Augmenting Surveillance System Capabilities by Exploiting Event Correlation and Distributed Attack Detection

Francesco Flammini, Nicola Mazzocca, Alfio Pappalardo, Concetta Pragliola, Valeria Vittorini

Augmenting surveillance system capabilities by exploiting event correlation and distributed attack detection

Francesco Flammini[1], Nicola Mazzocca[2], Alfio Pappalardo[1 2],
Concetta Pragliola[1], Valeria Vittorini[2]

[1] Ansaldo STS, Innovation & Competitiveness Unit, Via Argine 425, Naples, Italy
[1] University of Naples "Federico II", Department of Computer & Systems Engineering, Via Claudio 21, Naples, Italy
{francesco.flammini, concetta.pragliola}@ansaldo-sts.com
{nicola.mazzocca, alfio.pappalardo, valeria.vittorini}@unina.it

Abstract. In recent years, several innovative security technologies have been developed. However, many of the novel sensing technologies (e.g. video analytics) do not always feature a high level of reliability. Very often, they need to be precisely tuned to fit specific installations and provide acceptable results. Furthermore, in large installations the number of surveillance operators is low with respect to the number of sensing devices, and operators' tasks include facing critical events, possibly including strategic terrorist attacks. In such human-in-the-loop systems, ergonomics and usability issues need to be carefully addressed to increase system performance in terms of detection probability and low rate of false/nuisance alarms. This paper describes a multi-sensor event correlation approach for augmenting the capabilities of distributed surveillance systems. The aim is to provide advanced early warning, situation awareness and decision support features. The effectiveness of the framework is proved considering threat scenarios of public transportation systems.

Keywords: Physical Security, Surveillance Systems, Situation Awareness, Event Correlation.

1 Introduction

In modern society, the assurance of a secure environment has become an issue of paramount importance. Infrastructure protection against potential threats is usually performed by surveillance systems that are more and more large, distributed and heterogeneous [1]. The cyber-physical and human-in-the-loop nature of this field requires a set of multidisciplinary activities to be performed in order to adopt appropriate and effective protection mechanisms. Due to the variety of natural and malicious threat scenarios, a growing set of different sensing technologies are required. However, many of the developed novel innovative technologies (e.g. video analytics) do not always provide adequate reliability (see e.g. [2], [3]). Many

automatic and intelligent detection subsystems generates unnecessary warnings, which can be classified as false alarms or nuisance alarms. Therefore, with regard to the decision support feature of surveillance systems (e.g. for triggering countermeasures), it is very important to control the rate of these alarms [4]. The integration of information coming from different sources, as sensor networks, is the key for new generations of multi-modal surveillance systems, where many different media streams (video, audio, sensor signals) concur to provide a greater situational awareness and a better decision support [5].

So far, the potential capabilities of the traditional systems are limited by their low capabilities in data analysis and interpretation and hence in real-time prevention and reaction. Since a few human operators are usually employed in security surveillance, human-factors related issues also need to be carefully addressed, including cognitive ergonomics in human-machine interaction [6]. Therefore, the real need is for distributed surveillance systems, acting not only as supporting platforms, but as the concrete core of real-time data comprehension process [7], in such a way to achieve advanced early warning and situation awareness capabilities. These requirements are increasingly important in many application domains, like Homeland Security, environmental sensing, crisis management and other information-rich domains, where a large number of dynamic objects are engaged in complex spatial-temporal relations [8].

This paper describes how to augment the capabilities of an existing integrated surveillance system (briefly presented in Section 2) by means of a complex event correlation framework (briefly presented in Section 3). The details of the integration are provided in Section 4, while Section 5 presents some application examples. Finally, Section 6 draws conclusions and some hints about future developments.

2 A surveillance system for railway protection

The Security Management System (SMS) is a multimedia surveillance system used to improve the security of critical infrastructures, in particular the ones used for rail transportation [9]. It integrates intrusion detection and access control, intelligent video-surveillance and intelligent sound detection devices. The system may also integrate CBRNe (Chemical Biological Nuclear Radiological explosive) sensors to improve detection of terrorist attacks.

The SMS architecture (Fig. 1) is distributed and hierarchical; a dedicated network provides reliable communication among the sites and an integrated management system collects the alarms and supports decision making. In case of emergencies, the procedural actions required to the operators involved are orchestrated by the SMS.

