Computer Science and Information Systems 14(1):257–282 DOI: 10.2298/CSIS161014002L

A Distributed Architecture for Real-Time Evacuation Guidance in Large Smart Buildings

Marin Lujak[1], Holger Billhardt[2], Jürgen Dunkel[3], Alberto Fernández[2], Ramón Hermoso[4], and Sascha Ossowski[2]

[1] IMT Lille Douai, France, marin.lujak@mines-douai.fr
[2] CETINIA, University Rey Juan Carlos, Spain, {holger.billhardt, alberto.fernandez, sascha.ossowski}@urjc.es
[3] Computer Science Dept., Hochschule Hannover, Germany, juergen.dunkel@hs-hannover.de
[4] Computer Science Dept., University of Zaragoza, Spain, rhermoso@unizar.es

Abstract. In this paper, we consider the route coordination problem in emergency evacuation of large smart buildings. The building evacuation time is crucial in saving lives in emergency situations caused by imminent natural or man-made threats and disasters. Conventional approaches to evacuation route coordination are static and predefined. They rely on evacuation plans present only at a limited number of building locations and possibly a trained evacuation personnel to resolve unexpected contingencies. Smart buildings today are equipped with sensory infrastructure that can be used for an autonomous situation-aware evacuation guidance optimized in real time. A system providing such a guidance can help in avoiding additional evacuation casualties due to the flaws of the conventional evacuation approaches. Such a system should be robust and scalable to dynamically adapt to the number of evacuees and the size and safety conditions of a building. In this respect, we propose a distributed route recommender architecture for situation-aware evacuation guidance in smart buildings and describe its key modules in detail. We give an example of its functioning dynamics on a use case.

Keywords: Evacuation guidance, real-time routing, distributed evacuation coordination, complex event processing, situation aware routing.

1. Introduction

In a large building hazard, an efficient and rapid evacuation is of the utmost importance as hundreds or thousands of people need to be evacuated as quickly as possible. The objective of the building evacuation is an efficient relocation of people from a hazardous building under imminent danger to safe areas through safe and rapid evacuation routes.

Standard building evacuation approaches do not provide any means for modification of an evacuation plan as an incident unfolds. They usually rely on general recommendations and guidelines about what to consider and how to react in emergency evacuation (see, e.g., [4]). Frequently, no coordination is available except for predefined evacuation maps that contain (usually two) predetermined main evacuation routes that may get blocked as a hazard evolves. In the case of a hazard in large buildings, conventional evacuation approaches assume the introduction of trained evacuation coordinators at specific building locations, if possible. They guide evacuation based on an ad hoc decision making and incomplete situation awareness considering their locally accessible hazard information and smartphone communication with a main evacuation coordinator. Thus, the hazard real–time information dissemination and a timely update of evacuation routes is complicated causing an inefficient evacuation and in the worst case, human casualties.

While we cannot prevent imminent casualties of the hazard, we can avoid casualties due to a deficient emergency evacuation process that does not adapt timely to the hazard dynamics. To minimize evacuation–related casualties in a building emergency evacuation, smart building technologies can be used for diffusion of the evacuation information in real-time among evacuees. Recently, it was proposed that, by bringing together works from the fields of Agent-Based Social Simulation (ABSS) (see, e.g., [3]), Ambient Intelligence (AmI) (e.g., [44]), and Agreement Technologies (AT) (e.g., [41]), advanced methods and tools can be developed to address the aforementioned evacuation coordination problem [1]. AmI techniques are adequate to capture relevant features of the situation (sensors) and provide decision–makers with the means to act upon them (actuators). AT, on the other hand, are used to explore intelligent strategies for managing such advanced installations as large-scale open distributed social systems.

In this paper, we propose a distributed evacuation route guidance (ERG) system based on the integration of these technologies and a distributed crowd flow optimization. The objective is to minimize the overall evacuation egress time, i.e., the time from the start of evacuation until all people have moved out of the building through safe exits considering individual evacuation constraints. By safe exits, we mean all exits of a building that are considered to be safe based on real-time monitoring, and not only predefined emergency exits as in conventional emergency approaches.

