ABSTRACT
Modern surveillance systems embody more stationary and mobile sensors, actuators and analytics software to cover wider areas of surveillance thus leading to more complex surveillance systems than ever before. As the systems tap into larger areas of surveillance, automation will also play a more significant role in intelligently reacting to threatening events. To tackle complexity, intelligent filtering and alarm semantic enrichment mechanisms for alarms are required to allow security personnel to concentrate on important events instead of redundant and meaningless alarms. Additionally, an automated planner is required to provide optimal solutions on reacting to events occurring in the surveillance area. In this paper we present a software component capable of spatio-temporal reasoning and planning to be integrated with a third generation surveillance system. The Milestone XProtect platform is utilized as an example legacy surveillance system.

KEY WORDS
Knowledge Acquisition, Expert Systems, Complex Event Processing, Constraint Satisfaction Problem

1. Introduction
Third Generation Surveillance Systems (3GSS) for public safety are built upon distributed architecture of networked components for sensing, controlling, communication and analytics [1]. Building blocks of a typical modern surveillance system include 1) a network of geographically distributed sensors, 2) an IP (Internet Protocol) based transport network and 3) back-end components for data storage and analytics [1][2].

While surveillance in 3GSS primarily focuses on real-time analysis of video feeds, analytics and decision making on how to react to occurred events are also often supported by a range of multi-modal sensors such as audio, biometrics and positioning [2]. These sensors may be equipped with on-board intelligence or analytics capabilities, and they most often communicate using IP-based, potentially wireless, networks. The use of wireless networks facilitates the exploitation of mobile sensors and agents on the field, creating necessity of utilizing real-time and location-specific information in analytics and decision-making [1].

Sensors are typically considered as independent data sources towards the back-end analytics components where the data is analysed and alarms are triggered if significant events have been detected. However, events occurring in a certain area do not necessarily occur isolated but may have causalities and correlations between each other. These types of events are said to be temporally related. Events related via space rather than temporally are called spatially related. Combinations of the aforementioned events are called spatio-temporally related events. Events comprising of multiple events are known as complex events whereas the process of reasoning about these events is called Complex Event Processing (CEP).

Considering spatio-temporal relations between events is necessary particularly when surveillance areas grow larger as human security personnel involved in wide-area surveillance can easily be exposed to information overload and redundant alarms [3]. This may cause distractions and result in overlooking the alarms of importance. Deploying an intelligent filtering and alarm enrichment mechanism would provide only meaningful events to the surveillance personnel and enable them to concentrate on high-priority alarms.

Considering events occurring in the surveillance area as a whole instead of disparate events enables more intelligent automated reactions. Certain reactions may be simple, i.e., for a certain individual event a specific action has to be taken. However, more complex scenarios to resolve an event are plausible especially when mobile agents are utilized, or when numerous ways exist to resolve an event by considering the available resources. For instance, the closest police patrol to be dispatched to further investigate an event may already be involved in ongoing tasks while patrols further away may be too far to respond quickly enough. In this situation task priorities dictate how to proceed. Choosing an optimal set of agents
to resolve an event easily becomes a planning problem with a huge search space, in certain cases even an NP problem [4].

In this paper, we present a component for introducing spatio-temporal CEP including planning to 3GSS. The commercially available Milestone XProtect surveillance system\(^1\) is utilized as a platform. The presented component also takes into account mobile sensors which Milestone XProtect does not currently fully support.

This paper is structured as follows. First, Section 2 sets the context for the work presented in this paper by providing an overview on 3\(^{rd}\) Generation Surveillance Systems by introducing Milestone XProtect with its relevant capabilities as an example. Based on that, spatio-temporal reasoning, CEP and solving constraint satisfaction problems in this context are discussed in Section 3. In Section 4, we describe the conceptual architecture for enabling spatio-temporal reasoning, inference and planning capabilities in Milestone XProtect with available tool support. Finally, discussion and conclusions close the paper.