Data gathered from the heterogeneous sensing devices are processed by subsystems which generate the alarm events. Those alarms are first collected by peripheral control centers (Peripheral Security Places, PSP, positioned in the stations) and then centralized in line control centers (Central Security Places, CSP, close to the traffic management center). Every security place (peripheral or central) can be provided with a SMS operator interface.

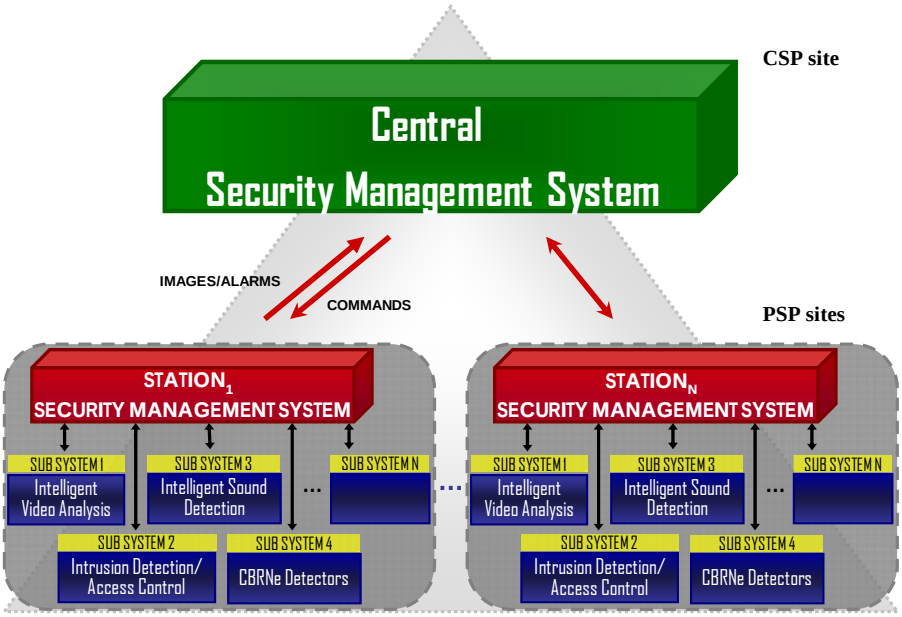

Fig. 1. SMS architecture.

The events detected by the available sensorial subsystems are stored in appropriate repositories, both at the PSPs and CSP sites. A simplified schema of the stored data, also named *Event History* in the following, is shown in Table 1.

Table 1. Schema of Event History

Field Name	Field Description	Field format (example)
IDev	Event Identifier	E*x* (e.g. E8)
IDs	Sensor Identifier	S*x* (e.g. S4)
IDg	Sensor Group Identifier	G*x* (e.g. G7)
Tp	Timestamp	*yyyy-mm-dd hh:mm:ss* (e.g. 2010-10-01 23:56:09)

In practice, the Event History is a database containing the list of basic events detected by sensors or cameras, tagged with a set of relevant attributes including detection time, event identifier, sensor identifier, etc. The detection time is a timestamp related to the event occurrence time, which should be a sensor timestamp (when a global clock is available for synchronization). A more detailed description of SMS and its sub-components is reported in [4].

The SMS is a highly heterogeneous security system, providing automatic event detection based on multi-sensor data. Advanced mechanisms for event correlation and for the detection of possibly complex threat scenarios could greatly augment SMS capabilities to improve detection reliability and to enable early warning and decision

support. The same holds for many similar and SCADA-like[1] distributed monitoring systems.

3 Detection of threat scenarios

In order to detect threat scenarios of any type and complexity, the SMS needs to be enriched with an expert system providing event correlation mechanisms. Many approaches proposed in literature cope with the problem of correlation, in particular of different alerts signals [10]. However, they are generally suitable to computer network monitoring systems. In physical security monitoring applications, it is more difficult to find systems that include advanced correlation features. To the best of our knowledge no existing physical security monitoring system features a scenario-based detection approach like the one described in this paper. This section briefly describes in particular the DETECT (DEcision Triggering Event Composer & Tracker) approach, which was firstly introduced in [11] and [12].

