

Event-driven decision support system for intelligent emergency management in critical infrastructures

Vuk Mijović, Nikola Tomašević, Valentina Janev, Sanja Vraneš

University of Belgrade, The Mihajlo Pupin Institute, Volgina 15, 11060 Belgrade, Serbia

{vuk.mijovic, nikola.tomasevic, valentina.janev, sanja.vranes}@pupin.rs

Abstract Critical infrastructures (CI) are difficult to handle due to their complexity, size and number of involved stakeholders. During emergency situations (e.g. fire or terrorist attack), CI operator in the control room is faced with a flood of information coming from different sensors and legacy monitoring systems. Since in these situations, time is critical and the operators are under a great deal of pressure, a holistic management of all involved technical systems and actors is needed, reinforced by the Recommendation and Decision Support System (RDSS) that helps emergency managers to take right and timely decisions. One way to provide an adequate RDSS support to the operator is proposed in this paper which is based on an intelligent, event driven layer that sits on top of the legacy CI monitoring system. Powered by the complex event processing capabilities and facility data model implemented in form of CI ontology, this layer processes events originating from different sources, conducts the situation and risk assessment, and reacts accordingly, either automatically or via recommendations proposed to emergency personnel. To validate the proposed approach, an event-driven RDSS was deployed on the airport use case (Nikola Tesla airport in Belgrade), as one of the most complex CIs.

Keywords *emergency management; event-driven decision support; complex event processing; emergency personnel training.*

1 Introduction

Critical infrastructures (CIs), such as airports, railway and metro systems, power grids and telecommunication systems, oil and gas pipelines, are difficult to manage from the perspective of emergency management (EM) due to their complexity, size, and number of involved stakeholders. As a consequence, they rely on numerous technical systems that address different operational aspects, such as fire protection, heating, ventilation and air conditioning (HVAC), evacuation and access control, public address system, etc., by providing notifications about state changes and reacting automatically when appropriate. CI operators, sitting in control rooms, usually use the control and monitoring system in operation (such as SCADAs, BMSs) to supervise all technical systems and to issue commands and implement corrective measures. However, reasoning capabilities of these legacy SCADA systems are limited and they do not take into account all available information, either sensor streams or human provided information. Therefore, apart from improving the reasoning capabilities, integration of all available data sources and legacy systems would be one of the prerequisites and key drivers for future EM systems.

Furthermore, due to their complex infrastructure and layout, consisting of different types of premises such as waiting halls, offices, hallways, equipped with various technical devices and systems, CIs are very difficult to supervise and manage. Particularly during the emergency situations, underlying technical system (e.g. fire protection system, evacuation and access control system and public address system among others) generate an avalanche of alarms and events (sometimes even thousands per second). Therefore, it is quite difficult for the operator to obtain a clear overview on the ongoing situation and react appropriately. By processing these so-called atomic events and correlating them, the EM system should facilitate real-time assessment of the safety and security conditions, react automatically whenever possible and provide timely and objective decision support to emergency personnel. This is especially important in cases where automated action is not feasible or trustworthy, thus ensuring holistic management of underlying technical subsystems, usually controlled by independent legacy monitoring systems that lack interoperability and need to be integrated.

So far, a number of solution for providing a decision support and collaboration in emergency situations were proposed in literature. For instance, an expert system based on Petri Nets for emergency response management, proposed in [1], operates on heterogeneous information and integrates various decision bodies in case of systems such as electricity distribution and water supply which are networked in terms of their inter-dependency. A multi-cloud computing system was proposed in [2] that supports dynamic processing of data during emergencies and continuously

adapts to the changing conditions at the scene. In [3] the authors investigated a cloud-based EM system for environmental and structural monitoring that utilizes the distributed computing capability to analyze the mass data collected by sensor networks deployed in civil environment. In [4] emergency evacuation system was proposed which takes cloud computing and big data characteristics into consideration during decision making after natural disasters. Authors of [5] introduce a low-cost mobile collaborative application for EM allowing ad hoc communication, decisions support and collaboration among firefighters in the field. In [6] authors proposed the use of advanced knowledge models to support environmental EM as an adequate response to the current needs and technology. At the same time, a number of EU projects were focused at development of EM solutions and technologies to provide an adequate support in crisis and disaster situations. By supporting the interoperability of contextual information [7] and resource allocation [8] to create an adequate command and control structure for the purpose of crisis management, EU project IDIRA was focused on the large-scale disasters. On the other hand, EU project ESS provided a suite of real-time data-centric technologies [9], [10] in form of a data fusion mediation system for information processing and fusion which can be ubiquitously accessible by the stakeholders during the emergencies. Moreover, EU project BRIDGE aimed to facilitate collaboration of stakeholders and enable integration of resources and technologies into EM workflow [11] by making the decisions based on the risk assessment [12] and processing the social media to detect relevant sub-events [13]. All these existing solutions are to the significant extent tailored to specificities of CI of interest, often limited in reasoning capabilities, and do not provide a holistic management capable of processing a multitude of heterogeneous events. Moreover, many of them do not support training of EM personnel as an important aspect of improving the overall responsiveness and resilience of the targeted infrastructure.

In this paper, one way to provide a Recommendation and Decision Support System (RDSS) and an adequate EM support to the CI operator was proposed as a result of European FP7 project EMILI which investigated management of emergencies in large infrastructures. The proposed RDSS support was leveraged on an intelligent, event-driven layer implemented upon the existing CI monitoring systems. Unlike the contemporary EM systems, this intelligent layer is capable of acquiring, filtering and processing a multitude of events (so called alarm avalanches) that are generated by different sources, such as field-level devices, technical systems and manually by involved stakeholders. Its Complex Event Processing (CEP) capabilities are supported by extensive EM rule base (made of rules and heuristics extracted from emergency manuals and acquired by human decision makers), but also by facility data model implemented in a form of CI ontology. Having such CEP engine as the core of the system, its reasoning capabilities were significantly improved comparing to the existing

solutions. Based on the triggered events, the proposed system conducts the situation and risk assessment, and reacts accordingly, either automatically or via recommendations proposed to EM personnel thus supporting their overall decision making process. Furthermore, the objective of this paper was to propose the conceptual basis for future event-driven RDSS in CIs. More precisely, the proposed EM system was envisaged as a mixture of event driven and service oriented architecture where CEP engine (and its rule base) plays the main role and executes all business logic in the system. In this way, the proposed system provides a holistic, integrated and intelligent treatment of events, in a flexible, extendable and adaptable manner, applicable to practically any type of CI. As a baseline for implementation of EM procedures and definition of the proposed approach, relevant contemporary emergency standards and guidelines (such as proposed by International Civil Aviation Organization (ICAO) in case of airports [14]) were taken into account. In addition, to support integration and interoperability of overall system, main principles of IEC 61970-451 standard (CIS Information Exchange Model for SCADA) [15] were taken for definition of communication protocol among system components (software modules and existing monitoring systems).

