Scalable Computing: Practice and Experience
Volume 12, Number 1, pp. 93–103. http://www.scpe.org
ISSN 1895-1767

VIDEO AND SENSOR DATA INTEGRATION IN A SERVICE-ORIENTED SURVEILLANCE SYSTEM

DAMIAN MIERZWINSKI, DARIUSZ WALCZAK, MARCIN WOLSKI, MARCIN WRZOS
NETWORK SERVICES DEPARTMENT
ICHB PAN - POZNAN SUPERCOMPUTING AND NETWORKING CENTER
POZNAN, POLAND
{DAMIANM, DWALCZAK, MARCIN.WOLSKI, MARCIN.WRZOS}@MAN.POZNAN.PL

Abstract. Video surveillance systems have become very popular these days. In households, enterprises and even cities the monitoring systems have been deployed to improve the sense of safety. Additional sensors which produce large amounts of data are very often used independently. A real challenge is a meaningful integration of video and sensor streams, which could assist people operating these systems. This challenge requires a new, smart approach to browsing, analyzing and integrating information of various modalities and from various sources. Nowadays motion detection, face recognition and integration with external sensors are the popular subject of research. We notice the lack of efficient and flexible solutions for managing video surveillance data for public or private security. The problem is scalability and manageability of more and more complicated systems connecting large amounts of devices. In this paper we would like to focus on managing video and other data sources in the service-oriented manner for a large scale domain. Simplifying monitoring configuration and usage is also in the scope of our interest.

Key words: video surveillance system; sensor data integration; video sources management; mobile surveillance; SOA; service integration; event processing

AMS subject classifications. 68M10,68M11,68M14,68U35

1. Introduction. Monitoring solutions, like those deployed in cities or in large companies, grow together with the area under surveillance. Public and private security forces very often have access to many surveillance systems, provided by various vendors. There exist many additional sensors and devices which could add new and useful features to the system if they could only be integrated with it. These require additional management tools and applications. An integrated approach would be desired.

Direct integration is possible for a few different systems but is nearly impossible on a large scale like in the case of integrating private surveillance systems. IT industry is aware of this problem and tries to solve it with Service Oriented Architectures (SOA) where each module exposes a standard interface for communication with other systems. Therefore, SOA principles should be investigated in a distributed video surveillance environment. Another emerging trend in software integration that is suitable to be used in surveillance and public security domains is Event Driven Architecture (EDA) and Complex Event Processing (CEP) [1].

There is a number of requirements for monitoring systems in the literature. They focus on video image quality, simple event detection, annotating and modelling, detecting complex events using simpler ones or finding patterns of events. Other requirements include scalability issues or lack of ontologies for visual surveillance systems. The quality and performance measurement of new surveillance systems should be addressed here as well [2].

Visual monitoring alone is very often not enough to satisfy the growing expectations put on the surveillance systems. Research interests are put on integrating video, audio and data originating from other sensors, e.g. motion sensors, door locks, RFID scanners. Methods of combining visual data from various video sources or tracking objects between them are also subject of research [3].

Scalability, effective management and QoS in large and distributed monitoring systems are the popular topics of research efforts [4, 5, 6]. Surveillance of highways, cities, forests and industrial terrains managed by the police, public forces or security companies are the best examples of systems that find the above properties important.

The paper presents how the SOA and CEP principles can be applied to large and distributed monitoring systems which have to be scalable and easily configurable. The contribution summarizes research and development achievements on mobile surveillance platform at Poznan Supercomputing and Networking Center[1] (PSNC). The work on the service oriented surveillance system has been started in collaboration with the Polish police and has gained a positive feedback from the target users. This work was earlier described in [7]. As a promising extension to this topic video and sensor streams integration is being conducted within the Future

[1]http://www.man.poznan.pl

Internet Engineering project[2]. The whole design has been extended with event processing mechanisms that simplify management of various types of devices.

This paper is organized as follows. Section 2 describes state of the art in large video surveillance systems. Section 3 describes the system architecture, video and event processing, services registration, video browsing, video description. Implementation details can be found in section 4, and a brief description of first tests in section 5. Section 6 concludes the paper.

2. Related work. First video surveillance solutions were based on an analogue technology called Closed Circuit Television (CCTV). In the most trivial case these were cameras connected with TV sets located in one room. Currently most of the video cameras use a digital Charge Coupled Device (CCD) to capture images and analogue techniques to distribute and store data. The conversion between digital and analogue signal causes video quality degradation. Thanks to technology evolution fully digital systems powered by computers can be used without conversion. With high performance computers additional features can be expected, e.g. real time events detection and license plate recognition. In [8] three main generations of surveillance systems were distinguished: analogue CCTV systems, automated visual surveillance done by combining computer vision technology with CCTV systems, and automated wide-area surveillance systems. This contribution focuses on 2nd and 3rd generation systems especially in distributed architectures. Image processing is out of the scope of this paper.