Threats scenarios are described in DETECT using a specific Event Description Language (EDL) and stored in a Scenario Repository. In this way we are able to store permanently all scenario features in an interoperable format (i.e. using XML). A high level architecture of the framework is depicted in Fig. 2.

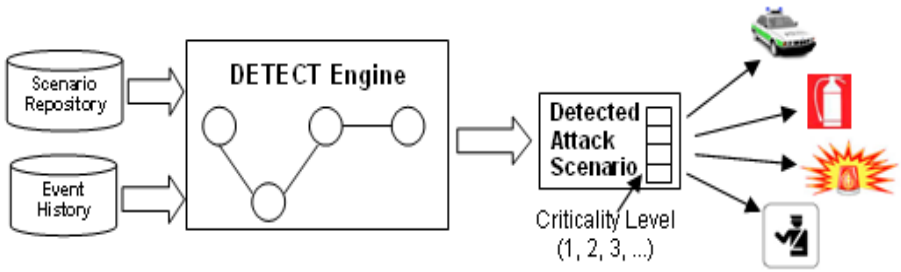

Fig. 2. The DETECT framework.

A threat scenario expressed by EDL consists of a set of basic events detected by the sensing devices (primitive events) which occur in a predictable sequence (with possible variants). The approach supposes an a-priori knowledge about the possible threats that are coded by a set of scenarios: they are identified during Vulnerability Assessment, which is a fundamental phase of Risk Analysis for critical infrastructures.

Generally speaking, an event is a happening that occurs at some location and at some point in time. In the considered context, events are related to sensor data variables (i.e. variable *x* greater than a fixed threshold, variable *y* in a fixed range, etc.). Events are classified as *primitive events* and *composite events.*

[1] SCADA stands for *Supervisory Control And Data Acquisition*. It generally refers to a system that collects data from various sensors, and then manages and controls them. They are often used to monitor and control industrial, infrastructure, or facility-based processes.

A primitive event is a condition on a specific sensor which is associated to some parameters (i.e. event identifier, time of occurrence, etc). A composite event is a combination of primitive events by means of proper operators.

Each event is denoted by an *event expression*, whose complexity grows with the number of involved events. Given the expressions $E_1, E_2, ..., E_n$, every application on them through any operator is still an expression. Event expression are represented by *event tree*, where primitive events are at the leaves, while internal nodes represent EDL operators.

According to the context in which a multimedia surveillance system – like the SMS described in previous section – works, the possible threat scenarios to be addressed can be more or less complex on the basis of the number of basic events that is necessary to correlate to detect them. In addition, to improve the detection reliability of such systems, two major techniques are typically adopted: *redundancy* (i.e. the use of more sensors than the ones strictly required) and *diversity* (i.e. the use of detection devices based on different technologies or working principles). Thus, depending on the circumstances, the side effect of these techniques may be a further explosion in the number of involved events in the expressions, and so in the event trees. This possible complexity is to be managed properly, by means of an ad-hoc complex event correlation framework, like the DETECT platform proposed in this section.

DETECT is able to support the composition of complex events in EDL through a *Scenario GUI* (Graphical User Interface), used to draw threat scenarios by means of an intuitive formalism and a user-friendly interface.

Furthermore, in the operational phase, a model manager macro-module has the responsibility of performing queries on the Event History database for the real-time feeding of detection models corresponding to the scenarios, according to predetermined policies. The latter are named *parameter contexts* and are used to set a specific consumption mode of the occurrences of the events collected in the database, as described in the following.

The EDL used in the event correlation framework to build event trees is based on the Snoop event algebra [13]. The tree construction is carried out according to the following Snoop operators: OR, AND, ANY, SEQ. Once the trees are built, the related detection models are ready to be inserted in the correlation engine. After that, the engine is fed by the primitive events gathered by the sensorial subsystems on site, and so it is able to recognize the partial or total matching with the known attack scenarios. Fig. 3 shows a sample event tree, where the leafs E_1, E_2, E_4, E_6 are primitive events and the operators are internal nodes. This tree represent the composite event $((E_1\ OR\ E_2)\ AND\ (E_2\ SEQ\ (E_4\ AND\ E_6)))$.