The ERG system should seamlessly detect evacuation requests and the current hazard context. We apply complex event processing for inferring this context by exploiting real–time data. Then, it should compute efficient and safe evacuation routes in real–time based on the ongoing emergency dynamics. A dynamic, context-sensitive notion of safety is a key factor for such routes, in particular as panic-related behaviors of stampeding and herding may occur at potential bottlenecks of evacuation routes depending, among other factors, on the number of people who intend to pass through them (see, e.g., [20]).

In such settings, an adequate notion of fairness of evacuation route recommendations is important to assure the trustworthiness of the system from the evacuees' viewpoint [35]: the guidance should not only achieve good overall performance of the evacuation process, but must also generate evacuation routes for each of the evacuees that each of them perceives as efficient and fair. For example, if there are two close-by evacuees at some building location, they should be proposed the same evacuation route, and if not possible, than the routes with similar safety conditions and evacuation time. Moreover, building evacuation scenario is intrinsically distributed in the sense that sensors and evacuees are geographically dispersed and the hazard conditions usually vary throughout the building. For this reason and to increase the solution scalability, we also distribute the computation of evacuation routes throughout the building.

This paper is organized as follows. Section 2 presents related work. We assume that the ERG system operates with landmark localization and complex event processing described in Section 3. In Section 4, we formally describe the evacuation coordination problem. In Section 5, we provide the system architecture and describe the main components. Section 6 describes the functioning of the distributed computation of evacuation routes including different necessary individual and global constraints. The functioning of the architecture is shown on a case-study in Section 7. In Section 8, we conclude the paper.

2. Related work

Traditional evacuation coordination approaches overestimate the probability of the burst of mass panic [38] and minimize the information given to evacuees. This may increase fear and undermine the crowd's shared social identity developed during the common experience of an emergency [15]. Even though human behavior in crowds has been studied for some time (see, e.g., [50]), the functioning of the emerging crowd behavior in a hazard and the connection between the triggers of mass panic and the panic dynamics have been explored only recently (see, e.g., [15,38,39]).

Newer research results demonstrate that panic outreach in crowds is not that common, and, lately, it has been observed and demonstrated that evacuees demonstrate solidarity and collaboration in emergency (see, e.g., [13,15,42,47]). Cocking et al. conducted two interview-based studies of survivors' experiences of different emergencies in [9], while Drury et al. in [13] report on a survey of 1240 adults affected by the 2010 Chile earthquake. It was found that people in a mass emergency can create a common social identity, thus behaving sensibly and displaying solidarity and social support for each other. This causes people to be cooperative and altruistic towards others even if strangers in life-threatening situations and results in coordinated and beneficial collective actions [9,15]. These actions include such features as mutual support and coordination, which in turn provide a basis for collective agency and adaptive action [14].

Contrary to the traditional emergency approaches that seek to herd evacuees as if they were unintelligent and instinctive [15], the previously mentioned findings indicate that we should consider that evacuees are eager to collaborate and will accept evacuation routes that are beneficial for all if they are not individually harmful. In this light, we draw our attention to the evacuation route coordination solutions that can facilitate self-organizing behavior of evacuees by giving them updated and useful evacuation route recommendations so that they can make system-wide rational and efficient evacuation decisions.

Understanding evacuee's decision making in choosing routes is important for delivering evacuation route recommendations. Crociani et al. in [10] treat evolving pedestrian behavior and introduce a general model for decision making related to pedestrian route choices. The model encompasses three aspects influencing these choices, as observed in an experimental observation: expected travel time, perceived level of congestion on the chosen path, and decisions of another preceding pedestrian to pursue a different path.

If each evacuee is given updated information about travel times on his/her available evacuation paths, and he/she chooses the fastest path, Wardrop's first principle states that, in a traffic assignment incorporating congestion effects, no evacuee can unilaterally reduce his/her travel time by shifting to another path. The emerging crowd behavior can be modelled through user optimization which leads to deterministic user equilibrium (see, e.g., [18]). The latter can be arbitrarily more costly for congested networks than the globally optimal evacuation route assignment (see, e.g., [24,30,33]). Moreover, assuming that all the evacuees share the same objective (e.g., fast evacuation), the Wardrop equilibrium solution is fair for the evacuees of the same evacuation origin, i.e., the evacuation times of the paths chosen by evacuees from the same origin towards safe exits are similar. On the contrary, the travel times of evacuation paths might not be fair for different origins, i.e., the ratio of travel times of the evacuation paths between any two evacuation origins in the network of the same or similar distances can be arbitrarily high (see, e.g., [24,33].