Highway monitoring solutions are probably the first and the most distributed surveillance systems. At the beginning simple inductive loops were used to detect cars. Nowadays similar sensors count traffic, measure queue lengths on ramps and at intersections or help at parking lots to find a free place. In [9] authors describe VDS240, a vehicle detection system based on wireless sensors. The sensor infrastructure consists of two different nodes, the magnetic one, placed in the center of the lane and the Access Point (AP) located on the side of the road. The AP aggregates data from magnetic sensors and sends it to the Traffic Management Center (TMC) or a local controller. The radio interface enables direct communication with up to 96 nodes within the range of 1500m. Similar systems are built using laser [10], radar [11], acoustic [12] and optical sensors [13]. Because of the hardware development and great improvements in the video analysis, sensors are replaced or supported with video cameras. It makes highway monitoring systems one of the most significant examples of distributed video surveillance solutions. Captured images are used to estimate all parameters usually measured by sensors. Depending on the quality of the video stream and the algorithms used the system can classify vehicles, trace lane changes or even recognise license plates. Usually the system requires only one camera per monitored location. The video scene must be segmented and transformed into objects. Their behaviour is recognised and tracked. An example of such a system was described in [14]. High resolution cameras can be applied to monitor objects smaller than cars (e.g. in Stockholm, Sweden cyclists are under surveillance). In [11] authors present a video analysis system applied to detect biking in the "wrong direction". They also analyse traffic conflicts between cyclists and other road users. They use the foreground-background segmentation for the trajectories estimation and the shape analysis of points of interest for the speed estimation.

Some new surveillance solutions are characterized by the fact that they use visual surveillance methods combined with information originating in additional systems, such as sensor devices or external knowledge bases. They are gathered under the terms smart surveillance or multisensor surveillance systems. The key idea is to enhance monitoring with sensors or integrate it with whole sensor networks where video cameras are just one type of sensors.

In [15] authors present a data fusion system for objects tracking. They use optical and infrared sensors to monitor the same outdoor scene. Redundant information is fused together to obtain a more accurate estimate. Physical sensors are connected directly to first level nodes according to their location. Nodes create a hierarchical structure. Data fusion is performed in a centralized fashion considering sensor reliability every time. Authors compared this to a single camera system and noticed more accurate trajectories.

Surveillance systems are getting bigger and more complicated. Software architects deal with this issue using service-oriented architecture. According to the SOA paradigm, software should be delivered as loosely coupled, and cooperating services which should be described, published and easily discovered. In such an environment new applications called business processes can be created by composition of existing services [16]. SOA is mature in the business world [17, 18, 19] and is expanding to new domains like video conferencing [20] or the public security sector [21].

[2] http://www.iip.net.pl

N.E.S.T [22] is an example of an SOA-based surveillance system. It was built by the Institute for Information and Data Processing, Fraunhofer IITB. Authors worked on decentralization, expandability and upgradability of today's monitoring solutions. They developed an architecture and set of services for video surveillance, e.g. motion detection and tracking, and abandoned luggage detection. In N.E.S.T an operator defines different surveillance tasks as automated processes, sometimes with human interaction. The tasks are modelled in BPEL language and executed in the BPEL-engine connected to services through a service-bus. There is a second JMS-based system bus for very frequent notifications. A third bus is planned for the future, it will be an infrastructure for streaming. Static and dynamic data are stored in the World Model through the Model Access Service. Presented services are suitable for a hotel scenario. In this case, the receptionist starts the new guest tracking process. It estimates his/her route to the room and checks if the guest did not get lost.

Another example of a smart surveillance system is developed by the SAMURAI project, funded by the EU FP7. One of the project objectives is object detection in multi-camera environment under real world conditions and on using multi-modal data fusion. The detection system is based on heterogeneous sensor network consisting of fixed position legacy CCTV cameras and mobile cameras with GPS receivers.

IBM in [23] highlights three key challenges that need to be addressed to enable the widespread deployment of smart surveillance systems:

1. The multi-scale challenge better information acquisition based on video analysis, e.g. face recognition, person tracking
2. The large system deployment challenge.
3. The contextual event detection challenge better interpretation of gathered information to detect events and identify trends

The Exploratory Computer Vision Group in IBM addressed these challenges in their solution [23]. The IBM Smart Surveillance System has two key components: the IBM Smart Surveillance Engine (IBM:SSE) and the IBM Middleware for Large Scale Surveillance (IBM:MILS). The first one is a software-only event detection technology based on video analysis likes: object detection, object tracking, object classification. It also provides real time alerts and creates Viewable Video Index description of all interesting activities in the video, implemented as a set of XML files. MILS converts this index to relational tables and provides powerful Query Services mechanism. Thanks to it user will be able to query for activities providing time period, object size, object class, object motion description. It will be also possible to query by context-based content similarity, e.g. [23] "Show all activities where there were blue cars or cars similar to this car here the user specifies an example car through an image".