2. Milestone XProtect as a 3\(^{rd}\) Generation Surveillance System

Typical 3GSS configurations utilize central back-end systems to provide database and analytics capabilities for a network of distributed sensors and actuators [1][2][5]. In addition, the back-end provides connections to user interfaces for human operators. As surveillance systems evolve towards automated intelligence, they have increasingly been utilized for automated analytics. Analysis techniques include object detection, object recognition, object tracking as well as more complex behavioural analysis [5]. As modern sensors themselves can feature some degree of onboard intelligence and analysis capabilities [2], the back-end systems can receive data which describes events detected by the sensor in addition to raw media feeds. The analytics components can also provide automated reactions for detected events [1]. Perhaps the most typical action taken is raising alarms for human personnel once an event has been detected [1][5][6]. Other possible types of actions to take include e.g. issuing PTZ (pan-tilt-zoom) commands on camera units to improve the situational coverage of the event, and operating electric lock mechanisms.

The Milestone XProtect surveillance platform is an example of a 3GSS back-end system which supports recording, analyzing and relaying surveillance data. The platform provides recording services for multiple IP and analog cameras, along with a uniform interface for video analytics software to process and analyze the recorded video streams. The analytics components may provide augmentations to the video streams and raise alarms in the system. The surveillance personnel can then monitor multiple video streams with these augmentations in addition to monitoring the triggered alarms. The alarms and the events might raise more alarms via alarm chaining and then again generate predefined actions for the connected sensors and actuators. Figure 1 represents the architecture which is discussed in detail below.

![Figure 1. Milestone XProtect in its context.](image)

2.1 Alarms and Events

The reaction mechanism Milestone utilizes is based on the following types of events and alarms: 1) time triggered events, 2) manual events, e.g. triggered by security personnel with provided additional monitoring applications, 3) hardware events, i.e. a physical device, often a camera, raises a signal to trigger an event, and 4) analytics events from an external application integrated via the Milestone SDK (software development kit). In addition to these, 5) the Milestone Alert Data (MAD) format is utilized. MAD is a special type of an event which can be utilized for enabling video stream augmentation in order to support surveillance personnel to better spot e.g. the cause of an event or detected objects. MAD events are typically sent multiple times whereas aforementioned events are usually triggered only when a significant incident has occurred.

The events can trigger the system to react by: 1) raising an alarm which can trigger an event and in this way form a chain of reactions, 2) causing a hardware output event which might be in form of a signal raised on e.g. camera input ports, 3) triggering a camera to e.g. start or stop recording, 4) notifying security personnel via a monitoring application, and 5) an alarm mechanism which provides security personnel with more detailed information about an occurred situation.

\(^1\)www.milestonesys.com
2.2 Control Functionalities
Cameras, microphones and speakers can be controlled by external software components via Milestone SDK. Microphones and speakers can be enabled or disabled and data can be transmitted between these two devices. If available, PTZ cameras can be controlled. Additional devices can be attached to IP camera output ports and controlled via Milestone SDK. Additionally Milestone XProtect internal functions can be applied to trigger actions in the attached devices.

2.3 Video Analytics as an Example
The aforementioned capabilities can be illustrated using the following example scenario. A camera is installed at an entrance to a restricted area at a governmental building. During night this area must not be accessed whereas during daytime only staff holding a special permission is allowed to enter. Still pictures of the person entering the area and a corresponding video snippet showing the incident are forwarded to the security personnel.

To enforce the aforementioned access control, video analytics in form of face detection and recognition is applied. The input video stream for the video analytics is directly retrieved from Milestone XProtect. As soon as the face detection algorithm has detected a face, a Milestone Analytics Event is created by utilizing Milestone SDK. The event contains relevant information about the event such as its type (in this case it might be “person entering”), identification code for the face and camera location (e.g. “main entrance, camera 1”). The event triggers the recording functionality to record a snippet of the specific video stream containing the detected incident, e.g. for further processing or in order to extract still pictures of the intruder’s face to forward it to the security personnel. Furthermore an alarm is generated to notify security personnel about possible trespassing. MAD is utilized to augment the video, containing e.g. coordinates for drawing bounding boxes around the detected face. This geometric data is then displayed in both the live and recorded videos as an overlay to spot events more quickly.

The driving force for reasoning is situation awareness. Endsley et. al. [7] describe it as “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future”. In other words spatio-temporal properties are the driving force for reasoning.

Knowing spatio-temporal relations between the events enables making intelligent decisions about the resolved knowledge. When surveillance areas get wider and numerous resources with different capabilities are available, careful planning of how to resolve the situations optimally becomes cumbersome particularly when it is required in addition fulfilling the hard goals, i.e. goals that have to be satisfied, optimize satisfying multiple levels of soft goals, too. The problem of planning becomes a constraint satisfaction problem.