Every CI is obliged to have a corresponding EM plan, depending on the applied (inter)national regulations, by which a set of operative actions are prescribed that should be followed during the emergency and disaster situations in order to protect people and property. The EM personnel usually have to undergo extensive training to be familiar with the actual emergency procedures both general and specific to the facility of interest. Therefore, in order to improve the emergency preparedness and overall effectiveness of CI personnel, it is crucial to apply realistic scenarios in practice, i.e. for training purposes. Unlike predominant, rather inflexible training practice, based on static pre-planning of emergency scenarios, the EM solution proposed in this paper allows for interactive instructor-trainee sessions, with dynamically changing training scenarios. Moreover, the proposed approach facilitates qualification and certification of emergency personnel, as well as e-learning and self-assessment of trainees. The training module relies upon the same rule base as the real-time RDSS meant to help emergency personnel to take right and timely decisions in case of real emergencies. Finally, in order to verify the proposed EM system and its holistic applicability, both event-driven RDSS support and training approach were deployed and demonstrated in a real CI. For this purpose, the airport use case was chosen, more precisely, Nikola Tesla airport (NTA) in Belgrade, as one of the most complex and demanding CIs. The proposed system was tested and evaluated on a case study of fire incident at the airport premises.

The remainder of this paper is organized as follows. Section 2 analyses the contemporary research challenges in EM domain. Section 3 describes the proposed EM solution by introducing the proposed system architecture and analysing relevant system components. Additionally, this section describes

in detail the proposed event-driven RDSS support and training approach envisioned for improving the preparedness of EM personnel in real emergency situations. Airport use case is described in Section 4, while Section 5 elaborates on particularities of system implementation. Evaluation of system performance during a fire incident at the airport premises is investigated in Section 6. Finally, Section 7 provides the concluding remarks of this paper.

2 Problem statement

Situation awareness is commonly recognised 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 [16],[17]. Emergencies on the other hand, are often large scale, distributed events that require many prompt decisions which need to be based on, often, incomplete, inconsistent or even contradictory information originating from a variety of sources.

Emergency management, disaster management and crisis management are often used interchangeably to refer to the management of such situations and it generally consists of four stages: 1) mitigation, 2) preparedness, 3) response and 4) recovery. This process is supported by different active and passive stakeholders and relies on a variety of ICT systems with focus on different aspects and phases of the process, such as integration of sensors, data gathering, data fusion and interpretation, communication, collaboration, visualization and decision support [18].

2.1 Reasoning capabilities over multiple data sources

CI consist of a number of technical subsystems, each of which can generate multitude of events and execute myriad of actions. In case of an emergency, these systems are likely to produce an avalanche of alarms, which can be overwhelming for the operator. Therefore, the operator needs to be able to disregard alarms that are less relevant, combine all available information and based on that make decisions. This means that in order to support the operator, the system needs to react to incoming events [19], while all sub-systems need to be treated in a holistic, integrated and intelligent way, resulting in a clear picture of the situation [20]. Existing SCADA systems provide functionalities that are necessary to monitor and control each device in the system, however they lack reasoning capabilities that would take into account all available information, i.e. incoming event streams as well as static information such as regarding CI's topology [7],[11]. The system described in this paper represents integrative, intelligent layer on top of the existing monitoring system that ensures holistic management of all sub-systems and collaboration of stakeholders by combining incoming data from heterogeneous sources with the available knowledge about the target CI.

2.2 Flexibility, adaptability and responsiveness

Having in mind that CIs are not static, they evolve over time. For example, sensors and subsystems may be replaced or modified, while rooms and areas might get different purpose over time. Therefore, the EM systems should be flexible, extendable, adaptable and responsive to new challenges and modifications, meaning that business logic should be easy to modify without affecting other components in the system. In existing EM solutions, logic is usually hardwired in core of the system or even at the PLC level, thus making it hard to adapt the system's behaviour taking into account the implementation of necessary modifications and redeployment of entire system [2],[6]. On the other hand, the EM approach proposed in this paper relies on the reasoning engine that executes logic based on supplied rule base. The benefit of this approach is twofold. Rule base can be modified without tampering the legacy monitoring system itself, meaning that the aforementioned situation can be solved simply by replacing the rule base, which is sometimes possible even on a live system. Rules are completely independent of the legacy systems, making it possible for CI's technical personnel to modify the rule base themselves, and thus change system's behaviour. Knowledge model of the infrastructure is formulated in an ontology that can be queried by the engine and modified to reflect any changes in the infrastructure itself.

2.3 Collaboration between stakeholders

Additionally, crucial to the success of EM is communication between stakeholders, i.e. communication on scene, from on-scene to off-scene and vice versa in order to reach a common understanding (i.e. consensus building) between all stakeholders, especially among responders [21]-[23]. Combined with situational awareness, this common understanding, often referred to as "shared situational awareness", plays a key role and helps to handle the emergency more effectively. It is achieved by tackling the following challenges [24]:

- provision of filtered information to the user, so that they are not overloaded with information;
- generation of knowledge from the collected data and information;
- design of a suitable collaborative infrastructure;
- generation of a common understanding of the situation.

However, often first responders, coming from different organizations (such as emergency medical services, firefighting services, police and local/regional/state authorities), have different organizational structures, communication and coordination protocols. This in combination with poor information sharing and coordination during disaster response, introduces obstacles in collaboration and hampers the decision-making process [25]. However, approach proposed in this paper focuses on prevention and early response phases, thus putting an emphasis on collaboration and coordination of

response activities. In this way, relevant information is delivered to involved stakeholders, providing adequate recommendations and effective collaboration as prescribed in the emergency manuals.

2.4 Disaster preparedness

Incidents are extremely rare and consequently, when an emergency actually occurs, operators find themselves in critical situations and unaware of all established procedures, rules of communication, response strategies, etc. To alleviate this problem, the system needs to enable training that would put operators in realistic emergency situations. Current solutions usually allow operators to get acquainted with the system and its features [26],[27]. In this setting, the user can only explore all functionalities of the system, while it is also possible to simulate various events, such as temperature reads, smoke detections, etc. However, in order to enable adequate training, realistic scenarios are needed. In this paper, the proposed system enables experts to define any scenario and provides means for simulating emergencies based on those scenarios. This way, operators can prepare for emergencies and get a chance to familiarize themselves with procedures, automatic reactions and alternative strategies in realistic settings. Training will also spur valuable feedback from the operator which will be useful for future improvements of the system.

3 Proposed emergency management approach

This section elaborates in detail the proposed approach undertaken for provision and implementation of holistic EM system [28]. The proposed solution integrates advanced concepts and technologies in operational and knowledge management, CEP, reactivity, facility data modelling, and legacy control systems, thus setting a baseline for a new generation of intelligent EM systems.

3.1 System Architecture

The proposed EM system follows a typical three-tier architecture as shown in Figure 1. The lowest tier encompasses Tangible layer, Data layer and Document Management System (DMS). Tangible layer is responsible for acquiring the low-level data from underlying control and monitoring systems (such as SCADA). To provide the possibility for training of operators in different emergency scenarios, an emulator was envisioned for generation of artificial low-level data/events. Real-world data, such as data stored in SCADA archives, are also part of this layer and can be used for simulation and replay of past events for the purpose of training. Data layer is comprised of CI models which represent the target infrastructure captured in an ontology. In addition to domain structuring and formalization, CI models also serve as a basis for ontology-based visualizations of situational assessment, facilitating annotation of “active objects”, i.e. visualized sensors and actuators that user

can interact with. Apart from CI models, Data layer includes event/activity logs and predefined scenarios of low-level events (as specified by the trainer) for performing the training procedures. Moreover, the lowest tier encompasses DMS responsible for EM documentation that is relevant for decision making and for justifying recommendations to the operator, such as EM procedures, legislations, best practice examples, etc.

