



SPY WP6

State of the art and feasibility analysis

D6.1.2

by
Arc Informatique

Due date of deliverable: t0+ **6**

Actual submission date: t0+ **12**



DOCUMENT HISTORY			
Version	Date	Comments	Author
01.01	02/28/12	First version	Arc Informatique



Surveillance imProved sYstem

	Name and function	Date	Signature
Prepared by	Florent Martin Project manager		
Reviewed by	Alain Bauchart Services & Development manager		
Approved by	Alain Bauchart Services & Development manager		
Authorized by	Alain Bauchart Services & Development manager		



CONTENTS

1. SCOPE 5	
2. ASSOCIATED DOCUMENTS.....	5
2.1 APPLICABLE DOCUMENTS	5
2.2 REFERENCE DOCUMENTS.....	5
3. TERMINOLOGY.....	6
3.1 ABBREVIATIONS	6
3.2 DEFINITIONS	6
4. INTRODUCTION.....	7
5. USER NEEDS AND REQUIREMENTS AND TECHNICAL REQUIREMENTS.....	9
5.1 VIDEO SYSTEM INTEGRATION IN BUILDING MANAGEMENT	9
5.2 VIDEO SYSTEM INTEGRATION TRANSPORTATION.....	11
6. OEM AND SOFTWARE COMPANY SOLUTION.....	13
6.1 MANUFACTURER PROVIDERS.....	13
6.2 VIDEO PLATFORM PROVIDERS.....	19
6.3 CONCLUSION	21
7. APPLICATION IN SPY PROJECT	22



SUMMARY

This document describes the state-of-the-art regarding video integration in SCADA software. It also contains a study of various video systems provided by main camera manufacturers and video software solution providers.



1. SCOPE

In SPY project one of the goals is to study the integration of video in SCADA software. In SPY global architecture, the SCADA is located in the control room and behaves as a client of the video server.

Aim of this document is to describe, based on some real examples, the needs, from a customer point of view in terms of accessing data from the video server and controlling cameras.

There are also some technologies issues, this document, throughout a selection of the main solution providers of the video domains, provides some capabilities of interfacing SCADA and video servers.

2. ASSOCIATED DOCUMENTS

2.1 APPLICABLE DOCUMENTS

- A1 Full Project Proposal – SPY (Surveillance imProved System), Eric Munier, 8th December 2010, version 2.0

2.2 REFERENCE DOCUMENTS

- R1 : Title and reference



3. TERMINOLOGY

3.1 ABBREVIATIONS

N/A	Non Applicable
TBC	To Be Completed
SCADA	Supervision Control And Data Acquisition
BMS	Building Management System

3.2 DEFINITIONS

4. INTRODUCTION

Security systems currently used in building management systems, infrastructures and other vertical market are mainly based on proprietary solutions.

Monitoring and controlling such systems means facing the following issues:

- Connection with many proprietary sub systems
- Interoperability between sub systems needs specific developments
- Global diagnostic is complex
- There is no data aggregation

In many cases, several dedicated supervision systems are installed which is, from an exploitation point of view very difficult to manage.



Fig 1.: Several supervision systems running in parallel

Trend now is to concentrate all the information on a single supervision system acting as a Hypervisor.