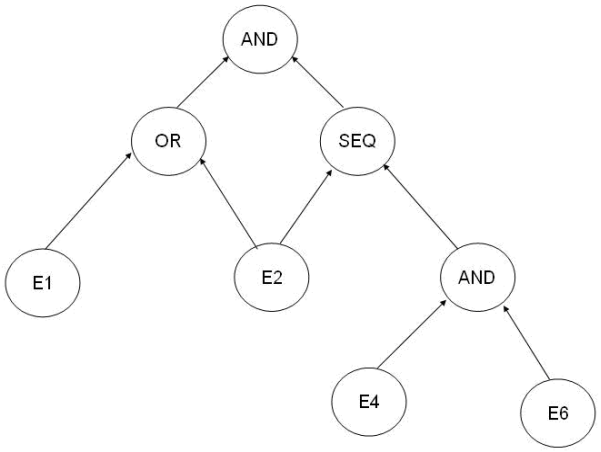

Fig. 3. A sample event tree.

The semantics of the Snoops operators is the following:

- **OR**. Disjunction of two events E_1 and E_2, denoted $(E_1\ OR\ E_2)$. It occurs when at least one of its components occurs.
- **AND**. Conjunction of two events E_1 and E_2, denoted $(E_1\ AND\ E_2)$. It occurs when both events occur (the temporal sequence is ignored).
- **ANY**. A composite event, denoted $ANY(m, E_1, E_2, ..., E_n)$, where $m \leq n$. It occurs when m out of n distinct events specified in the expression occur (the temporal sequence is ignored).
- **SEQ**. Sequence of two events E_1 and E_2, denoted $(E_1\ SEQ\ E_2)$. It occurs when E_2 occurs provided that E_1 has already occurred. This means that the time of occurrence of E_1 has to be less than the time of occurrence of E_2.

Furthermore, *temporal constraints* can be specified on operators, in such a way to consider that the logic correlations could loose meaningfulness when the time interval between component events exceeds a certain threshold. The aim is to define a validity interval for the composite event.

In order to take into account appropriate event consumption modes and to set how the occurrences of primitive events are processed, four parameter contexts are defined. Given the concepts of *initiator* (the first constituent event whose occurrence starts the composite event detection) and *terminator* (the constituent event that is responsible for terminating the composite event detection), the four different contexts are described as follows.

- *Recent*: only the most recent occurrence of the initiator is considered.
- *Chronicle:* the (initiator, terminator) pair is unique. The oldest initiator is paired with the oldest terminator.

- *Continuous:* each initiator starts the detection of the event.
- *Cumulative*: all occurrences of primitive events are accumulated until the composite event is detected.

The effect of the operators is then conditioned by the specific context in which they are placed.

When a composite event is recognized, the output of DETECT consists of:

- the identifier(s) of the detected/suspected scenario(s)[2];
- the temporal value related to the occurrence of the composite event (corresponding to the event occurrence time *tp* of the last component primitive event, which should be a sensor timestamp);
- an alarm level, associated to scenario evolution (used as a progress indicator and set by the user at the time of construction of composite event);
- the component event occurrences which have determined the recognition of the detected/suspected scenario(s);
- possibly other information depending on the detection model (e.g. likelihood of attack in case of heuristic detection, currently not yet implemented);

Fig. 4 shows an example screenshot of the output of DETECT when the correlation engine is activated.

[2] The difference between detected and suspected scenario depends on the partial or total matching between its tree and the known attack scenarios.