This paper addresses configurability, flexibility and sensor integration issues. Based on previous experience and state of the art in video surveillance, an SOA-based architecture for monitoring video delivery and CEP-based sensor data processing has been proposed.

3. System description.

3.1. Architecture. The proposed solution is based on SOA principles in order to address flexibility and integration issues of present video surveillance systems. SOA is an architectural style that supports service orchestration [24]. In particular, SOA defines the find-bind-execute paradigm to differentiate between service providers and service consumers and their loose coupling. We propose to apply this main paradigm to the video surveillance system. Instead of creating a direct relationship between the provider (video source) and the consumer (typically a person who is watching the video content) each video source is seen in the system as a single video service. All registered video services create the service cloud. Then someone who acts a service consumer explores the cloud in order to find relevant service, binds to its endpoint and executes the service.

Referring to SOA principles a prototype solution for a mobile monitoring station was prepared (see Section 4.1). The main part of the architecture is the Tiberinus server. The server consists of three main components (see Figure 3.1). The first one is an application enabling to register external video sources in the system. The only requirement for a device or application to register and become a video source is to expose RTP/RTSP video streams. The second component is a streaming server reflecting video streams originating from the video sources and a video recorder, an additional application that records video streams and stores them as files in the file system. The third component is an application that allows users to search the repository of registered video sources and archived video material. It also has the capability to playback and download archived and live videos. It provides methods of searching for the desired video sources and archived material, registering video sources, accessing their descriptions and services provided by them. Both the register and search application expose web service interfaces named Tiberinus-Manager and Tiberinus-Search, respectively.

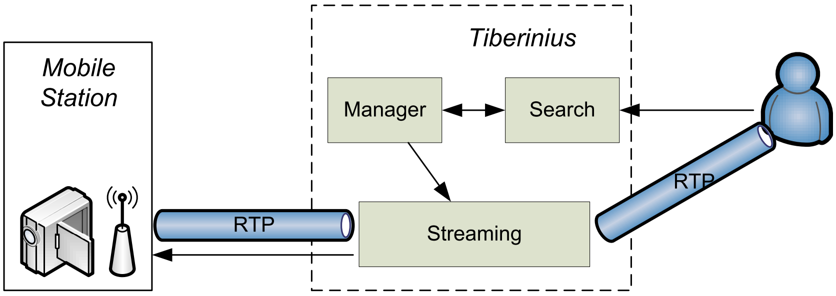

Fig. 3.1. *System architecture*

Server modules are developed as loosely coupled services so configuration is as easy as SOA infrastructure management. One can use standard tools like BPEL editors and runtime engines. In case of adding new features like face or license plate recognition service one has to define a new BPEL process and deploy it on the BPEL engine.

While SOA is proper for service integration there exist more convenient architectures for sensor data integration. Sensor devices are usually capable of sending simple events including appropriate data. In order to achieve high efficiency and low energy consumption sensor gathers data, sends it and goes to sleep mode its network interfaces are turned off. Low level events sent by sensors can be easily mapped to the events in the IT world. This task is performed by the event streaming layer which is dedicated to sensor data integration.

Figure 3.2 presents an extended view of the architecture with Tiberinus in a service ecosystem together with video analysis services and event processing layer.