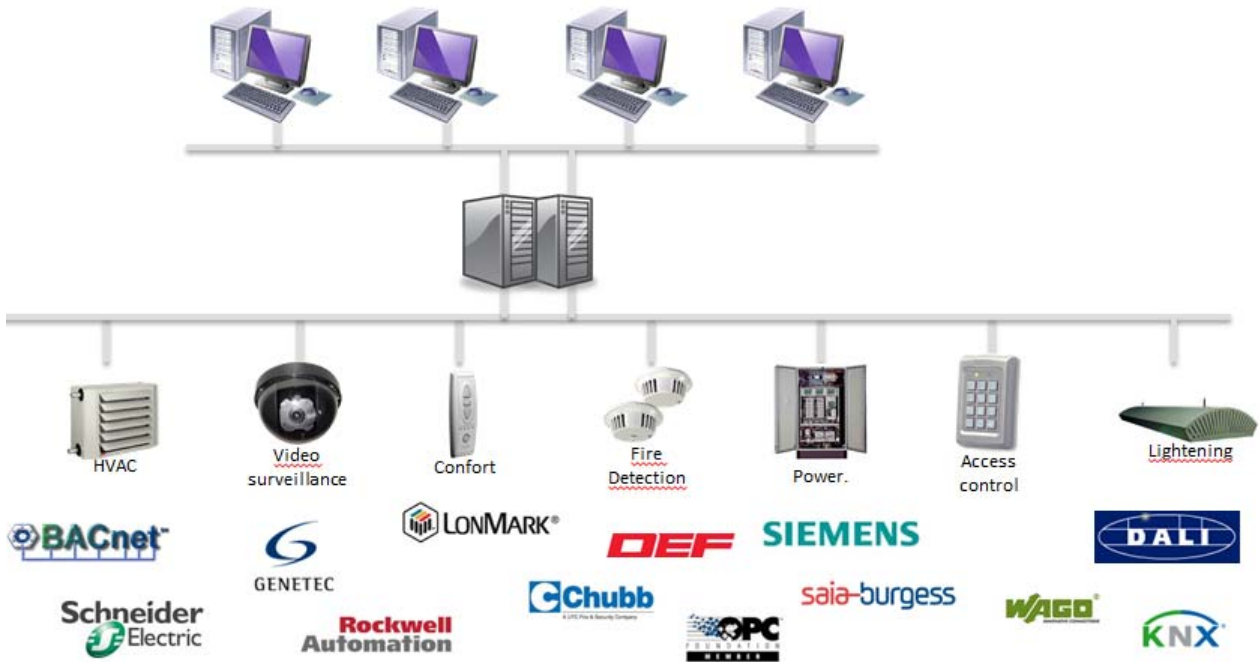


Fig 2: One single Hypervision system concentrating all data

This approach means being able to connect the hypervisor to each sub-system installed on the site.

In the present document, we will concentrate on the connection to the video surveillance sub-system.

5. USER NEEDS AND REQUIREMENTS AND TECHNICAL REQUIREMENTS

As describe, aim is to connect several sub-systems to a single hypervision software. This hypervision system shall fulfil the following requirements:

- Capacity to handle high volume of data (from hundreds of thousands to millions),
- Redundancy features in case of failure and for maintenance periods,
- High availability system,
- Fully customizable user interface,
- Multilinguism for both language and role,
- Data aggregation, consolidation,
- Openness.

5.1 VIDEO SYSTEM INTEGRATION IN BUILDING MANAGEMENT

The following picture is an example of what customer expect as video system integration into their SCADA system in the particular case of Building Management system.



Fig 3: Video integration in Building Management System



Surveillance imProved sYstem

In most case the operator console is a multi-screen hardware. At least a two screens console, one displaying SCADA system the other one the video streaming from camera.

On a typical system, the following user interface will be expected.

Displaying cameras angle:

The need here is that the operator can see on the floor layout the angle which can be seen by the cameras depending on their position. This is very important in particular when camera position can be controlled from the operator console to locate the action which is displayed on the video stream.

Alarm list:

Alarm management is a standard feature of SCADA packages. In the case of interfacing a SCADA with a video system, alarm may come either from the SCADA itself or may be retrieved from the Video server.

In both cases they should be displayed in the same alarm viewer.

Customizable mimic:

A SCADA package allows to draw and animate a fully customizable user interface. When used in BMS, mimics (windows) represent the floor layout on which monitored devices are associated to graphic animated objects. Typically, regarding video, each camera will be represented by an object from which a pop up window can be opened, displaying video stream and allowing the control of the camera.

Video display and camera control:

This screen contains a mosaic, each one containing the video stream from a selected camera. On the right pane, the operator can select and control cameras.