The middle tier represents the Processing layer of the system which consists of the most important components for situation assessment and emergency management. CEP, which is a core system component, is asynchronously fed by data from multiple streams mostly delivered by control and monitoring systems or generated by independent technical systems such as fire protection, access control and intrusion detection, HVAC control, Flight Information Display System (FIDS), etc. It operates on top of the rule base derived in large part from the EM documentation (best-practice examples, legislations, emergency procedures) and is used to filter, correlate and aggregate incoming events, thereby enabling identification of the current situation. This in turn leads to automatic reactions and recommendations delivered by the system for the operator. More precisely, the CEP engine serves as basis for recommendation and decision support component responsible for delivering adequate support to the operator in case of emergency. Facilitated by simulators (such as fire propagation simulator described in Section 5.5), it delivers main inputs to the component for automated response designated for triggering of actions which could be performed either in fully automated manner or with the approval from the operator.

Finally, the third tier of the system is represented by the Presentation layer which consists of sophisticated User Interface (UI) components that display all relevant information and enable the operator to interact with the system. CI visualization and monitoring components show the current state of the target infrastructure using 2D plans, 3D models, etc. Presentation layer also provides the Scenario editor and Training session player which are the components aimed at experienced experts that allow definition and execution of training scenarios, respectively. Finally, also as part of this tier, component for recommendation and decision support visualization displays the situation assessment and recommendations resulting from the processing of the CEP engine and allows the operator to quickly react to the situation at hand by choosing one of the proposed response strategies.

3.2 Emergency data flow

The overall flow of events through the proposed EM system is indicated in Figure 2. As it may be noticed, the raw (atomic) events are a product of changes in the real world identified through sensors or humans (in this context people can be seen as sensors as well). Therefore, raw events are collected and delivered by the designated SCADA system for further processing. These events are reasoned

upon by processing agents, resulting in derived complex events (that represent situation overview and suggested strategies) and automatic actions [29]. Derived complex events are then consumed or visualized by the UI, while on the other hand, actions are consumed by both the UI and the control system or other technical system interacting with the real world. Namely, the UI notifies the operator that an action has been automatically initiated, and subsequently the SCADA system acts on this information and executes the desired action. Raw events, such as sensor readings, are displayed on 2D and 3D views, while derived events encompassing the digested view and recommendations are presented by the RDSS. In case that the operator selects one of the proposed strategies, an action is generated and delivered to the interested components. Similar to the case of automatic reactions, newly generated action is displayed in the UI and acted upon by the designated control system.

In other words, the UI notifies the operator about all relevant events including automatically initiated actions, while the designated control system, such as SCADA system, executes them. Finally, success or failure to execute the initiated actions is detected by the sensors, resulting in action status events which are along with other events delivered for further processing and reasoning, thus concluding the circle of information flow (as shown in Figure 2). It should also be noted that the time itself can be seen as another source of events. Namely, absence of a particular event for a certain duration can be significant in many situations. For example, if the actuator fails to execute an action in a prescribed time frame it would be reasonable to conclude that it is a matter of malfunction.

As it can be concluded, the proposed EM system consists of a multitude of diverse components. Considering that CIs potentially consist of a number of legacy systems, and due to the nature of emergencies, it is important that the proposed EM system is capable of handling multi-source data over different protocols and formats, while at the same time being able to cope with multi-rate parallel event streams and their potential bursts that can be produced by the environment. Therefore, to support seamless emergency data flow (as shown in Figure 2), the integration middleware of the proposed system was leveraged upon the transportation layer with the aim to connect different components and legacy systems and to support different data formats and transportation protocols.

More precisely, to facilitate the communication over the transportation layer with diverse devices and systems (coming from different vendors using proprietary data models and protocols), a unified messaging format was defined providing a harmonisation of the acquired low-level data. In other words, in order to deliver harmonized data to the EM system for further processing, the data acquired from the underlying monitoring systems were first transferred and then fed into the EM system using a unified messaging format (as elaborated in more detail in Section 5.3).

3.3 Event-driven decision support

Emergency management is event-driven by its nature. Emergency itself is an event, and consists of many sub-events, which require appropriate response in order to remedy the situation at hand and prevent further escalation. Consequently, the RDSS of the proposed EM approach is event-driven as well. By inspecting the emergency procedures of CIs, it was concluded that emergency activities described by corresponding procedures can be translated into a rule base which could be utilized to improve the decision support capabilities of the EM system in time-critical situations. More precisely, this was reflected in delivering adequate recommendations to the operator, automatically derived by the system in accordance to the EM procedures, in terms of which actions should be initiated next (such as to send the area warden to confirm the incident, activate the fire suppression system in specific rooms, initiate the evacuation procedure, etc.), thus facilitating the decision making in time-critical moments. In other words, actions which should be conducted to deal with different kinds of emergency situations can be formalized using rules that describe different event patterns representing those situations, and possible reactions when certain pattern is detected. Additionally, by using rules, EM procedures which are usually stipulated by (inter)national regulations can be translated into explicit form and verified in different scenarios.

At the hearth of this approach is the CEP engine [30], [31]. The engine works on top of the supplied rule base which is evaluated against the incoming streams of events. Internally, the engine can be seen as a network of processing agents whose main role is to detect patterns in the incoming streams. In the proposed EM system, the processing consists of the following stages. In the first step, input streams are cleaned by filtering out irrelevant events, which is often based on the type of the incoming event and the current state of the CI (e.g. during emergencies, minor alarms can be ignored, while events that are relevant to the emergency at hand can be given higher priority). Therefore, in addition to detecting event patterns, the CEP engine also manages the state, such as the alert level of the CI. After filtering, retained events are enriched with static knowledge contained in the CI data models (represented in a form of ontology that models the CI), thus providing information such as type and locations of sensors, adjacency of rooms, etc. Namely, events coming from the field-level carry a reference to the device that generated it, which is then used to lookup additional information in the CI data model. Now, with the additional knowledge included and unnecessary inputs discarded, these intermediary events are split, combined and aggregated, leading to detection of relevant event patterns that give digested view of the current situation, or result in requests to execute necessary actions, or alternative strategies proposed to the operator. In this way, by pointing out the relevant events from the multitude of obtained alarms and data (e.g. by indicating

the type of incident, which assets and areas are under threat), and by delivering recommendations in fully automatic manner on which critical actions or strategies should be followed (e.g. in terms of technical system control, collaboration among involved stakeholders, etc.), adequate decision support capabilities were provided by the system. Furthermore, based on the current state and detected event patterns, the processing agents can also initiate state changes which should be often approved by the operator (e.g. suggesting that the alertness level should be increased in specific areas or entire CI). Finally, in addition to events coming from the field-level, the engine also reasons on inputs provided by the simulators which estimate evolution of the incident over time and thereby predict future safety conditions based on current sensor readings. This knowledge is particularly useful for planning the evacuation in order to avoid directing people to areas that are likely to become unsafe in the immediate future. In this way, system is capable of delivering more reliable support in the decision making process of the operator, by having an insight into the forecasted evolution of events.