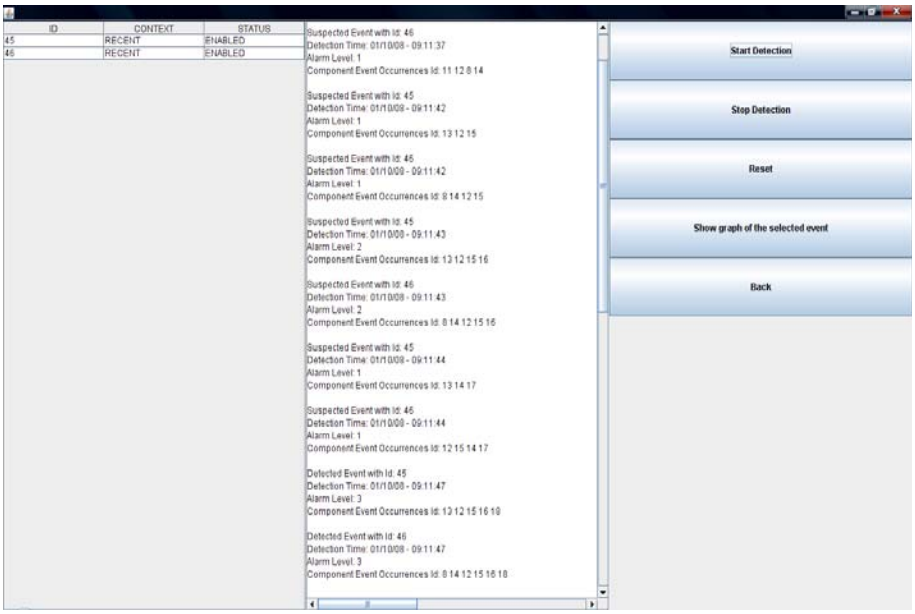

Fig. 4. An example screenshot of the output of DETECT.

4 An integrated system for the detection of distributed attacks

This section provides some details about the integration of the SMS with DETECT, in order to enrich the surveillance system with basic but effective reasoning capabilities.

In the integrated environment (see Fig. 5), DETECT and SMS share the Event History database and communicate by exchanging alert messages (from DETECT to SMS) and possibly commands (from SMS to DETECT). The commands consist of specific feedback from human operators which can be used to refine or update the detection models handled by DETECT.

On the one hand, SMS collects all alarms detected by the heterogeneous sensorial subsystems and store them into the shared database. On the other hand, the engine of DETECT is fed by each new entry in the Event History. The interface mode with the database can be synchronous (i.e. by means of queries performed periodically) or asynchronous (i.e. by event-based queries). In the first case, it is necessary to choose an appropriate time interval representing a good trade-off between the response time (short inter-query times are better) and the network/server load (long inter-query times are better).

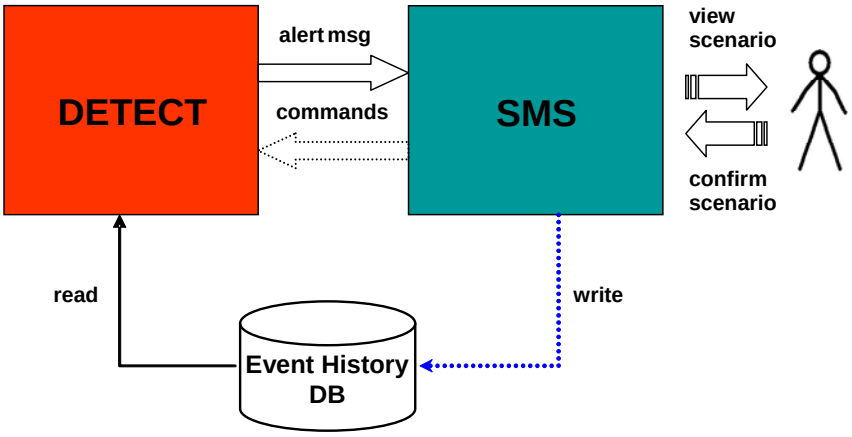

Fig. 5. The integration between DETECT and SMS.

The DETECT alert message need to be shown on a dedicated view on the SMS operator interface. Such a view gives to the security personnel information about the composite event that has been detected: semantic indication about the situation (explosive in tunnel, chemical attack, etc.), current phase according to the scenario evolution, confidence level of the alarm (if available), criticality level of the attack scenario (a static information obtained during the risk assessment process). If the detected scenario includes primitive events that have been already notified, SMS can drop them from the list of the alarms after confirmation by the operator. The composite event can then be stored in the Event History.

Depending on the specific user configuration required, primitive events can continue to be shown in a hierarchical or tree structured view (e.g. by double clicking on the scenario). According to the static (e.g. criticality level) and dynamic (e.g. evolution level) parameters of the attack scenario, the DETECT alarm may activate specific SMS procedures that will override procedures possibly associated with primitive alarms. In fact, featuring an intrinsic lower level of reliability, alarms from single sensors need to be verified more carefully by the operator, while composite events could even trigger automatic countermeasures, as it will be shown in the example of the following section. Fig. 6 shows an example of an operator interface including different screenshots of the integrated system. In particular, they show: the list of alarms with relative procedures (up), a vector graphics map which helps the operator to localize the source of the alarms (middle), the video streams automatically activated when an alarm is generated by smart-cameras or other sensors (down). The additional screenshot regarding the list of detected/suspected composite events is already showed in Fig. 4.