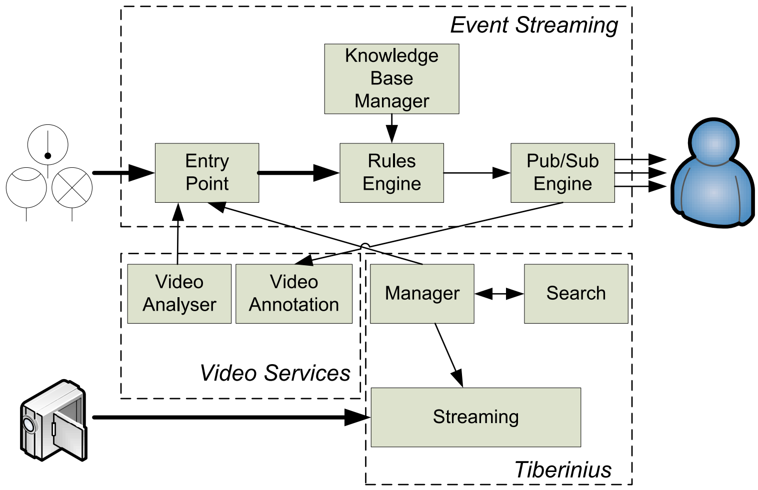

Fig. 3.2. *Tiberinus ecosystem*

The event processing layer allows users of a surveillance system to subscribe for the events they are interested in. The system will send notifications to the user in case of an alert is raised or a specified event on sensor device happens. It can be done on different levels of abstraction. The example of the low level event in the surveillance domain is "the light is turned on", on the other hand the high level event could be "employee X came to work". The first module in this layer collects events and exposes entry points for sensors and services. Events are grouped by time windows and passed to the event processing module. This module is based on Business Rules (BR) engine which applies user defined actions to a particular event configuration. BR engine can check if the event occurred or not, what are the event details and if there was any other associated event raised before. All actions are defined by rules. They can be managed by a special module with a user friendly interface. User can define his own events, rules and actions through this UI. All data is stored in the shared repository. Rules in BR engine generate new events which are distributed by the notification module. New event enters the event processing layer by the entry point or can notify services that were subscribed for this type of event.

3.2. Video source registration. Registration of video sources on the server is done using the Tiberinus-Manager web-service located on the Tiberinus server. The web-service can be accessed using either the SOAP or REST interface.

Registration of video sources on the server consists of two parts. The first part is submitting all the metadata to the Tiberinus server. This metadata includes descriptors such as the name of the video source, its text description, geographic location, registration date, etc. Additionally, information about all of the services that are provided by the video source is included, e.g. PTZ control, image parameter selection. The second part is sending the RTP/RTSP video stream to the server where it can be archived or reflected to client applications. When the video source is successfully registered on the server its description is made available through the Tiberinus-Manager web-service. Information about the URI location of the live video stream and locations of the archived material (either as a URL to a static file or to a video stream) is also made available in the same way.

Very few video devices have the capability to push and announce video streams that they produce. In order to broaden the spectrum of compatible devices, the process of pushing the stream to the server was replaced with pulling the stream from the video devices. After the first part of registration finishes successfully, Tiberinus connects to the address provided with the registration metadata and negotiates the connection using RTSP.

Video sources can be registered in many Tiberinus servers at the same time. This could be done in order to send the video sequences to different locations/institutions or to store the video material in more than one place. However, regarding that the mobile video source would often have limited bandwidth capacity (e.g. when using a modem or GSM/3G wireless connection) this would not be the optimal choice. Instead, video sources once registered in one Tiberinus server could be registered in other Tiberinus servers thus building a hierarchical structure. Registering video streams in Tiberinus servers that are higher in the hierarchy is done analogously. All of the metadata submitted during the registration will be the same. Optionally, it can be enriched with information about node hierarchy and a description of each node. The only element that will be different is the RTSP address of the video source, now pointing to the Tiberinus video server (see Section 3.5).

3.3. Searching and playing streams. In traditional CCTV recorders one has to connect the screen to the device and find relevant recording using remote control. In our opinion it is not enough, nowadays people are used to Video on Demand served via a web browser. Most TV shows, series and clips are available on web pages. Modern surveillance systems should adapt this common way of watching videos.

In presented system an end user can browse and search for the desired data using the web application. The search can be performed using multiple criteria. The user can input the name or a description of the video source, the date of the beginning or end of the recording. The user can select if s/he is only interested in finding live streams, registered videos or both. Additionally s/he can enter the type of the video source: either a static legacy CCTV camera or a mobile video source. The repository can also be searched using location criteria. The user can select a location on the map and enter search distance or enter the geographical coordinates manually. If the location criteria are used, only the results originating from video sources located in the searched locations will be presented.

Results returned by the web application contain all the metadata related to the video source, URLs to the video streams and, in case of archived videos, URLs to the files that can be downloaded. The user can view all of the videos using the QuickTime movie player embedded in the webpage.

Tiberinus provides also machine-oriented method of searching the videos which can be accessed using the REST or SOAP interface of the Tiberinus-Search web-services. Web-service exposes similar methods of searching for live video sources and archived material to those found in the user interface. All the search criteria mentioned earlier can be used.

3.4. Event processing. After registering a new video source the manager module generates an event. This event goes through the entry point to the rule engine. BR engine results go to the Pub/Sub module which notifies proper services (see Figure 3.3). Next, depending on the service functionality configuration the right action is taken. Similar scenario takes place when a new sensor is announced in the system and starts sending data. The multimedia processing services connect to the streaming server and immediately start to stream the audio-visual content in order to perform the analysis. After the analysis of the part of data (e.g. a single video frame) is completed a new system event containing its result is produced (see Figure 3.3). Depending on the nature of the service the event can be generated every time or can only be generated when the analysis result is positive. The example video processing services could consist of a face detection, automatic number plates recognition or movement detection services. In more complex systems those services could perform the unattended baggage detection, person identification or could perform image analysis on an x-ray, infrared or MMW images. The second set of services is configured to listen to the other system events, e.g. containing the data analysis results. It is possible to launch the services only if certain conditions occur, e.g. when two events of given types occur in a given time window or on given time and system conditions. At this point there is a possibility for a user to implement advanced processing rules combining events concerning different types of data or to use external data sources and context information. The video annotation tool is an example of such service. After a face detection event is published the video annotation tool updates the description of the video stream where the face was detected with time and spatial information about the event. When the human voice event was published this information is also included in the description. When another service identifies the person information in the video stream description can be further specified and extended. The annotation tool could be configured to listen to any kind of events and to extend the video file description in appropriate manner. The video annotator service is subscribed to receive the recognition event but not the sensor event. Additionally in the event processing engine there is a rule that will generate a complex event when a recognition and a sensor event appear in the system in a given time window. The video annotator is subscribed for the complex event and when it appears in the system the service is notified.