3.4 Training approach

The purpose of the training is to improve the effectiveness and emergency preparedness of EM personnel. The proposed training approach (shown in Figure 3) allows for an interactive instructor-trainee sessions, whereby two separate EM training software instances are meant for trainer and trainee (implemented under so called Simulation and Training Environment (SITE) [32]). During training, the real world is replaced by the emulator which produces raw events that are otherwise generated by sensors, and processes all commands that are in reality executed by the supervising SCADA system or technical systems directly. The emulator is controlled by the trainer who also supervises the training session through his software instance. In addition to having access to all components as the trainee /operator, the trainer can also influence the future evolution of the scenario by introducing additional problems or slowing and stopping the escalation of the situation at hand.

Before it is started, the emulator needs to be configured with a scenario that is created with the Scenario editor. This is achieved in three stages: preparation of event patterns, combining patterns to create scenarios, and definition of request/response patterns. Event patterns are reusable building blocks for creating scenarios and they contain any number of raw events such as sensor readings along with a time stamp relative to the start of the execution of the pattern. Their purpose is to represent sub-scenarios that can be reused across a number of scenarios (e.g. fire in a specific room, problems with the sprinkler system, etc.). The existing patterns are then combined in order to produce the scenario, while the request/response patterns are used mainly to represent problems such as faulty actuators or external conditions rendering execution of a command impossible. Namely, whenever a command aimed at supervising SCADA system or technical systems is issued, the

emulator successfully executes it by default. This behaviour can be overridden in the command/response pattern by manually defining system's behaviour in response to any command that is available in the system.

Furthermore, having in mind that it would be impossible to realistically emulate various specificities of real world such as behaviour of supporting personnel, the trainer is able to participate in the scenario as well, and for this purpose, an additional set of rules is used. The role of these rules is mainly to support the trainer by assisting him in impersonating supporting personnel, and reacting to operator's decisions. For example, when a report is expected from one of the supporting personnel, the trainer is presented with a list of possible responses to choose from. Similarly, these rules can monitor the execution of the scenario along with trainee's choices, and determine whether the trainee made the right calls. Based on this, the system can suggest the trainer to react to bad calls by introducing additional problems in the scenario, or to reward good choices by decreasing the severity of the situation at hand. Alternatively, simulators could be used to determine outcomes of trainee's actions. However, in many cases state of the art simulators are either: (1) not precise enough, (2) applicable only in special circumstances, or (3) take too long to execute. This makes trainer's involvement and ability to influence evolution of scenarios even more significant as it provides a way to compensate for the shortcoming of the simulators.

During training, trainee is subjected to conditions that are identical to the real life, meaning that he uses the same software as he would in the real life and that events produced by the emulator are treated the same way as if they were generated by the sensors. Furthermore, trainer is not a mere observer, and can influence the execution of the scenario in different ways, either through impersonating other stakeholders or by executing previously prepared arbitrary groups of actions, thereby steering the scenario in order to make the training session as realistic as possible. Namely, trainer receives all information about the current situation and gets an insight into what trainee sees and does. Based on that, he can influence evolution of the scenario either through reports of other stakeholders, execution of prepared groups of actions, and opting for alternatives suggested by the system based on the training rule base. With this approach, training is dynamic, while trainer's direct involvement ensures that the training is more realistic, and more challenging. This also makes training more interactive, thus improving the learning process.

4 Airport use case

To demonstrate the applicability of the proposed EM solution, both event-driven RDSS support and training approach were deployed and verified in a real CI. For this purpose, the airport was chosen as one of the most complex and demanding CIs, representing the critical hub for air transportation of

people and goods. Emergency situations that can occur at the airports are divided into aircraft emergencies and terminal and ground emergencies of which the second one is in the focus of this paper. Terminal and ground emergencies could include various scenarios such as structural fire incidents, bomb threat and explosions, hazardous materials contamination, and other kinds of technical system malfunctions (such as evacuation door stuck, emergency lighting power failure, etc.) that can significantly influence the response capability. Having in mind that fire is one of the most dangerous and demanding threats to handle, a structural fire has been chosen as a concrete emergency scenario for demonstration purposes. Main goal of this case study was the safety of people, while the secondary objective was to protect critical assets (such as power supply boards, data centre, etc.) and restore of normal operating condition.

Airports are commonly managed through a variety of technical systems, many of which are relevant for detecting and managing incidents (such as fire protection, smoke extraction, water and power supply, evacuation and access control, etc.). Furthermore, in order to provide an adequate management and support to the airport operator in case of emergency, these technical systems need to work together in an integrated manner and therefore have to be coordinated. To demonstrate the proposed system and its integration and interoperability capabilities in real airport premises, Nikola Tesla airport (NTA) [33] serving Belgrade, Serbia, was chosen due to its topological complexity and diversity of technical systems with numerous sensors and actuators operating at the site.

4.1 Case study – fire scenario

To demonstrate the applicability of the proposed EM solution, a fire scenario was defined which was inspired by an actual incident happened at NTA in 2009. The incident started in a sorting room with luggage carousels located at the ground floor of NTA Terminal 2 (Figure 4). More precisely, fire and smoke detectors were triggered by fire and large amount of exhaust gases caused by explosion in malfunctioning engine of the vehicle exiting the sorting room towards the airport apron. Basically, entire fire scenario utilized for demonstration purposes was emulated based upon the real events recorded by the main BMS of the airport (SCADA View6000 [34]) during the course of real incident. Furthermore, the demonstration scenario was additionally complicated with spread of fire toward the neighbouring premises and malfunction of vital technical systems (such as assuming that certain evacuation doors were stuck during the evacuation procedure) to validate and analyse the decisions made by the system. Actually, since sorting room was surrounded by fire resisting walls (resistant to fire up to 120 minutes), but with a single layered non-fire resistant floor, it was assumed that fire started to spread toward the basement of the airport building. The overall goal was to

evacuate the jeopardized people with regards to the safety of areas on the evacuation routes, start the fire suppression and smoke extraction and consequently to make threatened areas safe again.

In case of a fire, fire protection system is essential for suppressing the fire in affected areas, while smoke extraction is used to remove the smoke which is often the cause of death in such incidents. At the same time, HVAC is used to adjust the air flow in order to cut the fresh air supply to areas under fire, while evacuation and access control, emergency lighting, and public address systems are used to assist people in the self-rescue phase.

In addition to supervising and coordinating technical systems, the operator in the control room, who is the main actor in resolving EM situations, also has to coordinate the actions with other involved stakeholders. Among others, this includes EM Coordinator who is appointed to approve large-scale actions (for instance, evacuation has to be approved first by airport director), area wardens, medical staff and airport personnel that assist either in the detection and confirmation of the incident, or activities related to ensuring the safety of people and protection of property.