Fig. 6. An example of operator interface of the integrated system.

As stated previously, the SMS architecture is distributed and hierarchical. This configuration can be repeated also for the DETECT architecture in order to make possible the detection of simultaneous and distributed attack, which could not be recognized otherwise. In fact, only having a global view on the current status of all

peripheral sites, it is possible to consider specific critical events. As a matter of fact, although they may be unlikely and/or apparently not meaningful from a local viewpoint, they may assume a different and concrete importance from a global viewpoint. This is especially true in case of a simultaneous occurrence of the same event in more places.

5 An example scenario

This section reports an example of application of the overall approach to a case-study in a metropolitan railway environment. Historically, these mass transit systems, being easy to access public places, are vulnerable to many threats of various kind and seriousness. In fact, they can be theater of criminal acts, aggressions, vandalism as well as sabotages and terrorist strikes. The following is a description of how to detect complex scenarios of terrorist attacks by exploiting heterogeneous sensing devices.

As already mentioned, modern smart-surveillance systems suitable for the protection of metro railways are made up by several non fully reliable sensorial subsystems. When single alarms are not reliable, automatic countermeasures cannot be activated and operators response is slowed down. Mechanisms of alarm correlation can contribute to reduce the FAR (False Alarm Rate) and at the same time improve the POD (Probability of Detection). Improvements in detection reliability can be achieved adopting two main techniques: redundancy and diversity.

Through complex computer vision algorithms, the video analytics allows for the detection of events of different complexity, like intrusions in critical areas, abandoned objects, abnormal behaviors (person running or loitering, downfalls, etc). Since the detection of an event can suffer from the intrinsic reliability of the algorithm, as well as from issues due to environmental conditions (e.g. changes of lighting, presence of reflective materials, occlusions), **redundancy** in cameras dislocation can improve detection reliability and overall system resiliency against both accidental and intentional faults. For example, assuming the use of more intelligent cameras with overlapped views from different viewpoints to detect an abnormal behavior in a platform, the events detected by each camera can be combined with a simple AND logic.

However, the most interesting application of redundancy is when it is used in combination with **diversity**, by exploiting devices based on different technologies. In the assumption that the abnormal behavior includes screaming, which is detectable by means of appropriate audio sensors, the information coming from the microphone and the cameras installed in the platform can be combined using a more complex approach, based on the use of advanced logical and temporal operators.

Let us suppose to address a chemical attack, similar to what happened in the Tokyo subway on March 20, 1995 using Sarin gas. Sarin is a chemical warfare agent (CWA) classified as a nerve agent. It is a clear, colorless, odorless, and tasteless liquid in its pure form, and can evaporate and spread in the environment very quickly.

The current available technologies to identify the contaminated areas, for example include Ion Mobility Spectroscopy (IMS), Surface Acoustic Wave (SAW), Infrared Radiation (IR), etc. They are employed in ad-hoc standoff detectors and each of them

is characterized by different performances. One of the most accurate device, the automatic scanning, passive, infrared sensor can recognize a vapor cloud from several kilometers with an 87% detection rate [14]. Thus, to improve sensitiveness and reduce the number of false alarms, different technologies are often integrated in the same standoff detector (for example, the IMS and SAW detection are typically combined). More in general, it is possible to combine heterogeneous detectors and to correlate their alarms (e.g. IMS/SAW and IR detectors), in such a way to get an early warning system for the detection of chemical agents. Exploiting the redundancy and diversity also of these devices, increasingly complex correlations (logic, temporal, and spatial) can be implemented.

A likely scenario consists of a simultaneous drop of CWAs in many subway platforms in the rush hour. Let us suppose that dynamic of events is the following:

1. the attackers stay on the platforms, waiting for the simultaneous drop of CWA;
2. the first contaminated people fall to the floor;
3. the people around the contaminated area run away and/or scream;
4. the CWA quickly spread in the platform level and reach the escalators to the concourse level.

