,
 [FOVVert1] [float] NULL ,
 [videoFormatTypes] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
 [geoLocationTypes] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
 [capabilitiesText] [varchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [capabilitiesXML] [varchar] (150) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
 [profileName] [varchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [platformType] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [focalLen1] [float] NULL ,
 [FOVHoriz2] [float] NULL ,
 [FOVVert2] [float] NULL ,
 [profileLocation] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [focalLen2] [float] NULL ,
 [lensFNumber] [float] NULL
) ON [PRIMARY]
GO
```

7.5.2 GVS GVSCameraInstances Table

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[gvsuser].[GVSCameraInstances]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [gvsuser].[GVSCameraInstances]

GO

CREATE TABLE [gvsuser].[GVSCameraInstances] (
 [statusDescr] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
 [platformDescr] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [mobile] [bit] NULL ,
 [providingTimeLocation] [bit] NULL ,
 [providingVideo] [bit] NULL ,
 [currentTimeLocationFmt] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
 [currentVideoFmt] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [canProvideTimeLocation] [bit] NULL ,
 [canMove] [bit] NULL ,
 [canPan] [bit] NULL ,
 [canZoom] [bit] NULL ,
 [id] [int] NOT NULL ,
 [cameraDescrTableId] [int] NULL ,
 [locLat] [float] NULL ,
 [locLong] [float] NULL
) ON [PRIMARY]
GO
```

7.5.3 GVS GVSFeedData Table

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[GVSFeedData]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[GVSFeedData]

GO

CREATE TABLE [dbo].[GVSFeedData] (
 [feedId] [int] IDENTITY (1, 1) NOT NULL ,
 [descr] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [cameraId] [int] NOT NULL ,
 [locTopLeftLat] [float] NOT NULL ,
 [locTopLeftLong] [float] NOT NULL ,
 [locBotRightLat] [float] NOT NULL ,
 [locBotRightLong] [float] NOT NULL ,
 [startTime] [datetime] NOT NULL ,
 [endTime] [datetime] NOT NULL ,
 [isLive] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
 [feedState] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
,
 [URL] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [source] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

7.5.4 GVS GVSObjAttrs Table

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[gvsuser].[GVSObjAttrs]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [gvsuser].[GVSObjAttrs]

GO

CREATE TABLE [gvsuser].[GVSObjAttrs] (
 [name] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
 [description] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [type] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
 [defaultValStr] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [objectId] [int] NOT NULL ,
 [id] [int] NOT NULL
) ON [PRIMARY]
GO
```

7.5.5 GVS GVSvideoObjs Table

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[gvsuser].[GVSvideoObjs]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [gvsuser].[GVSvideoObjs]

GO

CREATE TABLE [gvsuser].[GVSvideoObjs] (
 [name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
 [description] [varchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
 [id] [int] NOT NULL
) ON [PRIMARY]
GO
```

7.6 Implemented Geo Video Service

The direct link to the videos are:

mms://geovideosvr.intergraph.com/SanDiegoCedarFiresAerial

mms://geovideosvr.intergraph.com/SanDiegoCedarFiresCar

The link to the SOAP server for GVS is:

The URL is http://gvs.intergraph.com/GVSBeta/Service.asmx

For the WSDL it is
http://gvs.intergraph.com/GVSBeta/Service.asmx?WSDL