Chen and Feng in [7] propose two heuristic flow control algorithms for a real-time building evacuation with multiple narrow doors: with no limitation on the number of evacuation paths and k required evacuation paths, respectively. Filippoupolitis and Gelenbe in [17] proposed a distributed system for the computation of shortest evacuation routes in real-time. The evacuation recommendations are computed by decision nodes and are communicated to the evacuees located in their vicinity. However, this approach considers only the cost proportional to physical distance that increases as the hazard progresses and does not take into consideration crowd congestion on the routes.

To provide for efficiency and fairness of route assignments in congested networks, in [30] and [33], Lujak et al. propose a traffic route optimization model focused on the system-optimum that considers the concepts of fairness and envy-freeness. Applying this approach to the evacuation, we can aim at the system optimum constrained by individual evacuee requirements and thus further improve the evacuation efficiency.

Computationally, the evacuation path finding has some similarities with the assignment of road routes to vehicles and the medical emergency management where patients are assigned to ambulances and hospitals. The application of the concept of distributed decision making (e.g., [45,54]) was proposed in medical emergency management [31,32] and traffic coordination [33] to increase the scalability in these domains. While the last three mentioned distributed coordination approaches find an efficient solution considering system optimum constrained by individual actors' requirements, [33] considers also fairness and envy-freeness of individual decision makers. These two concepts are crucial for avoiding panic and disturbance in the coordination of building evacuation, especially in the presence of evacuees with special needs.

The safety conditions during evacuation can change rapidly. Therefore, the communication of updated evacuation path information is important for safe evacuation of all the evacuees in a building under hazard. Possible communication devices can be, e.g., smartphone-like devices [53], variable message signs [43], LED displays installed on the building walls, etc. Since nowadays a smartphone is the most used communication and navigation device for outdoor locations and it can provide a context aware personalized real-time evacuation information, we consider an evacuation route guidance system that uses smartphones. In [25], common trends of architectural design, technologies, properties, and drawbacks of indoor positioning systems based on communications supported by smartphones are analyzed. Even though it is still not frequently used for inner–space navigation due to the open issues with indoor positioning (see, e.g., [12,25]), there are various works on the smartphone use for indoor route recommenders (see, e.g., [21,28,56]).

In [28], a light-weight application running on a smartphone for intelligent parking service is presented. It works based on a precise indoor positioning solution, which fuses WLAN signals and the measurements of the built-in sensors of the smartphones. The positioning accuracy, availability and reliability of the proposed positioning solution are adequate for facilitating indoor parking navigation. In [56], Yim presents an indoor-location-based, context-aware, and video-on-demand Android app that actively recommends showcases that the user most likely wants to visit in a museum. Another recommender system that arranges personalized visits through a museum was proposed in [21]. The visits are arranged based on user profiles and visitor location data provided by in-door localization techniques. Such situation-aware recommender systems can be considered as a special type of the current Context-Aware Recommender Systems (CARS) [2].

To provide evacuees with safe evacuation routes adaptive to changing hazard conditions, real–time computation and seamless handling of changing safety conditions are crucial. In [19], Guest et al. developed computer visual analytic system and real-time visual analytic tools for situationally aware evacuations of large urban structures. They used two different levels of detail in the representation of building graphs that can be used as a part of a visual analytic system for near real-time response. This in turn permits situational changes to be incorporated into the underlying models and rerouting of evacuees.

In [17], a distributed system for the computation of evacuation routes in real–time was proposed. The system is composed of a network of decision and sensor nodes distributed within the building. The former compute the evacuation routes in a distributed manner and communicate them to evacuees or rescue personnel in their vicinity. The concept of an "effective length" is given and it depends on the physical length of the link and the hazard present along the link. The system computes the shortest evacuation path in respect to its effective length. However, it does not consider possible congestion on the proposed evacuation paths and the influence of human factor on evacuation times.