In each subway site, it is possible to address the attack scenario by means of two intelligent cameras positioned at platform end walls, a microphone between them, two standoff detectors for CWAs positioned on the platform and on the escalator.

The scenario can be formally described by means of the notation "sensor description (sensor ID) :: event description (event ID)":

Intelligent Camera (S1) :: Fall of person (E1)
Intelligent Camera (S1) :: Abnormal running (E2)
Intelligent Camera (S2) :: Fall of person (E1)
Intelligent Camera (S2) :: Abnormal running (E2)
Audio sensor (S3) :: Scream (E3)
IMS/SAW detector (S4) :: CWA detection (E4)
IR detector (S5) :: CWA detection (E4)

Given the scenario described above, the composite event **drop of CWA in platform** can be represented by the event tree in Fig. 7, built using the DETECT framework.

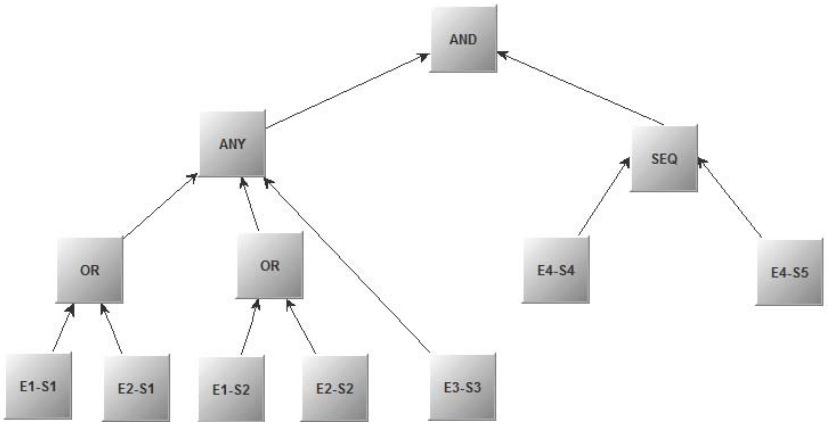

Fig. 7. Event tree associated to “drop of CWA in platform”.

A partial alarm level (e.g. 1) can be associated to the scenario evolution in case of occurrence of the ANY event (at the left of tree). The *m* parameter of ANY is set to 2 (trough the Scenario GUI) , this means that when 2 out of 3 distinct events detected by intelligent cameras and/or microphone occur, the monitored situation is considered abnormal (in fact each of the single events: person who falls, runs or scream, can not represent a meaningful state of alert). Besides, a temporal constraint can be set on ANY operator, in such a way to catch real alarm conditions: e.g. if fall and scream are detected at a distance of time of 30 minutes, that could not represent an alert condition for the specific scenario. In the specific example, it could be set to 5 minutes to take into account the latency of both gas propagation and intoxication symptoms.

An higher alarm level (e.g. 2) can be associated to the scenario evolution in case of occurrence of SEQ event (at the right of tree). The use of the sequence operator is due to the different assumed locations of the CWA detectors: IMS/SAW detector at platform level, IR at escalator or concourse level, in such a way to detect correctly the spread of CWA. If IR detector gives a warning before the one based on IMS/SAW detection, this could be an abnormal condition due to a false alarm and should not cause the activation of a warning. To further avoid false alarms, also a temporal constraint should be set. In this case, it can be set to 10 minutes to be conservative while taking into account the movement of air flows between different environments.

Finally, it is necessary to set the parameter context to regulate the consumption mode of the occurrences of events in feeding the detection engine. In this case, the assumption is that only the most recent occurrence of each event is meaningful. Thus, parameter is set to “recent context”.

The use of many alarm levels can be useful to trigger countermeasures properly: e.g. the alarm level 1 can trigger the opening of the turnstiles; at level 2 an appropriate ventilation strategy can be activated; finally, the detection of the whole composite event can be associated to actions like: evacuation message from public address, stop trains from entering the station, and emergency call to first responders.