4.2 Fire scenario workflow

An emergency situations at airports are generally seen to have three different stages (according to ICAO recommended practices [14]), whereby each stage corresponds to designated level of alertness (for instance, denoted as ALERT 1, ALERT 2 and ALERT 3). In case of an incident (such as fire in this particular case), corresponding scenario workflow is presented in Figure 5 encompassing these three stages of alertness. As it may be seen from Figure 5, ALERT 1 is the lowest level of alertness, and it denotes that fire was detected somewhere at the airport. As immediate reaction to the raise of ALERT 1, an area warden is sent to the jeopardized area to verify whether the fire actually happened and to send back a report. As a results, if it was a matter of false alarm, the level of alertness would be reduced, while if fire was confirmed by an area warden, corresponding mitigation and response actions would be started. According to the real fire incident experienced at the airport, this stage lasted for about 1 minute, after which fire started rapidly to spread.

In case of fire confirmation, the second level of alertness is raised, i.e. ALERT 2, during which the size of fire and fire mitigation successfulness are estimated. ALERT 2 also considers an activation of Emergency Operations Centre (EOC) which should serve as central command and control facility responsible for further EM actions that should be carried out at the airport. In case it is estimated that the severity of fire would turn from moderate to major, ALERT 3, i.e. the highest level of alertness is raised which suggests for large-scale mitigation actions and evacuation of people from the airport. For the purpose of training, identical fire scenario as experienced in reality at the airport was defined by an experienced emergency manager utilizing the Scenario editor. Based on such scenario,

corresponding atomic events were generated via Training session player to evaluate trainee's preparedness against the described scenario workflow.

During emergencies, many sub-processes are taking place, and in the case of fire, the most important and challenging are the following [35]: 1) *fire detection* (performed via fire detectors, personnel reports or emergency calls; detection of fire is confirmed automatically in case of multiple fire alarms), 2) *situation assessment and categorization* (incident severity is estimated based on the ratio of triggered and total number of fire detectors in a jeopardized area), 3) *emergency prevention - fire mitigation* (some of the mitigation actions could be initiated automatically, such as shutting down the ventilation system, activation of water curtains and evacuation path lighting), 4) *risk assessment and evaluation* (performed based on the relevant parameters such as incident nature and severity, number of people that are jeopardized, and criticality of technical assets under threat), and 5) *evacuation management* (allocation of the safest evacuation routes [36] depending on the location of exits, safety state along the evacuation routes and the number of people to be evacuated). More detailed description and investigation on possible EM workflows at the airport is provided in [35].

5 System implementation

Prior to deployment of the proposed EM system in previously described airport use case, thorough analysis of currently available software tools and technologies was carried out. Technologies chosen for implementation of relevant system components are presented in the following.

5.1 CEP engine

CEP engine is needed to evaluate a set of rules designed to detect emergency situations and complex event patterns that precede them. During execution, the system acquires raw events and injects them into the engine, and at the same time archives any significant events (both raw and complex events derived by the engine) in order to provide means for post mortem analysis and recreation of past scenarios for the purpose of training or rule evaluation. On the other hand, the CEP engine processes incoming events based on the specified rule base, which also includes handling of the event storage based on the time windows and event orderings defined in the rules (as specified in condition of the Rule 3 in Table 1). Having in mind that engines vary in terms of rule expressiveness, performance, and other features such as high availability, a range of CEP tools was investigated in [37] with regards to emergency management and mission-critical applications. Evaluation was based on the core features which were identified as most relevant in the considered domain: comfortable and intuitive specification of events, integration of event streams and static data, possibility to modify event specification, ability to process multiple streams, recovery, logging, time model, consuming

approach, temporal relations, accumulation, aggregation and negation, possibility to execute actions, performance and high availability. Taking into consideration the maturity, stability and sustainability of the solutions, Esper [38] was selected as the most suitable for implementation of the proposed solution [37]. It lacks some of the concepts that are relevant in the EM domain, but the same effects can be achieved indirectly using other mechanisms. For instance, Esper does not have a notion of an action. However, actions can be broken down into series of events. Action is started by generating an “action request” event and sending it to the SCADA, while success of the execution can be determined by waiting for the appropriate sensor reading event to be generated in the prescribed time frame.

Moreover, one of the advantages of using Esper is that external information can be held outside the engine and acquired via static method calls. This is required for instance when incoming raw events need to be enriched with “static” knowledge, e.g. location of the sensor. Alternatively, such knowledge can be injected into the so-called named windows. Contrary to other types of windows, the named windows represent streams where events can be added and removed manually, making them ideal for storing information that is rarely changed and needs to be queried by the engine. Compared to using static method calls, this approach is more reactive. If something in the model needs to be changed, it can be fed to the engine as an event upon which the named window, as well as all queries can be updated immediately. On the other hand, in the static method approach it is needed to periodically poll all external variables and generate an event if a change is detected.

5.2 Rule base

Serving as a backbone of the CEP engine, rule base was defined to facilitate decision support in time critical situations. Rules for the airport use case were derived primarily from the emergency related procedures that are used at NTA and partially from the ICAO guidelines [14]. The rule base consists of around 50 Event Condition Action (ECA) rules which were specified to improve the situation awareness and to facilitate decision support capabilities of the system. Different types of incidents were taken into account such as fire incidents, terrorist attacks (e.g. bomb threat/explosion) and incidents with hazardous materials. As mentioned in [35], rules were defined by translating the EM procedures, while respecting the low-level signals and airport operating modes. Some of the relevant parameters that are included in definition of rules are the following: incident severity (determined based on the number of triggered fire detectors, emergency calls and personnel reports), fire/smoke propagation, safety of areas (estimated based on the data coming from sensors and facility layout data such as information regarding the adjacent rooms), incident location, operating condition of

relevant equipment, and safety of evacuation routes (e.g. depending on the operational status of emergency exits/doors).

Some of the rules contained within the rule base are given in Table 1 using DURA based pseudo syntax [39], [40] and ECA formalism. For example, Rule 1 monitors operating alarms (such as fire detection, access control malfunction, etc.), and in case evacuation is initiated or is already in progress, it alerts the operator about possible issues that are likely to arise. On the other hand, as defined in Rule 2, fire detection events need to be treated differently depending on external state such as operating mode/regime of the airport facility due to different availability of security personnel (e.g. during nights when incidents should be treated with more caution and alarm is raised immediately). Finally, Rule 3 assigns corresponding alert level based on the severity of detected incident. More precisely, in case incident was detected, warden is immediately notified to check the area in question after which a confirmation of the assignment and report are expected. If everything goes as intended and the report is submitted, alert level is assigned based on the incident severity defined in the warden's report. Based on this rule set, the CEP engine, which is the core of the system, correlates the atomic events into complex events providing a high-level meaning, conducts the situation and risk assessment, and supports system reactions, either automatically or via recommendations proposed to EM personnel.

In practice, the rules first need to be formulated by designated safety and security experts in natural language (or possibly in pseudo syntax of their choosing), and then automatically or semi-automatically translated into target CEP engine's syntax by the developers maintaining the system. Afterwards, the effectiveness of the rules can be evaluated by replaying past scenarios from the event archive.