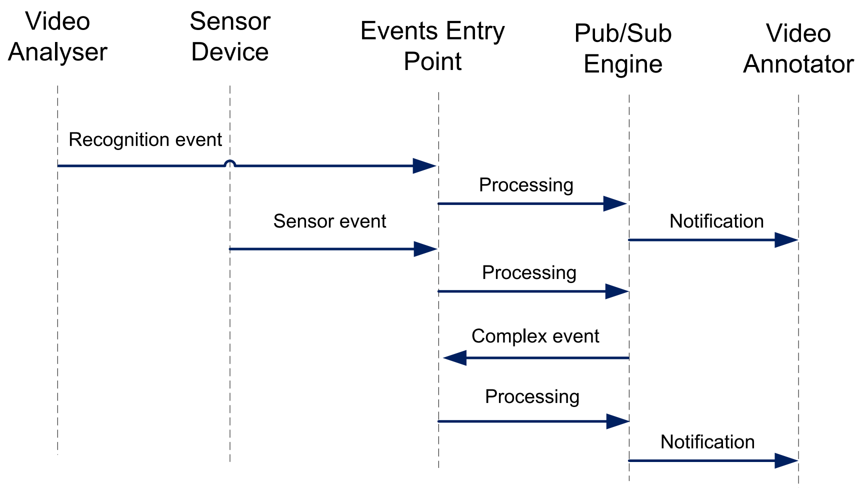

Fig. 3.3. *Video annotation processing path*

More complex service is the tailgating detection service. It is configured to listen for the "door opened" and "person detected" events. When the system detects one event of the first type and more than one event of the second type on the configured window in the video image from the camera pointed at the door then it produces an alert. The alert is an event of the new kind: "tailgate detected". Other services can listen for that event and react, e.g. confirm it or notify selected people. All of the events in the system can have specified priority level and confidence factor. Other more complex scenarios could include the use of information from the sensors, other image processing services and context services.

3.5. Distributed architecture. The previous sections present the basic configuration of the video surveillance system and sensor fusion ecosystem.

However, to address requirements and challenges posed by large scale domains, where multiple video sources may exist, distributed configuration needs to be used. A hierarchical architecture similar to [25] has been created but in presented case structure is not forced by geographical location. It is configured based on business rules and user requirements. Individual user or company might wish to investigate different location from one place and pass streams to proper institutions or a cloud service. Two types of nodes can be distinguished: leaves and internal nodes. A leaf can work as a fully featured local surveillance system. Internal nodes aggregate, archive and pass streams according to user requirements (Figure 3.4). Their connections depend on use cases. Bigger companies with more than one location would probably like to aggregate streams and archive them in one place.

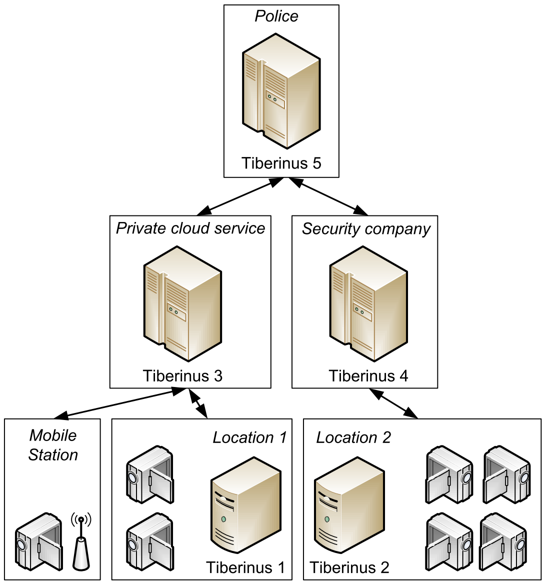

FIG. 3.4. *Example of Tiberinus distributed configuration*

Small businesses usually outsource security services and give them access to streams from the local surveillance system.

4. Prototype.

4.1. Ad-hoc monitoring station. As proof of concept, a Mobile Monitoring Station (MMS) prototype has been prepared. In general it can be described as an autonomous and manageable streaming source. It can be used to monitor areas not being covered by standard monitoring solutions.