The evacuation guidance system has to deal with humans, each one with possibly different individual characteristics. Due to these behavioral differences and to the human factor, generally, there is still a lack of coordination methods that are able to take human behavior into account. An evacuation route recommendation system that considers the panic behaviors of herding and stampeding in the evacuation routes was proposed in [35]. The system considers the influence of stress on human reactions to the recommended routes and iteratively ponders users' response to the suggested routes influenced by stress-related irrational behaviours until system acceptable routes are found. Moreover, the influence of affiliate ties and self-concerned individuals among evacuees on the evacuation performance was studied in [34]. Here, Lujak et al. model self-concerned and social group behaviour via individual and team reasoning. The recommended routes take into consideration the affiliate ties to guarantee evacuee's compliance with the routes.

As a continuation of the works [33,34,35] that deal with different aspects of building evacuation, in this paper we propose a distributed architecture that uses necessary sensory, localization, semantics, processing, and distributed optimization technologies that can provide real time situation–aware evacuation route guidance.

3. Used technologies

In this Section, we describe the technologies used for the localization of the evacuees in indoor environments as well as the technology for event processing.

3.1. Localization with landmarks

A prerequisite for avoiding congestion in finding evacuation routes is a detailed knowledge about the location of all persons in the building. An overview of indoor ultrasonic positioning systems with related state of the art can be found in [22] and [36]. In [36], Lymberopoulos et al. present the results from comparing 22 different technical approaches to indoor localization. They observe that all tested systems exhibited large accuracy variations across different evaluation points. This fact raises concerns about the stability and reliability of current indoor localization technologies. Some of the various technological approaches to localize persons in buildings are:

- *Wifi*. The intensity of a WiFi signal can be measured (RSSI received signal indication) to derive the distances to several access points, which allows calculating a person's position via trilateration. Unfortunately, WiFi does not yield good accuracy: the distance between a smartphone and a WiFi access point is often rather large and may not be precisely estimated on base of the RSSI, because the signal strength changes significantly with environmental conditions.
- *RFID*. Radio Frequency Identification technology can also be used for indoor positioning. Persons equipped with passive RFID tags can be detected by RFID readers that are spread in the building. RFID technology has several drawbacks: First, it is rather expensive to equip a building with an adequate number of RFID readers. That means that the number of RFID readers is relatively small and localization must also apply triangulation based on distance measures, which causes the same drawback as the one described above for WiFi. Secondly, it might be difficult to provide each person with a personal RFID tag.
- *iBeacon*. IBeacons support indoor navigation (see, e.g., [6,16]) by using Bluetooth low energy (a wireless personal area network technology marketed as bluetooth Smart) to send in a configurable frequency a unique ID that can be read by any smartphone. Therefore, an iBeacon infrastructure is set up easily: Beacons are cheap enough to distribute many of them, so that they can form a much denser network in the building. Furthermore, no specific beacon readers are necessary, because usual smartphones are capable of reading and processing beacon signals.

Table 1. Characteristics of indoor localization technologies

	Sender	Reader	Accuracy
WiFi	few senders per floor	1 reader per person	low
RFiD	1 sender per person	1 reader per room	medium
Beacon	many senders per room	1 reader per person	high

Table 3.1 summarizes the characteristics of indoor localization technologies. The superior accuracy of iBeacons is evident: there are as many readers as users, and each building part can be equipped with beacons resulting in a dense net of landmarks. Furthermore, iBeacons provide sufficient localization accuracy [8,40]. Therefore, we assume the presence of a sufficient number of beacons during evacuation to cover completely the building.

User smartphones The personal smartphone plays two different roles: as a reader of the iBeacon signals to localize its user and to exploit its built-in sensors so as to derive more details about the current situation of the user.

– *Beacon reader for localization:* In smartphone operating systems such as iOS and Android, the capability of reading iBeacon signals is already integrated. If we assume that the space is equipped with several iBeacons with non-overlapping ranges, as soon as a user approaches an iBeacon within the predefined range, the smartphone triggers an event carrying the iBeacon ID. Then the smartphone perceives that it is near that iBeacon and can forward this information to a server that coordinates emergency situations. An iBeacon ID is hierarchically structured, (i) a UUID specifies the particular institution (such as a university), (ii) a major ID could correspond to a certain building and (iii) a minor ID to a certain room.
– *User activity recognition:* The built-in sensors of a smartphone can be exploited to derive the current activity of its particular user. There exist multiple works on how to use smartphone sensors for performing activity recognition (see, e.g., [23,46,48,51]). For instance, in [27], different machine learning techniques are applied, such as, e.g., decision trees, logistic regression and neural networks to classify accelerometer data as certain activities. In evacuation, relating the current behavior of geographically close users is crucial to detect mass panic situations, e.g. the situation when most persons in a room are running. Furthermore, the smartphone serves as an individualized communication channel to its user for providing personalized routing guidance.