5.2 VIDEO SYSTEM INTEGRATION TRANSPORTATION

The following picture is an example of what customer expect as video system integration into their SCADA system in the particular case of transportation system.



Fig 4: Video integration in Transportation System

As in the previous example (BMS), the operator console allows an access to all data coming from both the SCADA system and the Video Server.

Alarm and Events list:

They contain alarm and event generated by the SCADA from data acquired on field devices but also from the Video server. Note that alarm generated by the video server can be system alarm (e.g. camera failure) or alarm from the video streaming analysis (e.g. passenger on track)

Camera selection:

This panel allows selecting the camera from which the streaming is to be displayed. Note that compared to the previous one in this example only one video is displayed.



Surveillance imProved sYstem

Camera control:

This tool bar is used to control recorded video stream. Note that operator need to make a link between a recorded alarm (Log) and the associated video. This is one big requirement of such system integration.

User authentication:

Security is of course expected on such monitoring and control system. It can be a built-in security system or it can be connected to a directory (using LDAP protocol).

Other sub-systems:

The role of the hypervisor is to aggregate all data coming from all sub-systems. Thus, from the console user interface, the operator can have access to lighting or energy sub-systems.

6. OEM AND SOFTWARE COMPANY SOLUTION

6.1 MANUFACTURER PROVIDERS

6.1.1 Argos:

The supplies consist of:

- SDK "DVR IP"
 - o one video thumbnail ActiveX
 - o one ActiveX for records replay
 - o specific dll
 - o one sample program
- an OPC server (OPC-DA 2.0)

Optionally, it is possible to have software (Video Analytics) that allows the configuration of the detection of specific events.

The first ActiveX is only a video thumbnail, but it provides various functions to control the camera:

- PTZ
- Start/stop/read records (remote or locally)
- Take/display snapshots
- Set some configuration parameters
- Audio commands (speaker and microphone)

It can be use several times in the same form as shown below:



Fig 5: Argos Video Live ActiveX

The second ActiveX is an extension of the first one. It already includes a set of commands to select a record to replay.

<p>SPY - Surveillance imProved System State of the art and feasibility analysis D6.1.2</p>	<p>Page 13/23</p>
--	------------------------------

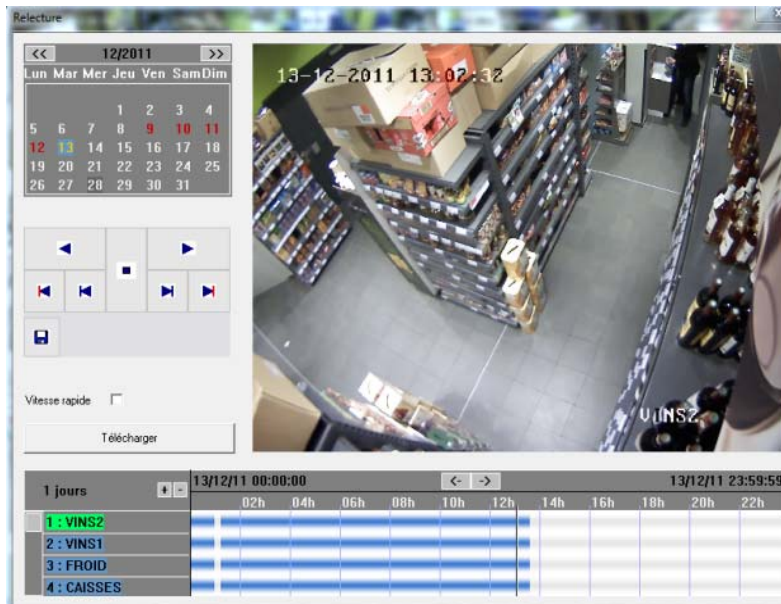


Fig 6: Argos Video Record ActiveX

The basic ActiveX provides a list of events corresponding to mouse and keyboard events (MouseDown, MouseUp, KeyDown, ...).