The prototype MMS is a car with installed equipment which streams video over wireless networks: namely 802.11g or UMTS (see Figure 4.1). The main part of MMS is an access point, OSBRIDGE 3GN[3]. It connects to the remote server and establishes an encrypted vTun[4] channel. Other devices can communicate and upload video streams through this channel. Axis M7001 video server, which allows connection of an arbitrary analogue video camera has been used as the video source. Every device is supplied with power from the primary battery used in the car, or an additional extended battery if needed.

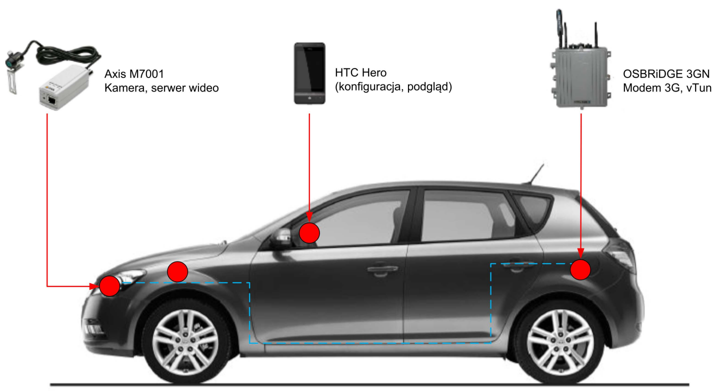

FIG. 4.1. *General MMS prototype overview*

The MMS is managed over the WIFI network by an HTC Hero mobile phone based on Google Android 1.6, with a dedicated application installed. It is simplified to two main actions: registering and unregistering video

[3]http://www.osbridge.com/?q=en/node/93
[4]http://vtun.sourceforge.net/

sources at the remote server. Moreover, it is possible to view video streams that are generated by currently registered cameras. This gives a possibility to verify the camera configuration (zoom, angle, focus, etc.). When registering video sources using the mobile phone application, information about the phone location (taken from the phone's GPS receiver) and direction (taken from the phone's digital compass) are sent to the server. Therefore, it is advised to align the phone with the camera's line of view to store information about the camera's direction.

4.2. Tiberinus server. The Tiberinus server consists of three components that were briefly described in Section 3.1. The software managing video streams and storing the metadata information was developed by PSNC. Darwin Streaming Server[5] was used as the streaming server enabling live and stored video material playback. VLC[6] media player was used to fetch the stream from video source, send it to video server and optionally store it in the file system. All of the web services and the web application allowing search and playback of video streams were also developed for this project.

All of the components of the Tiberinus server were installed and configured to work on a virtual machine running on the Linux operating system. This solution enabled testing the system in different network topologies and configurations.

4.3. Sensor data integration. As a proof of concept laboratory configuration of the event processing layer was build. Implementation is based on JBoss Drools[7] - Business Logic integration Platform. This package consists of three modules: Event processing, Rules and Workflow. First two were crucial while the third one was useful for integration with services layer. Knowledge base was build through Drools Guvnor application which is a web based rule manager for Drools. Notification module "Pub/Sub Engine" is based on XMPP protocol and implemented upon Jabber server.

Sensor network is based on SunSPOT[8] devices and Axis IP Cameras. SunSPOTs provide data about the environment like temperature, light level and acceleration of the devices. The Axis IP Camera's provide basic movement detection functionality. Video services were developed on basic functionality of OpenCV library. The annotations on the surveillance material are performed using the MPEG-7 standard.

5. Proof of concept. The proof of concept performed in the laboratory and in car environment aimed to confirm if the system architecture, as well as software solutions are capable of handling a basic set of video streams in a service oriented manner.

5.1. Laboratory configuration. In the first scenario one set of devices was located in an office building in Poznan. It consisted of an analogue camera and a video encoder. The camera was placed outside and attached to a balcony facing cars stopping at traffic lights. The parameters of the camera were set to focus on the cars' licence plates. The camera was connected to a video server which was connected to the Tiberinus server using an Ethernet connection.

The second set of devices was located in the same office building. It consisted of a Wi-Fi capable IP camera and a wireless 3G router. The camera was placed inside the building and was pointing at the same road junction, but was configured to capture the whole scene. The camera was connected to the wireless router using a Wi-Fi connection. The router was connected with the same Tiberinus server as before using the 3G connection, and a VPN tunnel was established between them.

After the infrastructure was set, a testing procedure was performed. The video sources were registered using the mobile application. After that the system was searched for desired video materials: currently registered live video sources and video sources located near the coordinates of the office building. In the second scenario an additional Tiberinus server was deployed. The second set of devices was connected to this additional server. Both of the Tiberinus servers were connected to the ESB server. Similar testing procedure was performed. The video sources were searched using two different Tiberinus web applications.