In the following subsections, means to express spatio-temporal relations and means to handle CEP with ontology-based approach and production rule system based approach are briefly discussed in order to provide background knowledge about the technology selections later discussed in this paper. After these, constraint satisfaction problem is introduced.

3.1 Spatio-Temporal Reasoning
In order to describe spatio-temporal properties a description technique is needed. OWL-DL² is a language typically utilized for describing concepts. It can be used to represent temporal properties while it is not straightforward [8]. In [9], a technique, which enhances the 4D fluent mechanism to represent spatio-temporal properties, is introduced. The technique claims that all Allen’s temporal relations [10] as well as topological and directional spatial relationships between the events via Region Connection Calculus (RCC-8) [11] can be inferred.

In addition to the ontology-based description techniques, production rule systems, among others [12], such as JBoss Drools³ exist to describe spatio-temporal relations. In OWL-based (Web Ontology Language) approaches spatio-temporal relations can be directly expressed and reasoned with by applying OWL concepts and available reasoners. However, in rule-based approaches the description technique for the both is dependent on the rule language. In Drools, temporal reasoning supporting all Allen’s temporal relations [10] is provided by the rule language but spatial reasoning and classification have to be developed using the rules themselves. This is similar to OWL-based approaches where the ontological means have to be applied to express spatio-temporal relations.

---
² www.w3.org/TR/owl2-overview
³ www.jboss.org/drools
3.2 Complex Event Processing

CEP is essential for making decisions about events considering their correlations rather than considering events as unrelated facts in the system. Production rule systems, such as JBoss Drools, are well suited for CEP as declarative rules see all facts in the working memory of the applied reasoning tool and thus can consider their relations by applying the facilities provided by the rule language.

Considering OWL-based approaches, OWL accompanied with SWRL\(^4\) (Semantic Web Rule Language) can also be applied for CEP as discussed in [13] and [14]. Facts are inserted into the knowledgebase similarly to Drools and the SWRL rules trigger to process the knowledge. This kind of approach provides natural technique to connect the events to the domain ontology and in this way enable developing rules which take into account all facilities provided by OWL. In the rule-based approach, connecting the events to the ontologies depends on the rule language.

3.3 Constraint Satisfaction Problem

Finding a solution for an event with constrained resources might not be straightforward notably in case of multiple variables and multiple hard and soft goals to fit. Particularly when there are multiple events to solve at the same time, careful balancing has to be performed especially when there might not be enough resources to fulfill all requirements. Thus, compromises have to be made while an optimal solution considering the available resources has to be found.

In certain cases constraint satisfaction problems may even be NP-complete problems. Therefore, brute force techniques are not feasible and simple algorithms that would strive to fill the most significant goals first rarely return an optimal solution. Advanced optimization algorithms exist to prune the search-space quickly compared to the brute-force approach. OptaPlanner\(^5\) encapsulates a set of optimization algorithms such as First Fit, Best Fit and Cheapest Fit as construction heuristics and Hill-Climbing, Tabu Search and Simulated Annealing to name a few of the supported metaheuristics. OptaPlanner utilizes Drools for scoring the possible solutions which as a combination shortens the time needed to find an optimal solution within the set time limits. In addition to utilizing optimization algorithms for local search-based methods there is a form of declarative programming languages called Answer Set Programming (ASP) [15] which is oriented towards solving NP-hard search problems by applying backtracking-based search algorithms.

4. Intelligent Surveillance System

Our proposed solution for empowering a 3GSS back-end system such as Milestone XProtect with advanced inference and planning capabilities is depicted in Figure 2. The conceptual architecture applies the Layered Architecture style with surveillance units, e.g. cameras and police patrol cars providing data and alarms to Milestone XProtect, at the bottom. In case of mobile surveillance units the devices will also provide their position information to Position Manager to track them. Position Manager in cooperation with the Converter component enriches the Milestone XProtect alarms with position information among others and forwards the enriched events to the Inference Layer. The Inference Layer is responsible for event classification and CEP to provide more insight to the events occurring in the surveillance area. The events that are considered to be alarming are forwarded to the Planning Layer where they are inspected and counter actions are generated. The latter are then forwarded to Milestone XProtect to enable device controlling and to inform the security personnel with means provided by the surveillance platform. Throughout this section, the scenario presented in Section 2.3 forms a starting point for introducing advanced reasoning and planning to the system.