It is also possible to get alarms and status from the encoders and cameras through the ActiveX, but only when it is shown and connected to the wanted encoder.

In this case the ActiveX has 2 opened connections:

- One for the video stream
- One for the status and alarms

The OPC server provides several items that indicate the status of the encoders:

- Connected or not
- Firmware version
- Number of connected cameras (analogical and IP)
- Number of IO configured

The status of the cameras:

- Connected or not
- Activity detected

And the status of the IOs (switch on or off).

It also provides access to commands linked to the cameras:

- Start/stop recording
- Change IP address
- Change camera Id
- Change the protocol (Sony, Samsung, Axis or AST NET)



It needs a list of the encoders to which it should connect, that can be completed in a .ini file

With Video Analytics another set of items are available through the OPC server for each configured channel:

- Global activity
- Activity in a specific zone
- Alarms in a zone

6.1.2 Bosch:

The supplies consist of:

- ATL COM dlls
- One ActiveX

Their SDK provides the following features.

- Network device detection.
- Concurrent network connections to multiple devices.
- Live video rendering from multiple devices including in-window pan / tilt / zoom (PTZ) control.
- Playback video rendering from multiple devices including direction, speed, and stepping control.
- Live and playback audio rendering.
- Audio streaming to capable devices.
- Recording of live video and rendering of recorded video.
- Still image capture.
- Control of device video and audio.
- Control of relay outputs.
- Event notification from device relays and alarms.
- Device event searching including input alarms and motion alarms.
- Integrated diagnostic logging.

The ActiveX is mainly used to display the video stream. It also allows to directly send PTZ command through mouse inputs, but there is no public functions to send them by a different way.

The SDK is compatible with these following development platforms:

- Visual Studio 2003, 2005, and 2008

SPY - Surveillance imProved System State of the art and feasibility analysis D6.1.2	V01.01	Page 15/23
---	---------------	----------------------



- Microsoft Visual C++ 6.0
- Microsoft Visual Basic 6.0

6.1.3 Geutebrück:

The supplies consist of:

- A set of 3 DLLs (SDK) with several samples designed for the different development platforms
- An OPC Server

One of the samples provided is a video thumbnail ActiveX.

Other programs are provided:

- configuration tool for the video server
- video-wall manager
- joystick simulator to pilot cameras

The OPC Server exports the following resources:

- Cameras
- Monitors
- Digital input contacts
- Digital output contacts

For the cameras the OPC Server provides the following information:

- The global number and name of the camera
- The current state (available / normal / synchronization or contrast failure / sensor alarm)
- The current synchronization signal state (available / failed)
- The current contrast state (OK / alarm)
- The current video sensor state (available / armed / unarmed / alarm)
- The current tour
- The current camera position
- The following commands can be sent directly from the OPC client side:
 - Start or stop a tour
 - Travel to a specific fixed position



Surveillance imProved sYstem

- Activate or deactivate a video sensor, or trigger a test alarm.

For the monitors the OPC Server provides the following information:

- The global number and name of the monitor
- The current state (available / dark / normal switching / alarm switching)
- Which camera is currently switched through
- Which cycle is currently running
- The following commands can be sent directly from the OPC client side:
 - Switch a camera through to the monitor, or clear the monitor
 - Start or stop a cycle

For the digital inputs the OPC Server provides the following information:

- The global number and name of the input contact
- The current state (available / open / closed / sabotage)
- The following commands can be sent directly from the OPC client side:
 - Emulate a switching operation

For the digital outputs the OPC Server provides the following information about:

- The global number and name of the output contact
- The current state (available / open / closed / blinking)

The SDK is compatible with .Net platform and with C++/MFC platform (especially Microsoft Visual Studio 2008, C++, MFC ©).

It can provide the same features as the OPC Server. In addition, other functions are available:

- Pilot cameras (PTZ)
- Connect to a video stream
- Replay archived video
- Audio management

It is also possible to Remote control to a video wall (GsView), for example to display a specific camera with simple text commands.