Other sensors and infrastructure More sensors are necessary for achieving situation awareness in the emergency recommender system. For instance, smoke and temperature sensors could be used for fire detection. The signals of these sensors could be collected and analysed on a centralized emergency management system. This system could also provide a central hub for the data of all user smartphones for calculating the global situation in a building such as room occupancy and general user behavior.

3.2. Complex Event Processing (CEP)

One of the key issues in emergency recommender systems is detailed knowledge about the current situation in the building. In our scenario, an appropriate and individualized guidance for all people in the building requires the information about:

- the smart space network structure and dimensions
- the current position of each person and the occupancies of all spaces in the building
- the situations that can provoke panic
- the space safety for each constituent part of the smart space network that can be jeopardized by, e.g., fire or built-up smoke, or panic related behaviors.

Apparently, such situational knowledge cannot be predefined, but must be inferred by exploiting real–time data. Usually, real–time data is provided by sensors, which monitor their environment and produce a continuous stream of data. In our scenario, we use smartphone sensors and further sensors that are permanently installed in the environment, such as iBeacons, temperature and smoke sensors. Each set of sensor data they emit corresponds to a particular event in the environment.

Situational knowledge can be considered as a dynamic knowledge with a high change frequency. In emergencies, streams of events must be evaluated in real-time to achieve situation awareness. A solitary event is usually of low significance since it represents a single incident in the physical world. For instance, it is of low importance if a single person is running, but if a majority of people is doing so, it may indicate mass panic.

Complex event processing (CEP) is a software technology that extracts the information value from event streams (see, e.g., [11,16,49]). CEP analyses continuous streams of incoming events in order to identify the presence of complex sequences of events (event patterns). The main goal of CEP is to derive the meaningful abstracted and complex (compound) events out of the observed streams of simple atomic and uncorrelated events [6]. In other words, CEP detects the relationships between relevant simple events from an event cloud and infers new single complex event with a significant meaning based on these events (see, e.g., [11,29]). For instance, a panic event can be inferred, if the smartphones of a majority of visitors in a certain area emit a running event.

Event stream processing systems manage the most recent set of events in memory and employ sliding windows and temporal operators to specify temporal relations between the events in the stream (each event has a timestamp). The core concept of CEP is a declarative event processing language (EPL) to express event processing rules. An event processing rule contains two parts: a condition part describing the requirements for firing the rule and an action part that is performed if the condition matches. The condition is defined by an event pattern using several operators and further constraints.

In the following, we use a simplified pseudo language for expressing event processing rules, which is easier to understand than an EPL of a productive CEP system. This pseudo language supports the following operators:

$\vee$ $\wedge$	Boolean operators for events or constraints
NOT	Negation of a constraint
$\rightarrow$	Sequence of events ($e_1 \rightarrow e_2$ means that e_1 occurred before e_2)
Timer	Timer(time) defines the time to wait
	Timer.at(daytime) is a specific (optionally periodic) point of time
.within	defines the time window in which the event has to occur.

An event processing engine analyses the stream of incoming events and executes the matching rules. Luckham introduced the concept of event processing agents (EPA) that communicate to each other by exchanging events [29]. An EPA is an individual CEP component with its own rule engine and rule base. Several EPAs can be connected to an event processing network (EPN) that constitutes a software architecture for event processing.

4. Evacuation coordination problem

If real-time building information is available to evacuees and they can negotiate their evacuation, it becomes possible to provide a selection of safe and efficient routes. Therefore, we assume that the building and evacuees are monitored in real–time by a strategically positioned network of sensors through indoor localization systems and tracking technologies described in Section 3.

Starting from the above stated assumption, let us define a building for evacuation. Let $G = (N, A)$ be a connected digraph representing the smart building network where N is the set of n vertices representing offices, halls, and in general, any portion of space within a building separated by walls or partitions from other parts. In the case of larger spaces, for simplicity, the same are divided into regions represented by nodes completely connected by arcs $a \in A$, where A is the set of arcs $a = (i, j)$, $i, j \in N$ and $i \neq j$, representing doors, gates or passages connecting nodes i and j.