5.3 Integration middleware

For the integration middleware, a number of enterprise service buses (ESBs) were evaluated, such as JBoss ESB, Apache Service Mix, OpenESB, Mule ESB, WSO2 ESB. By comparing the performance of mentioned alternatives, WSO2 ESB [41] imposed itself as the strongest candidate, which was further implemented to serve as the transportation layer of the proposed system [42]. WSO2 ESB takes a lightweight approach and relies on Apache Synapse Web service mediation and routing engine which focuses on providing fast XML message processing. WSO2 ESB was chosen since its primary purpose is to connect different software systems by supporting a plethora of data formats and transportation protocols. It consist of a number of endpoints, each of which represents one of the systems that are to be integrated, and communicates with them using formats and protocols that the system in question supports. Internally, messages were routed from the source endpoint, through a

network of nodes based on content or prescribed paths, and delivered to the target endpoint representing target system. Another benefit of this approach was that WSO2 ESB usually enable content-based routing, as well as basic filtering and transformation, thus removing some of the load from the processing layer and the CEP.

In order to support various types of communication and information exchange, such as sending alarms/events, action commands, polling the sensor readings etc., between the SCADA systems and the EM system components, as well as between system components themselves, a common communication protocol, i.e. a message format had to be defined (proposed in [43]). This format was based on the IEC 61970-451 standard (CIS Information Exchange Model for SCADA) [15], which is an effort to standardize SCADA communication that is invariant of the application domain. As such, the proposed format is XML-based, generic and extendable. It utilizes XML schemas constructed according to the main principles of the IEC 61970-451 standard, and enables the exchange of three types of messages: (1) alarm-event messages utilized for representation of atomic and complex events, (2) action-request/response messages utilized for representation of action commands and corresponding system responses, and (3) data-request/response messages utilized for polling the sensor readings.

5.4 Ontology based Facility Model

For the sake of the airport use case implementation, a comprehensive facility data model in form of an airport ontology [44] was developed. The aim of ontology was to classify and define related entities of the airport infrastructure (i.e. Terminal 2 of NTA), but also to reason upon the entities defined within that domain. It models and connects all static knowledge referring to the airport building topology, necessary for the decision support in critical situations. In the same time, it enabled the integration and interoperability of different technical systems at the airport (such as HVAC, access control system, fire protection system, public address system, etc.).

The objective of the airport ontology was to facilitate the interpretation and semantic enrichment of SCADA signals using the underlying spatial and topological model of the airport, including vendor data, the equipment characteristics, protocols and standards used, etc. Static knowledge captured in airport ontology contains more than 1300 instances, including 500 sensors devices, 400 actuators, and 100 functional components. It also includes semantic relations denoting crucial information such as location of sensors, adjacency of rooms, or which technical system a device is a part of. The ontology was accessed and queried using Jena, a free and open source Java framework for building Semantic Web and Linked Data applications.

5.5 Fire propagation simulator

Among other components, the proposed EM system also includes the simulator of the fire propagation. Implemented fire propagation simulator was initially proposed in [45]. It is leveraged upon physical models of the target infrastructure and various input variables/parameters such as:

- number of persons under threat,
- person flux through hallways/doors,
- concentration of smoke,
- room temperature, and
- smoke flow rate through hallways/doors.

The simulator estimates propagation of fire and smoke throughout underlying infrastructure physical model of NTA, thereby forecasting the evolution of safety conditions in time. Simulations are triggered when evacuation is required and when operating conditions change in a way that affects the evacuation process, thereby requiring changes in the evacuation strategies. When simulations are done, corresponding event is generated in order to notify the system that the new evacuation strategy is ready.

5.6 Advanced visualization

Complex events that represent the output of CEP processing are then visualized in the main UI of RDSS (as proposed in [35]). In that way, the operator was provided with the situation overview and assessment, as well as with the corresponding recommendations delivered by the system (as shown in Figure 6). As it may be seen from Figure 6, the left part of the main UI screen serves to provide an overview of the ongoing situation (including detected events and alarms). More precisely, in this section of the main UI screen, the state and incident severity are shown for affected fire sector and belonging areas under threat. Purpose of the right part of the main UI screen was to show the recommendations generated as a result of reasoning performed upon the acquired atomic events. Recommendations were indicated by their title (representing a type of detected incident), short description (providing more details on incident) and list of suggested actions (which operator could choose to be initiated). In addition, a justification of each recommendation given by the system (i.e. the provenance of the recommendation) can be provided upon request.

Furthermore, apart from the main UI, RDSS includes also corresponding visualization modules (2D and 3D), Action control panel, as well as Document management panel. Their role is to give detailed information about the infrastructure and enable execution of all available commands. Safety of the areas, including state of the fire detectors, suggested evacuation routes, malfunction of emergency doors, escalators, elevators, are presented in both 2D and 3D visualization (Figure 7 and Figure 8

show NTA ground floor with locations of fire/smoke detectors indicated as green/red dots). Purpose of the Action control panel was to provide the possibility to the operator to define and initiate the execution of multiple actions (defined as a list of desired commands). Furthermore, it could be used to monitor the status of the initiated actions which could be in execution, completed or failed to be executed. Document management panel was utilized to provide additional information to the operator or trainee regarding the suggested procedures and actions, stakeholders which should be involved, etc.

RDSS UI was implemented using SWT (Standard Widget Toolkit), a Java GUI library, while the visualization of the 3D model of the airport building was implemented using jMonkeyEngine, a Java 3D engine. The model was built using actual architectural plans of NTA and stored in OBJ format, which is the most universally accepted format for representing 3D geometries. It contains layers representing each of the three floors of NTA building and specific objects such as fire/smoke detectors, thus enabling selection of elements which will be visualized. Finally, supporting documentation (such as EM procedures and manuals) was stored in the Alfresco document management system. Documents were accessed through the WebDAV protocol and shown to the operator via SWT's browser component.

6 System performance

Fire scenario used for system performance evaluation, as mentioned in Section 4.1, was defined according to the actual incident which happened at NTA in 2009. Actual incident included triggering of fire/smoke detectors in a sorting room, followed with the fire suppression activities and personnel evacuation. Entire incident lasted for about half an hour, whereby the first fire/smoke detection was taken for the beginning of an incident, while its end was declared when all personnel were evacuated, fire entirely suppressed, and normal operation was restored. The evacuation activities took into account only the personnel in a sorting room, while people in the neighbouring areas were not evacuated despite their lives could have been easily jeopardized by spreading of the fire. On the other hand, fire scenario defined for demonstration purposes consisted of real events (such as fire/smoke detections) recorded by the main BMS of the airport during the incident. As mentioned, apart from the real events, additional complications in form of technical systems malfunction were introduced (such as evacuation doors stuck, etc.). Fire suppression activities were simulated and they lasted as in the case of real incident, while events of fire/smoke detections were artificially generated and fed into the proposed EM system which was running at the airport.