These are some diagram demonstrating the interaction between a client application (PcVue) and the SDK.

SPY - Surveillance imProved System State of the art and feasibility analysis D6.1.2	V01.01	Page 17/23
---	---------------	----------------------

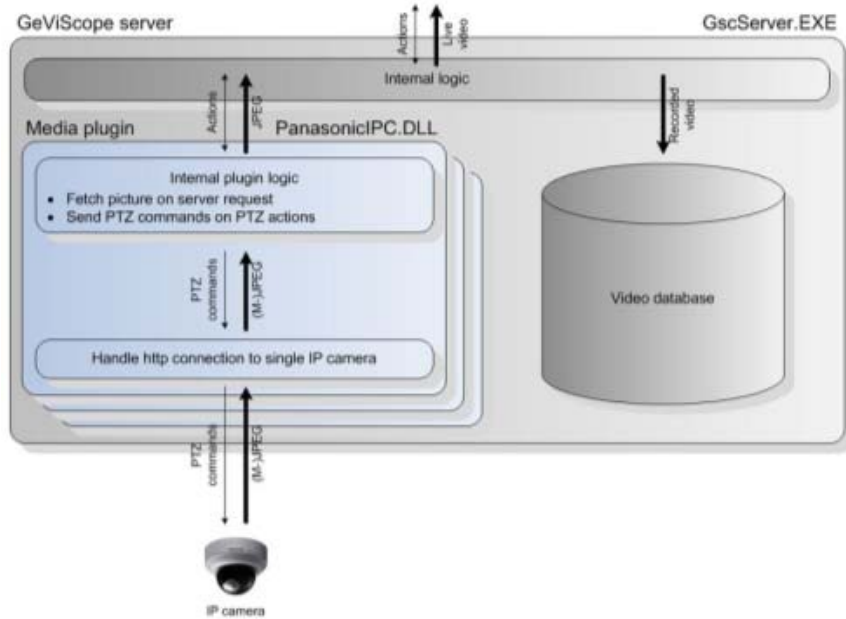


Fig 7: Geutebrück interactions between custom application and their Server

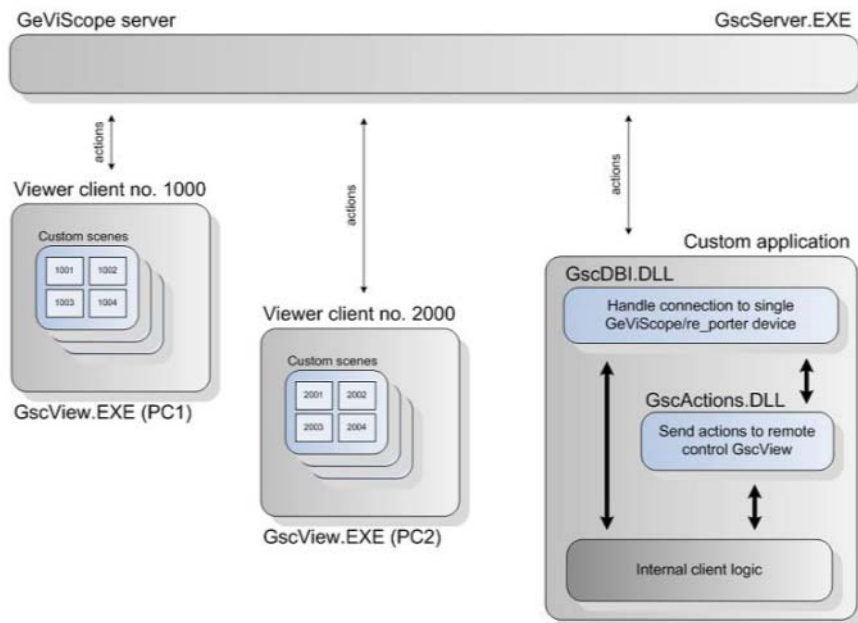


Fig 8: Geutebrück interactions between custom application and their Server

6.2 VIDEO PLATFORM PROVIDERS

6.2.1 Seetec

Their supplies are composed by a set of .Net dlls with some samples, and their configuration tool. It is possible to set some advanced detections rules in their gateway with will raises alarms or record video.