Assuming the simultaneous use of the correlation engine in each peripheral sites and in the main control center, it is possible to address strategic terrorist attacks

(which often feature simultaneous strikes). In the considered example, if the control center detects the simultaneous (possibly partial) evolution of the above described scenario in different subway platforms, then the evacuation of the involved stations and the block of train traffic could be triggered immediately. This approach can enable an advanced situation awareness, early warning and decision support. Accordingly it is possible to improve the impact of countermeasures in a significant way. As stated in previous section, this is the key to detect simultaneous and distributed attacks, which could not be recognized otherwise, and to react promptly. The hierarchical architecture of the integrated monitoring system, including both SMS and DETECT, is functional for this purpose.

6 Conclusions and future work

The paper describes an approach to augment the capabilities of distributed surveillance systems in order to better support security operators in responding to threats. The approach is based on the DETECT framework, which implements a model-based detection engine, currently limited to Event Trees but suitable to accommodate different detection models. The decision making of the operators is supported by indications of suspect scenarios with increasing alarm levels in order to better and quickly discriminate between false and real alarms. Operators can then bias their behavior accordingly, guided by custom event management procedures.

In particular, further efforts are going towards the enrichment of the framework with stochastic extensions, by associating sensor events with reliability parameters (e.g. probability of false alarms) and automatically computing the reliability of composite events. That would help the operators to also be aware of the real level of reliability of the detected scenario.

References

1. Garcia, M.L.: The Design and Evaluation of Physical Protection Systems. Butterworth-Heinemann (2001)
2. Goldgof, D.B., Sapper, D., Candamo, J., Shreve, M.: Evaluation of Smart Video for Transit Event Detection. Project #BD549-49, FINAL REPORT, http://www.nctr.usf.edu/pdf/77807.pdf (2009, last access January 6th 2010)
3. Martin, P.T., Feng, Y., Wang, X.: Detector Technology Evaluation, http://www.mountain-plains.org/pubs/pdf/MPC03-154.pdf, (2003, last access January 6th 2010)
4. Bocchetti, G., Flammini, F., Pragliola, C., Pappalardo, A.: Dependable integrated surveillance systems for the physical security of metro railways. In: IEEE Procs. of the third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), pp. 1--7 (2009)
5. Zhu, Z., Huang, T.S.: Multimodal Surveillance: Sensors, Algorithms and Systems. Artech House Publisher (2007)
6. Wickens, C., Dixon, S.: The benefits of imperfect diagnostic automation: a synthesis of the literature. In: Theoretical Issues in Ergonomics Science, 8(3), pp. 201--212 (2007)

7. Cucchiara, R.: Multimedia Surveillance Systems. In: Proceedings of the third ACM international workshop on Video surveillance & sensor networks (2005)
8. Flammini, F., Gaglione, A., Mazzocca, N., Moscato, V., Pragliola, C.: Wireless Sensor Data Fusion for Critical Infrastructure Security. In: Advances in Intelligent and Soft Computing, Volume 53, pp. 92--99 (2009)
9. Flammini, F., Gaglione, A., Ottello, F., Pappalardo, A., Pragliola, C., Tedesco, A.: Towards Wireless Sensor Networks for Railway Infrastructure Monitoring. In: Proc. ESARS 2010, pp. 1--6, Bologna, Italy (2010)
10. Pouget, F., Dacier, M.A.: Alert correlation: Review of the state of the art. Technical Report RR-03-093.
11. Flammini, F., Gaglione, A., Mazzocca, N., Pragliola, C.: DETECT: a novel framework for the detection of attacks to critical infrastructures. In: Safety, Reliability and Risk Analysis: Theory, Methods and Applications, Martorell et al. (Eds), Procs of ESREL'08, pp. 105--112 (2008)
12. Flammini, F., Gaglione, A., Mazzocca, N., Moscato, V., Pragliola, C.: On-line integration and reasoning of multi-sensor data to enhance infrastructure surveillance. In: Journal of Information Assurance and Security (JIAS), Vol. 4, Issue 2, pp. 183--191 (2009)
13. Chakravarthy, S., Mishra, D.: Snoop, An expressive event specification language for active databases. Data Knowl. Eng., Vol. 14, No. 1, pp. 1--26 (1994)
14. Davis, G.L.: CBRNE - Chemical Detection Equipment. eMedicine, http://emedicine.medscape.com/article/833933-overview, (2008)