For every arc $(i, j) \in A$, there is an arc cost function $f_{ij}(x_{ij})$, which without the loss of generality, we assume is proportional to an average travel time function $t_{ij}(x_{ij})$, where x_{ij} is a person flow per time unit, i.e., $\sharp per/min$. $t_{ij}(x_{ij})$ is in general an increasing nonlinear function because of the effects of congestion on the arc travel time. Different functions can be considered, but for simplicity and without loss of generality, we consider the average travel time function proposed for normal use by the U.S. Federal Highway Administration traffic assignment model: $t_{ij}(x_{ij}) = \tau_{ij}(1 + 0.15(x_{ij}/u_{ij})^4)$ [37], where u_{ij} is the capacity, and τ_{ij} is the travel time of arc $(i, j) \in A$ in free flow conditions (without congestion).

Let $O \subseteq N$ and $D \subseteq N$ be the set of all evacuation origins and destinations respectively. We assume that there are $|O|$ origin nodes disjoint from $|D|$ destination nodes, where $|O| + |D| \leq n$. Here, origins are all areas with evacuees inside the smart space network while destinations are their near safe building exits. To represent the relation between all safe exits in the graph, we introduce fictitious sink node $\hat{d} \in N$ that is adjacent to all the destination nodes (safe exits) by fictitious (dummy) arcs (with zero cost and infinite capacity). In this way, we assume that graph G includes (together with actual nodes) also fictitious node $\hat{d}$ and its incoming dummy arcs.

Then, let R_o represent a number of evacuees (evacuation requests) per time unit who request to leave origin node $o \in O$ to go to any of the safe exits $d \in D$ and, hence, to fictitious destination $\hat{d}$. Furthermore, we assume that the evacuation requests are detected by the sensor network in the building at the beginning of the time window and their variations are negligible throughout the window.

Our objective is to safely evacuate as many evacuees as possible from all origins $o \in O$ over the safest and the most efficient evacuation paths to any of the safe exits $d \in D$ and, hence, to fictitious destination $\hat{d}$. In that respect, let P_o be a set of simple paths from origin o, where $o \in O$, towards fictitious destination $\hat{d}$. Then, if we model the evacuation as a unified crowd flow, each individual is seen as a unit element (particle) of that flow and the objective is to maximize the flow of demands (evacuation requests) with certain constraints we consider in the following.

Let us assume that safety status S_a is given for each arc $a \in A$ as a function of safety conditions that can be jeopardized by a hazard. We normalize it to the range $[0, 1]$, such that 1 represents perfect conditions while 0 represents conditions impossible for survival, with a critical level for survival $0 < S_a^{cr} < 1$ depending on the combination of the previously mentioned parameters. More details on data quantizing and fusion whose result is the arc safety status can be found in Section 5.2.

If each constituent arc $a \in k$ of a generic path k has safety $S_{a \in k} \geq S^{cr}$, then path k is considered to be safe. On the contrary, path is considered unsafe and its harmful effects may threaten the evacuees' lives. The proposed evacuation paths should all satisfy safety conditions $S^k \geq S^{cr}$. However, when such a path is not available, a path with the maximal safety should be proposed where the travel time passed in the safety jeopardized areas should be minimized. Since safety may vary throughout a path, we introduce a normalized path safety that balances the minimal and average arcs' safety values:

$$S^k = \sqrt[|a \in k|]{\prod_{a \in k} S_a}, \ \forall k \in P_o, \ o \in O. \tag{1}$$

Then, let $\bar{P}_o$ be a set of available safe simple paths with a maximized safety (1) acceptable in terms of duration in free flow for each evacuation origin $o \in O$. By acceptable in terms of duration in free flow, we mean the paths whose traversal time t_k in free flow is within an upper bound in respect to the minimum free flow duration among all the available evacuation paths for that origin. Let $\bar{P}_O$ be the set of all such paths. From set $\bar{P}_O$, we want to find the paths that, considering congestion produced by evacuation requests R_o for all $o \in O$, are temporally efficient and envy-free.