Comparing to the actual incident when 15 fire detectors were triggered in a sorting room, emulated fire scenario included 6 more atomic events (apart from fire detections) indicating confirmation

message/report by warden and technical device malfunctions. Throughout the scenario, EM RDSS system derived 15 complex events/actions suggesting ventilation system shut down, smoke extraction and water curtain activation, unlocking the doors to restricted areas, assigning a warden to check the alarm and starting the evacuation. Furthermore, scenario included evacuation not only of the personnel in the sorting room, but also of the people in the adjacent areas. In this case, fire scenario lasted for about 15 minutes which was twice less time needed for completion of people evacuation and restoration of normal operation as compared to the actual incident. Having in mind the scalability and the computational capability of crucial system components (such as CEP engine and transportation layer [37], [42]), the proposed EM system can easily cope with more event intensive scenarios comparing to the demonstrated one (e.g. experiencing above thousand events per second). In this regard, the proposed system components were evaluated in terms of computational power, i.e. the number of processed events per time unit. For this purpose, Intel(R) Core(TM) i7 CPU Q820@1.73GHz with 8GB RAM was utilized. CEP engine, implemented with Esper, was evaluated by sequentially injecting emulated atomic events (such as fire detection events). In this way, CEP engine managed to process 12.375 atomic events/s, whereby, at the same time, it derived 57.790 complex events/s and actions in average (such as inform the operator, are risk increased, fire confirmed) as intermediate or output steps. Regarding the WSO2 ESB communication middleware, substantial benchmarking have been already reported in [46]. Taking into account reported results and the average size of XML messages (around 700B) that ESB was dispatching, around 3.700 messages per second were handled in average (in case of single thread content based routing). In comparison with CEP engine performance, it can be concluded that ESB determines the overall system performance. However, it should be kept in mind that the overall performance could be even further improved by introducing multi-thread dispatching of messages.

Another parameter for assessing the proposed EM system was the questionnaire-based evaluation made by operators in the control room of NTA, who were using the system during the emulated fire incident in both operational and training mode. Airport operators (more precisely, 3 of them appointed in shifts for 8 hours) evaluated the system as efficient and user friendly, and the recommended actions as intuitive and easy to follow. With the proposed EM system in place, instead of seeing individual alarms, operator gets a high-level overview of the ongoing situation. System processes individual alarms from disparate systems, and notifies the operator which areas are in danger and determines the severity of the situation at hand. Furthermore, certain actions, such as notifying the warden, are executed automatically, while the operator is guided through all the steps involved in the collaboration with other stakeholders. Consequently, the operator does not need to pay attention to every single fire alarm, or to know by heart all operating procedures. For example,

depending on the current situation, the system automatically notifies other stakeholders and keeps track of their activities. If they do not act in prescribed time frame, the operator is notified and advised on how to proceed. As a result, operator was not overwhelmed with information, he had more time, and he was able to focus on making right decisions based on valuable high-level knowledge automatically delivered by the EM system, thus leading to improved management over the emergency situation.

7 Conclusion

CIs in general are difficult to manage in emergency situations due to their complexity, size, and number of underlying technical systems and involved stakeholders. Legacy supervisory and control systems are limited in reasoning capabilities and usually do not take into account all available information, coming from monitoring equipment and sensors or manually provided by human. In this paper, one way to provide an intelligent EM support to the CI operator was proposed. The proposed solution was leveraged on an intelligent, event-driven layer implemented upon the existing ICT infrastructure. This intelligent layer providing an advanced RDSS was based on an innovative CEP concept capable of acquiring, filtering and processing a multitude of events (so called alarm avalanches). Existing field-level devices forward relevant events to the proposed RDSS system where events are enriched, correlated and reasoned upon in order to derive high-level knowledge and enable holistic management of the CI. More precisely, this approach provides an overview of the ongoing situation, and based on the assessed situation system automatically reacts and suggests adequate recommendations to the CI operator. In this way, the proposed EM system reduces the quantity and increases the quality of the information that the operators deals with, thus contributing to quicker and better decisions in critical situations.

Serving as the core of the system, a set of CEP rules was defined based on the existing emergency procedures and manuals of the target CI. These rules were evaluated by the CEP engine thus generating more meaningful complex events representing an overview of the situation and recommendations that are visualized to the operator. In addition to being able to efficiently process event streams originating from the monitoring equipment and sensors, this approach also provides an easy way to specify and modify background logic of the system to accommodate practically any CI of interest. Namely, logic is expressed in CEP rules that can be added, removed and modified in real time, without affecting any of the components in the EM system. Furthermore, the proposed system incorporates also a training module which provides a way for interactive instructor-trainee sessions and definition of realistic and flexible training scenarios. During the scenario execution, simulators are used to generate events and react to trainee's actions according to the pre-defined scenarios. To

account for limitations of the simulators and lack of supporting personnel, the trainer can influence the scenario by providing needed inputs and generating new events. Finally, the proposed EM system was deployed and demonstrated at airport use case, for which NTA in Belgrade was chosen as a test-bed platform. System was evaluated on a concrete case study inspired by an actual fire incident that occurred at the NTA premises.

Future research and development activities will be aimed to complement the proposed EM system by developing smartphone applications for on-scene stakeholders in order to support communication and collaboration with the operator. Development of a graphical rule editor would support domain experts in the process of definition and modification of CEP rules without requiring an assistance of IT experts. Furthermore, scenario library for training purposes will be extended and additional scenarios will be elaborated also including outdoor critical situation (for instance, in case plane slips off, or plane crashes, etc.). By going outside the CI, open data, social media, crowdsourcing and public location-based services as well as integration with high-level systems such as 112 emergency system, will gain relevance for providing improved situation assessment, thus opening up the new possibilities for the research in domain of EM.

Acknowledgment

The research presented in this paper is financed by the European Union (FP7 EMILI project, Pr. No: 242438) and partly by the Ministry of Science and Technological Development of Republic of Serbia (SOFIA project, Pr. No: TR-32010).

References

- [1] Liu, X., Li, W., Tu, Y.L., & Zhang, W.J. (2011). An expert system for an emergency response management in Networked Safe Service Systems. *Expert Systems with Applications*, 38, 11928–11938.
- [2] Mitra, K., Saguna, & Ahlund, C. (2014). A Mobile Cloud Computing System for Emergency Management. *IEEE Cloud Computing*, 1(4), 30-38.
- [3] Qiu, M., Ming, Z., Wang, J., Yang, L.T., & Xiang, Y. (2014). Enabling Cloud Computing in Emergency Management Systems. *IEEE Cloud Computing*, 1(4), 60-67.
- [4] Moulik, S., Misra, S., & Obaidat, M.S. (2015). Smart-Evac: Big Data-Based Decision Making for Emergency Evacuation. *IEEE Cloud Computing*, 2(3), 58-65.
- [5] Monares, Á., Ochoa, S.F., Pino, J.A., Herskovic, V., Rodriguez-Covili, J., & Neyem, A. (2011). Mobile computing in urban emergency situations: Improving the support to firefighters in the field. *Expert Systems with Applications*, 38(2), 1255-1267.
- [6] Hernández, J.Z., & Serrano, J.M. (2001). Knowledge-based models for emergency management systems. *Expert Systems with Applications*, 20(2), 173-186.
- [7] Anagnostopoulos, C., & Hadjiefthymiades, S. (2013). Multivariate context collection in mobile sensor networks. *Computer Networks*, 57(6), 1394-1407.