\(^4\) www.w3.org/Submission/SWRL

\(^5\) www.optaplanner.org

Figure 2. Conceptual architecture for enabling spatio-temporal reasoning and planning.
4.1 Tools for Inference and Planning

There are numerous tools for reasoning and inference as discussed in [12]. For resource constraint satisfaction problems there are tools supporting optimization algorithms such as OptaPlanner or DLV-Complex\(^6\) for ASP. Deciding which are the optimal toolset and paradigms for enabling the aforementioned kind of intelligence would be of its own topic of discussion. For instance, it is not a clear-cut decision whether to choose an ontology-based approach over a rule-based approach as discussed in [16]. Drools and OptaPlanner combined have been selected for implementing our architecture as they provide a set of features needed by the architecture. The same technology foundations between the two also speak of utilizing them as a combination to reduce compatibility and integration issues. The drawback is that support for e.g. classification via ontologies has to be implemented with rule-based approach instead of utilizing any of the widely available OWL reasoners. On the other hand temporal relations are natively supported by the rule language whereas in OWL it requires the adaptation of the utilized ontology [8].

JBoss Drools is selected as a rule engine as it provides an intuitive technique to express and reason with temporal relations between events. Drools comprises of a set of tools where the most essential is the rule engine Drools Expert. The rule engine implements the Rete algorithm [17] for efficient pattern matching which enables Drools to scale for large-scale rule-bases and for handling multiple events.

Drools provides its own rule definition syntax which does not conform to any standard. Facts are inserted to the Drools working memory as POJOs (Plain Old Java Object) and they can be queried by conducting queries or by applying an observer to subscribe the changes in the working memory.

The rule language is broad and expressive and allows embedding custom evaluation code. Drools also supports the concept of time via the Drools Fusion module which has been developed particularly for handling time-dependent complex events. For instance, a sliding window can be utilized to limit the application of a set of rules only to the facts that have occurred within the window. Currently Drools does not natively provide support for spatial reasoning whereas the technique presented in [8] does. However, Drools offers a possibility for implementing custom functions which can be taken advantage of to express and compute spatial relations.

OptaPlanner is selected for solving constraint satisfaction problems. It is a tool for efficiently finding solutions for constraint satisfaction problems by providing a set optimization algorithms that shortens the time to find the solution compared to applying brute force means.

OptaPlanner utilizes Drools as its rule-engine to evaluate the solution candidates. As scores, OptaPlanner provides means to express hard goals and also unlimited amount of various levels of soft goals to satisfy. This means that it will optimize the solution by first finding the best match for hard goals and then starts to optimize the lesser goals.

4.2 Converting Alarms to Events

Considering spatio-temporal properties the alarm format utilized by Milestone XProtect system is insufficient. The alarms contain timestamps but e.g. interval types of alarms cannot be presented directly. Additionally, spatial properties are typically described just by location such as “main entrance, camera 1”. It is apparent that this technique does not scale nor allow a location to be defined precisely in order to enable advanced spatial reasoning. In particular for mobile sensors this technique is not suitable. Thus, Milestone XProtect alarms have to be enriched with additional information to produce better comprehension of spatial alarm properties.

In our design, the Alarm2Events Layer is responsible for subscribing alarms from Milestone XProtect via its SOAP (Simple Object Access Protocol) API (Application Programming Interface) and to enrich alarms with position information. The position information for alarms is provided with a configuration file.

An alarm originating from a specific source, e.g. “main entrance, camera 1”, can be mapped with a configuration file to a surveillance unit and its current position. Furthermore, a unit may be equipped with sensors which have their position related to the unit itself. When an alarm is received from Milestone XProtect its position will be mapped according to the unit position and then further to the pose of the related sensor. The sensor properties then describe the form of an event as a polygon. In this way it is possible to transform discrete observations into observations with shape which enable spatial reasoning.