The SeeTec SDK provides controls that can be included in third party applications. You get controls that display the images of a camera, but you do not get the images of the camera. Although it is possible to get single images from the controls, this should only be used as result of user input (for example if the user wants to save the current image), it must not be used to get a constant stream of images.

A camera is either in live or archive mode, the mode cannot be changed once the camera component is created. If you want to display live and archive, you must create two distinct components, one for live images and one for archive images.

For live cameras, the SeeTec SDK provides a PTZ component to allow the user to control the camera. This component can be embedded in an application, and it is possible to create a component that uses api commands to control the camera.

For archive cameras, the SeeTec SDK provides an archive player component to allow the user to control the camera. The user can see where the currently selected camera has recorded images (in different colors for standard and alarm recordings) and he can control various archive player settings like speed or synchronized mode. This component can be embedded in an application, and it is possible to create a component that uses api commands to control the archive settings.

But with a self-developed component, it is not possible to give the user the information about recording ranges. This information is currently not available through api calls, it is only visible with the SeeTec archive component. Everything else that is available in the archive component is also available through api calls.

The last component is a button component. Buttons must be configured by the administrator and allow the user to trigger actions in the SeeTec system, like starting alarms or navigating a camera to a predefined preset position.

The SDK also provides detailed information about the SeeTec installation. All objects in the SeeTec system (like video servers, alarms or ptz actions) are called entities, and it is possible to get information about every entity in the system. These lists can be filtered by type, so it is possible to show a list of all alarms or cameras to the user. Each entity has the same set of properties. Detailed information about entities and their configuration is hidden by the SDK, it is for example not possible to get the frame rate setting of a video source. The details are entirely handled by the underlying objects of the SDK and not visible to the user or developer.

SPY - Surveillance imProved System State of the art and feasibility analysis D6.1.2	V01.01	Page 19/23
---	---------------	----------------------



If the state of the SeeTec system changes, the SDK send a notification to the third party application. There is a broad range of possible notifications, like camera errors, alarms, user logins, and each one generates an event with detailed information about the change.

For most of the events, it is not possible to get events from the past. The exceptions to this rule are alarm events. It is possible to query alarm events by using certain criteria like time or the alarm scenario. So it is for example possible to query for all alarms that occurred in the last 24 hours.

6.2.2 Mirasys(one of the SPY partners):

Mirasys solution supplies consist of a set of dlls and software to configure their Gateway.

It is possible to set some advanced detections rules in their gateway with will raises alarms or record video.

The Gateway SDK is currently available in two different languages: C# and Java.

It provides the following features to all its clients using Promesa protocol version 3.0:

- Connection
- Access control
- Profile handling
- Real-time video streaming
- Playback video streaming
- Activity search
- Ptz control
- Digital I/O
- Recorder events
- Alarms

Promesa is a specific protocol dedicated to the Gateway of Mirasys.



6.3 CONCLUSION

Manufacturer solutions seem to be faster to integrate into custom application. They provide tools and ActiveX that enable to develop specific interface in a short time. But the customer can't use cameras from other manufacturers.

Video Plateform solutions are more flexible because they allow the user to have several kinds of cameras from different manufacturer. The solutions mentioned here also include natively some analytics functions, which is not always the case with manufacturer solutions.

All of these solutions provide quite the same basic features in term of displaying streams, piloting cameras and retrieving status of the cameras and encoders.



7. APPLICATION IN SPY PROJECT

In SPY project, aim is to study a standard client which could interface itself via specific driver to video servers.

The main data to be exchanged between the SCADA and the video server should be:

- Live video streaming
- Recorded video streaming
- Camera status
- Camera alarms (video analysis)
- I/O s
- Camera control

This client shall be a component which can easily be integrated in the SCADA user interface.