Finally, the system was configured as in the first scenario but with Ethernet connection instead of 3G. The server was set to record the video material. In this configuration it was left unmodified for a week. About 13GB of video material was obtained together from both cameras, compressed with the h264 video codec. This video material could be used for further analysis.

[5] http://dss.macosforge.org/
[6] http://www.videolan.org/vlc/
[7] http://www.jboss.org/drools/
[8] http://www.sunspotworld.com/

5.2. Mobile configuration. Accordingly, a wide scope of tests has been performed using the MMS as video source and uploading data through UMTS. The car with a surveillance system installed streamed video from many different locations which resulted in variable strength of a 3G signal. Static tests were conducted at a parking lot. Moving objects like cars or people walking on the pavement were recorded. Mobile tests were performed from the car being in motion, riding through the streets of the center of Poznan.

As expected, the quality of gathered records is noticeably lower in comparison to video streamed over wired networks. Subjective experience can be described as low resolution but smooth video, which was achieved for TCP as the transport protocol for RTP stream. Surprisingly, the UDP protocol had to be abandoned because of a very high ratio of lost frames when testing in poor network conditions, i.e. high ratio of lost packets and high latency.

5.3. Event processing. Using architecture described in 3.4 and implementation described in 4.3 several tests were conducted. Axis Video camera and its motion detection functionality was used. SunSPOT sensor network was installed. Simple video analysis service which detects faces on the image was developed. With this tools complex events related to conference room were detected, e.g. possible conference room occupancy or room condition changes. Integration with services layer was also tested thanks to video and notification services.

Complex events add several additional dimensions to the monitoring process. It was proven that in the test scenario the decisions based on extra assumptions are more accurate. A turned on light does not imply meeting inside a conference room someone might forgot to turn it off. Motion detection also does not imply meeting maybe someone is doing cleaning. Face detection might be helpful and face detection event might indicate meeting if faces are looking at the screen together. Usually one sensor is not enough to make the right decision and take specified action. When using the combination of those events the false positive rate decreases.

In order to detect some real life context events rules were written in special Drools domain specific language. One of that rules is shown below.

```
rule "possible-meeting"
when
   LightEvent(status==LightEvent.ON)
   $m : MotionEvent()
   FaceDetectedEvent(counter > 1, this after $m)
then
   PossibleMeetingEvent e = new PossibleMeetingEvent(new DateTime())
   insert(e)
end
```

FIG. 5.1. *Drools rule example*

Each rule contains two main sections, the conditions section after word *when* and actions section after word *then*. Declarations start with *rule* keyword followed by name of the rule.

Figure 5.1 shows *possible-meeting* rule which detects a potential meeting in a conference room. Rule conditions say when the rule actions have to be taken. In this example all three events have to occur: the light must be turned on, motion has to be detected and then the face detection service has to find more than one face. These real life events are mapped to events in the system, accordingly: LightEvent, MotionEvent and FaceDetectedEvent. When all conditions are fulfilled the new event is created. New event notifies proper services that the conference room might be occupied. During the creation the current time is saved within the event. Last line of the actions section inserts this event to the rule engine. This way the new event can be used by other rules.

Rules give a user great flexibility in combining low level events into higher level events which should trigger proper services or notify user. Drools Guvnor provides a user friendly interface for building rules so the user do not have to know all the language details.

6. Summary. This paper described the way the SOA principles and tools can enhance modern video surveillance systems. Easy configuration and integration of distributed and differentiated video solutions can enhance public and private security forces work. Flexible composition of smart surveillance services like face or licence plate recognition is a great advantage of loosely coupled CCTV implementation. In addition, an

IP-based and loosely coupled architecture is ready to scale, which has been proven in many enterprise solutions. Easy access to recorded and live streams will improve the efficiency and usability of CCTV. The user friendly interface should encourage home and small business users to invest in modern surveillance solutions.

The event processing subsystem is flexible and can be extended to fit the requirements of the particular surveillance site. The system events and event processing rules can be extended during the development process. The simple video analyzing services as described are to demonstrate the basic capabilities and show the future opportunities of such services in the domain of visual surveillance using the architecture presented in this paper.

The crucial part of further development works will be mobile station tests in operational conditions. The first evaluation has discovered that mobile video sources very often have weak processing abilities. Additionally network connections in a mobile environment often have limited bandwidth and therefore the image quality is strongly dependant on the network provider and technology used. In such conditions management and description of video sources is crucial.

The first attempt described in the related work paragraph and presented proof of concept solution confirmed that SOA is a promising direction of research for large surveillance systems. Scalability tests, sensor's data fusion, semantic description of systems and content are the most interesting topics for further work based on the presented prototype.