In order to utilize mobile units, e.g. police patrol cars, Position Manager has to be made aware of unit mobility. Mobile surveillance units in the system must register their position information to Position Manager which then listens for periodical position updates from the units. Position Manager maintains a position history for

\(^{6}\) www.mat.unical.it/dlv-complex
each unit in order to find the best-fit pose for events when needed because position updates are not synchronized with the alarms received from Milestone XProtect.

This phase results in events containing position information and other additional. The processed events are now ready for further inference and post-processing.

4.3 Reasoning and Inference

Once an event is enriched with position information, it is forwarded to the Inference Layer. The responsibility of the inference layer is to provide more knowledge about the events occurring in the surveillance area through reasoning about the events, i.e. classification of the events, and CEP to detect complex spatio-temporal relations between them.

Reasoning in context of surveillance systems can comprise of all scenarios that e.g. OWL-DL can handle. We consider type inference to be the most important, i.e. an event of certain type is also of second type if it is in the same inheritance hierarchy. Multi-inheritance has to be supported in order to enable easy connection of various ontologies. Enforcing inheritance hierarchy enables further processing of events just by considering their super-classes instead of always having to refer to their exact classes. This makes writing and maintaining the rules easier. For instance, a “person entering” alarm detected during night time triggered by a camera deployed at the front of the main entrance door can be classified as “trespassing” which on the other hand is an “alarming” event.

Further inference of events requires taking into account temporal and spatial properties of the events because there might be no use of finding e.g. correlations between events which occur in different districts of a town. Typically, events have to occur within certain vicinity to each other in order to form a more complex meaning.

Consider a scenario where a camera deployed at the front of the main entrance has alarmed about trespassing, and some minutes later a backdoor camera triggers the same alarm. It can be inferred that these events most likely are related to each other particularly when the detected face of the possible trespasser is the same. Now, instead of having only two “trespassing” alarms, the system can infer that the trespasser is an intruder thus it generates an “intruder detected” alarm which is of type “alarm” to which has to be reacted upon.

4.4 Mission Generation

As soon as all relevant information is available, the events to be reacted on are forwarded to the Planning Layer. The Planner subsystem computes the most suitable actions to be taken by considering the events in its working memory and capabilities of the stationary and mobile surveillance units existing in the surveillance area. Such actions can be e.g. alarming the security personnel via simple notifications, generating exact tasks for the security personnel, or controlling cameras, electric locks and other devices.

There are two phases to be accomplished before a set of so-called missions, i.e. actions, can be generated: 1) find an optimal set of surveillance units which can handle the situation on hand, and 2) for that optimal set of surveillance units, generate missions that take into account the requirements to be fulfilled as set by an event and the capabilities of the set of available surveillance units. Figure 3 presents the division of the tools and the process to accomplish the aforementioned tasks.

![Figure 3. Planning and mission generation using Drools and OptaPlanner.](image)

4.4.1 Planning

In Figure 3, capabilities represent the capabilities a surveillance unit can have. For instance, PTZ camera has a capability to pan, tilt and zoom whereas a security person has a capability to stop a trespasser. Profile depicts the detailed information about an event via an event ontology. Recipe defines how certain surveillance unit capabilities can satisfy a certain event profile. For instance, to deal with an intruder a security person has to be alarmed to handle the situation, all nearby cameras have to be triggered to record, and nearby doors of the building have to be made sure they are locked.

Profile, capabilities and recipe are all added to OptaPlanner working memory during its initialization phase. Additionally, score calculation rules are included to its production memory. In this way, OptaPlanner can rapidly access all information required to find an optimal configuration within defined time limits. Adding these into the working memory also eases the maintenance of
its decision-making as no source code modifications are necessary to modify its behaviour.

The task of OptaPlanner is to find an optimal solution within predefined time limitations by considering available surveillance units and their capabilities, events and their profiles, and recipes that can solve the events. Not only OptaPlanner should find the surveillance units that can fulfill the hard goals, e.g. to catch a trespasser but also to satisfy a set of medium and soft goals and everything in between. For instance, estimated time and cost to reach the target location, and other insensitivities play a role in finding an optimal solution.

As a result of this phase, a set of event-surveillance unit pairs has been generated. These pairs present an optimal set of surveillance units to solve the event in question by considering all the goals set for scoring. For instance, to deal with an intruder, the closest security person is selected although he would be on his routine patrol in some other building, nearby cameras and electric locks are also selected.