- [8] Kostas, K., Panagidi, K., & Hadjiefthymiades, S. (2013). Optimal Spatial Partitioning for Resource Allocation. *10th International ISCRAM Conference*, Baden-Baden, Germany, 747-757.
- [9] Malewski, C., Simonis, I., Terhorst, A., & Bröring, A. (2014). StarFL - a modularised metadata language for sensor descriptions. *International Journal of Digital Earth 2014*, 7(6), 450-469.
- [10] Nüst, D., Janowicz, K., Maué, P., Bröring, A., & Malewski, C. (2011). Semantically-Enabled Sensor Plug & Play for the Sensor Web. *Molecular Diversity Preservation International Sensors*, 11(8), 7568-7605.
- [11] Büscher, M., Liegl M., & Thomas, V. (2015). Collective Intelligence in Crises. *Social Collective Intelligence: Combining the Powers of Humans and Machines* (243-265). Springer.
- [12] Braut, G.S., Rake, E.L., Aanestad, R., & Njå, O. (2012). Risk Images As Basis for Decisions Related to Provision of Public Services. *Risk Management*, 14(1), 60-76.
- [13] Pohl, D., Bouchachia, A., & Hellwagner, H. (2013). Supporting Crisis Management via Detection of Sub-Events in Social Networks. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, 5(3), 20-36.
- [14] ICAO (2004). Aerodromes - Annex 14 to the Convention on International Civil Aviation. *ICAO International Standards and Recommended Practices*. (4th ed.). <http://www.icao.int/>
- [15] IEC (2003). IEC 61970: Energy Management System Application Program Interface – Part 451: CIS Information Exchange model For SCADA, *Standard proposed by International Electrotechnical Commission*.
- [16] Endsley, R. (1995). Toward a theory of situation awareness in dynamic systems: Situation awareness. *Human factors*, 37, 32-64.
- [17] Endsley, M.R., & Connors, E.S. (2008). Situation Awareness: State of the Art. *Power and Energy Society General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, USA, 1-4.
- [18] Kraus, L., Janev, V., & Vraneš, S. (2012). Different approaches to ICT-enhanced emergency management. *56th ETRAN Conference*, Zlatibor, Serbia, VI2.7-1-4.
- [19] Mühl, G., Fiege, L., & Pietzuch, P. (2006). *Distributed Event-Based Systems*. Springer.
- [20] Mijović, V., Janev, V., & Vraneš, S. (2013). Main Challenges in Using LOD in Emergency Management Scenarios. *24th International Conference on Database and Expert Systems Applications (DEXA 2013)*, Prague, Czech Republic, 21-25.
- [21] Fan, Z., & Zlatanova, S. (2010). Exploring Ontology Potential in Emergency Management. *Gi4DM Conference - Geomatics for Disaster Management*, Torino, Italy, 109-122.
- [22] Benaben, F., Hanachi, C., Lauras, M., Couget, P., & Chapurlat, V. (2008). A Metamodel and its Ontology to Guide Crisis Characterization and its Collaborative Management. *5th International ISCRAM Conference*, Washington, DC, USA, 189-196.
- [23] Franke, J., Charoy, F., & Ulmer, C. (2011). Handling Conflicts in Autonomous Coordination of Distributed Collaborative Activities. *20th IEEE Conference on Collaboration Technologies and Infrastructures (IEEE WETICE 2011)*, Paris, France, 319-326.
- [24] Harrald, J., & Jefferson, T. (2007). Shared Situational Awareness in Emergency Management Mitigation and Response. *40th Hawaii International Conference on System Sciences (HICSS 2007)*, Waikoloa, Big Island, HI, USA, 23-30.

- [25] Bharosa, N., Lee, J., & Janssen, M. (2010). Challenges and obstacles in sharing and coordinating information during multi-agency disaster response: Propositions from field exercises. *Information Systems Frontiers*, 12 (1), 49-65.
- [26] Maciejewski R., SungYe K., King-Smith D., Ostmo K., Klosterman N., Mikkilineni A.K., Ebert D.S., Delp E.J., & Collins T.F. (2008). Situational Awareness and Visual Analytics for Emergency Response and Training. *IEEE Conference on Technologies for Homeland Security*, Waltham, MA, 252-256.
- [27] Campbell B.D., Mete H.O., Furness T., Weghorst S., & Zabinsky Z. (2008). Emergency Response Planning and Training through Interactive Simulation and Visualization with Decision Support. *IEEE Conference on Technologies for Homeland Security*, Waltham, MA, 176-180.
- [28] Vraneš, S., Stanojević, M., Janev, V., Tomašević, N., Mijović, V., & Kraus, L. (2013). EMILI Deliverable D6.7 "Documentation and on-line tutorial".
- [29] Etzion O., & Niblett P. (2011). *Event Processing in Action*, Manning Publications Co., Stamford.
- [30] Luckham, D.C. (2011). *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison-Wesley Longman Publishing, Boston.
- [31] Vranes, S., Stanojevic, M., Janev, V., Mijovic, V., Tomasevic, N., Kraus, L., & Ilic, Z. (2011). Application of complex event processing paradigm in situation awareness and management. *22nd International Workshop on Database and Expert Systems Applications (DEXA 2011)*, Toulouse, France, 289-293.
- [32] Vraneš, S., Stanojević, M., Janev, V., Tomašević, N., & Mijović, V. (2010). EMILI Deliverable D6.3 "Design of the first prototype of the integrated EMILI-SITE system".
- [33] Nikola Tesla Airport, Belgrade, Serbia, <http://www.beg.aero/>
- [34] SCADA VIEW6000, Institute Mihajlo Pupin, <http://www.pupin.rs/en/products-services/process-management/scada/>
- [35] Janev, V., Mijović, V., Tomašević, N., Kraus, L., & Vraneš, S. (2012). Dynamic Workflows For Airport Emergency Management Training. *23rd International Workshop on Database and Expert Systems Applications (DEXA 2012)*, Vienna, Austria, 5-9.
- [36] Kraus, L., Stanojević, M., Tomašević, N., & Mijović, V. (2011). A Decision Support System for Building Evacuation based on the EMILI SITE environment. *IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2011)*, Paris, France, 334-336.
- [37] Mijović, V., & Vraneš, S. (2011). A survey and evaluation of CEP tools. *17th Conference and Exhibition YU INFO*, Kopaonik, Serbia, 638-642.
- [38] ESPER, EsperTech Inc., <http://www.espertech.com/esper>
- [39] Hausmann S., Brodt S., & Bry F. (2011). Dura: Concepts and Examples. *Technical report*, University of Munich.
- [40] Hausmann S., & Bry F. (2013). Towards complex actions for complex event processing. *7th ACM international conference on Distributed event-based systems (DEBS 2013)*, Arlington, Texas, USA, 135-146.
- [41] WSO2, Middleware, <http://wso2.com/>
- [42] Ahuja S.P., & Patel A. (2011). Enterprise Service Bus: A Performance Evaluation. *Communications and Network*, 3(3), 133-140.
- [43] Tomašević N., & Konečni G. (2011). Generic message format for integration of SCADA-enabled emergency management systems. *17th Conference and Exhibition YU INFO*, Kopaonik, Serbia, 71-76.

- [44] Tomašević, N., Batić, M., Konečni, G., & Vraneš, S. (2012). Ontology-based airport data model. *2nd International Conference on Information Society Technology ICIST*, Kopaonik, Serbia, 158-163.
- [45] Bettelini, M., Rigert, S., & Seifert, N. (2013). Optimum Emergency Management Through Physical Simulation - Findings from the EMILI Research Project. *World Tunnel Congress*, Geneva, pp. 290-297.
- [46] Anfar S. (2014). WSO2 ESB Performance Round 7.5, <http://wso2.com/library/articles/2014/02/esb-performance-round-7.5/>