REFERENCES

[1] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, 2002

[2] W. Liu, P. Miller, J. Ma, W. Yan, *Challenges of distributed intelligent surveillance system with heterogenous information*, 2009

[3] W. Hu, T. Tan, L. Wang, S. Maybank, *A survey on visual surveillance of object motion and behaviors*, IEEE Transactions on Systems, Man and Cybernetics, Volume 34, 2004

[4] S.S. Yau, N. Ye, H. Sarjoughian, D. Huang, *Developing Service based Software Systems with QoS Monitoring and Adaptation*, Proceeding of the 12th IEEE Int'l Workshop on Future Trends of Distributed Computing Systems, Honolulu, Hawaii, USA, October 2008

[5] S. Sutor, F. Matusek, F. Kruse, K. Kraus and R. Reda, *Large-Scale Video Surveillance Systems: New Performance Parameters and Metrics*, The Third International Conference on Internet Monitoring and Protection, Bucharest, July 2008

[6] F. Licandro, G. Schembra, *Wireless Mesh Networks to Support Video Surveillance: Architecture, Protocol, and Implementation Issues*, EURASIP Journal on Wireless Communications and Networking Hindawi Publishing Corporation, Volume 2007, Article ID 31976

[7] D. Mierzwinski, D. Walczak, M. Wolski, M. Wrzos, *Surveillance System in Service-Oriented Manner*, synasc, pp.427-433, 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2010

[8] M. Valera, S.A. Velastin, *Intelligent distributed surveillance systems: a review*, IEEE Proc.-Vis. Image Signal Process, April 2005

[9] A. Haoui. et al., *Wireless magnetic sensors for traffic surveillance*, Transport. Res. Part C (2007), doi:10.1016/j.trc.2007.10.004

[10] A. Rakusz, T. Lovas, A. Barsi, *LIDAR-Based Vehicle Segmentation*, International Conference on Photgrammetry and Remote Sensing, Istanbul, Turkey, 2004.

[11] J. Fang, H. Meng, H. Zhang, X. Wang, *A low-cost vehicle detection and classification system based on unmodulated continuos wave radar*, Proceedings of the 2007 ITSC, Seattle, WA, 2007

[12] R. López-Valcarce, C. Mosquera, F. Pérez-González, *Estimation of Road Vehicle Speed Using Two Omnidirectional Microphones: a Maximum Likelihood Approach*, EURASIP Journal of Applied Signal Processing, vol. 8 pp.1059-1077, 2004.

[13] G. Gritsch, M. Litzenberger, N. Donath, B. Kohn, *Real-Time Vehicle Classification using a Smart Embedded Device with a 'Silicon Retina' Optical Sensor*, Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008

[14] L. Vibha, M. Venkatesha, G.R. Prasanth, N. Suhas, P. Deepa Shenoy, K.R. Venugopal, L.M. Patnaik, *Moving Vehicle Identification using Background Registration Technique for Traffic Surveillance*, Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong

[15] L. Snidaro, G.L. Foresti, G. Luca, R. Niu, P.K. Varshney, *Sensor Fusion for Video Surveillance*, 7th Int. Conf. on Information Fusion, 2004

[16] M.P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, *Service-Oriented Computing: State of the Art and Research Challenges*, 64-71, November 2007

[17] D. Krafzig, K. Banke, D. Slama, *Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series)*, Prentice Hall PTR Upper Saddle River, NJ, USA, 2004

[18] C. Legner, R. Heutschi, *SOA Adoption in Practice-Findings from Early SOA Implementations*, European Conference on Information Systems, p. 16431654, 2007

[19] C. Baroudi, F. Halper, *Executive Survey: SOA Implementation Satisfaction*, Hurwitz and Associates, 2006

[20] J. Alcober, G. Cabrera, X. Calvo, E. Eliasson, K. Groth, P. Pawalowski, *High Definition Videoconferencing: The Future of Collaboration in Healthcare and Education*, eChallenges e-2009 Conference, October 21-23 2009, Istanbul,

Turkey
[21] C. Mazurek, M. Stroiński, D. Walczak, M. Wolski, *Supporting high-tech crime investigation through dynamic service integration*, Fifth International Conference on Networking and Services, 2009
[22] A. Bauer, S. Eckel, T. Emter, A. Laubenheimer, E. Monari, J. Mosgraber, F. Reinert, *N.E.S.T. - Network Enabled Surveillance and Tracking*, Future security: 3rd Security Research Conference Karlsruhe; 10th-11th September 2008
[23] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C. Shu, and Y. Tian, *Smart Video Surveillance*, IEEE Signal Processing Magazine, march 2005
[24] The Open Group, *SOA Source Book*, Van Haren Publishing, 2009
[25] M. Pellegrini, P. Tonani, *Highway traffic monitoring: Main problems and current solutions*, Advanced video-based surveillance systems, 27-33, 1999

Edited by: Dana Petcu and Alex Galis
Received: March 1, 2011
Accepted: March 31, 2011