### 4.4.2 Final Decision Making

The next step is to provide the event-surveillance unit pairs back to Drools for final decision-making. In case if hard goals were not able to be fulfilled, the ActionGeneration component decides upon the action to take, i.e. raising an alarm to the system regarding the unfulfilled configuration, or ignoring it and generating missions for the available surveillance units. Once the ActionGeneration component chooses to generate missions, it scrutinizes the available surveillance units’ capabilities and event profiles, utilizing them to generate missions to be issued for the surveillance units and surveillance personnel.

Missions are delivered to Milestone XProtect in the format of events and alarms, MAD data (see Section 2.1) and commands (see Section 2.2). Commands are utilized to directly control the related devices via Milestone XProtect. For instance, upon intruder detection, PTZ commands are sent to cameras to point to the position where the latest alarm was created and remaining related cameras are set to patrol mode. Furthermore electric locks are closed. MAD data is utilized to augment the video streams if the intruder is still in vicinity. An alarm is generated to notify the chosen security person about the incident. The generated alarm goes through the Converter to map e.g. location expressed in polygons to a human understandable form. As a result, devices will have their orders and security personnel are alarmed about the occurred event.

### 5. Discussion

Rules for reasoning and inference about the events are described in Drools rule files which can be loaded into the system even dynamically. In addition, the event ontology is also described through facts in the same rule files. Describing the knowledge in the rule files significantly eases the system maintenance as source code modifications are not required when knowledge about the events evolves. In rapidly changing conditions, e.g. during a large ongoing situations such as riots, rules for the rule-base can be added and removed with ease. This also applies to planning as recipes, capabilities and profiles are inserted as facts into OptaPlanner working memory which can be updated when knowledge evolves. Similarly, the scoring rules can be updated accordingly.

Concerning generating control commands for the devices and alarms for the security personnel, the presented component is largely dependent on features provided by the underlying surveillance platform. Although utilizing an existing surveillance platform as a central integration point has its pros, as cons it limits controlling mechanisms of devices and providing only alarms for surveillance personnel is not enough as they themselves still would need to consider how to react to possibly even complicated scenarios. Considering more dynamic future surveillance applications, means for controlling the surveillance units and means to provide assignments to them have to be improved. Thus, it would be worthwhile to provide the personnel elaborated missions accompanied by additional background information about the events in a form of e.g. video stream which is currently not possible with Milestone XProtect. The presented component should be enabled to directly communicate with the surveillance personnel and other devices to provide them adequate information instead of simple alarms and control commands. Unfortunately, this requires adaptation from the devices and other clients thus raising the cost in the deployment of an intelligent dynamic surveillance system.

Prototype of the presented system has been developed. Such a prototype has been designed to integrate with Milestone XProtect however the current effort concentrates on controlling surveillance units directly to enable developing more dynamic and intelligent surveillance system than it is now possible to develop with Milestone XProtect.

### 6. Conclusion

3GSS are typically built on distributed sensing and controlling components communicating over possibly
wireless IP networks with a back-end system for data storage and analytics. Intelligence for 3GSS is largely based on advanced video analytics which can enable the system to react according to the triggering events. However, events typically have so far been seen as disparate non-related incidents to react. With surveillance systems becoming complex and the areas under surveillance growing wider, events have to be considered as a set of spatio-temporally correlated incidents. Not only should advanced inference capabilities be available to improve the situation awareness, but reactions also have to be planned accordingly. In case of multiple and potentially mobile resources to take advantage of and various constraints to satisfy, planning becomes cumbersome when it is important to optimize the use of available resources.

In this paper, a component to provide advanced spatio-temporal reasoning and inference capabilities to an existing surveillance platform, Milestone XProtect, has been presented. The proposed component is capable of CEP but also automates action generation for surveillance units, personnel and devices, by finding optimal solutions for the raised incidents. It can be completely implemented in the back-end system. Thus, no modifications are required to an existing system in case when no mobile devices are applied. The presented component takes mobile devices into account but means for delivering the position information to the presented component has to exist.

Research on the intelligent surveillance architecture is ongoing. Areas of focus for the future work include ensuring the efficiency of spatial reasoning as well as applications of mission generation for different types of surveillance systems, units and actuators.

References