The concept of envy-free paths is introduced in [33]. A path allocation θ is *α-envy-free*, where α is a maximum tolerance factor for non-enviousness ($0 < \alpha \leq 1$) if:

$$\gamma_o \geq \gamma_{o'}^{\alpha}, \ \forall o, o' \in O | o \neq o', \tag{2}$$

where γ_o is a normalized mean path duration cost of each evacuation origin $o \in O$ defined as $\gamma_o(x_o, \{x^l\}_{l \in \mathcal{M}(o)}) = \sqrt[|\bar{P}_o|]{\prod_{k \in \bar{P}_o} t^k \cdot x^k}$ and $\mathcal{M}(o)$ is a set of the origins whose paths use one or more same arc(s) as $o \in O$ and are therefore coupled with it. In other words, Formula 2 says that there is no evacuee at origin o' that envies any other evacuee at origin o for getting assigned the path with a lower duration than α^{th} power of the path duration assigned to the evacuee on o'.

In the following, for the self-completeness of this work, we present an adapted mathematical programming model from [33] for the optimization of evacuation routes through Nash social welfare maximization with included envy-freeness constraints:

(N) :

$$\min z(\mathbf{x}_O) = \sum_{o \in O} \log \left[\sqrt[|\bar{P}_o|]{\prod_{k \in \bar{P}_o} \sum_{a \in A} t_a(x_a) \cdot \phi_{ak} x^k} \right] \tag{3}$$

subject to:

$$\sum_{o \in O} \sum_{k \in \bar{P}_o} \phi_{ak} \cdot x^k \leq u_a \ , \forall a \in A \tag{4}$$

$$\gamma_o \geq \gamma_{o'}^{\alpha}, \ \forall o, o' \in O | o \neq o' \tag{5}$$

$$\sum_{k \in \bar{P}_o} \psi_{ok} \cdot x^k = R_o, \ \forall o \in O \tag{6}$$

$$x^k \geq 0 \ , \ \forall k \in \bar{P}_o, o \in O \ , \tag{7}$$

where Φ is the $[|A| * |\bar{P}_O|]$ arc-path incidence matrix, and Ψ, the $[|O| * |\bar{P}_O|]$ evacuation origin-path incidence matrix. Capacity constraints (4) limit the total flow across all paths passing through each arc $a \in A$. Furthermore, (5) is a constraint on envy-free evacuation origin paths while through constraints (6), we model the fulfillment of evacuation requests among path flows by forcing the sum of path flows of each commodity $o \in O$ to be equal to the commodity demand (the number of evacuation requests R_o). (7) is a constraint on non-negative values of path flow x^k for each $k \in \bar{P}_o$, $o \in O$.

5. Proposed distributed architecture for evacuation guidance

In this Section, we present the proposed distributed ERG system and describe the components comprising it. Then, we give the details on the elements of CEP.

5.1. System architecture

We propose a solution concept for a distributed evacuation route guidance system that combines different CEP modules in order to provide situation awareness for a distributed evacuation route recommendation algorithm that is explained in Section 6. An overview of this architecture is given in Figure 1.

The objective of evacuation route guidance architecture (ERGA) is to provide individualized route guidance to evacuees over an app on their smartphones that is connected with a cloud (see, e.g., [5]) based on the evacuation information received from connected smartphones within the building and the building sensor network. It should be activated on request by any connected user within the building or remotely by the sensor network in the building, hence informing evacuees of the ongoing evacuation. However, in the case an evacuee does not have a smartphone, he/she could still follow the evacuation directions on LED displays on the walls of a smart building.

ERGA consists of user agents (UA) and a network of smart building (SB) agents. As stated earlier, a smart building is represented by a graph $G = (N, A)$. Each node in N represents a physical space and is represented by an SB agent. The arcs between the nodes represent the evacuees' movement options and, in the same time, the communication channels between neighboring SB agents. The SB agents are a constituent part of a smart building and process and sense their individually assigned physical space over a strategically distributed network of sensors within that space. In the case there are evacuees present within the space, an SB agent takes a role of the evacuation origin agent.

User agents The user agent is associated with the application on a smartphone of an evacuee (see Figure 1). It manages and stores all the information that is related to a specific evacuee in the building. The UA is intended to be executed as an app on the smartphone of an evacuee. Here, we assume that people that enter the building own a smartphone with the evacuation app installed, or they have been provided with some smartphone–like device that runs the app when they start to evacuate. The user agent contains three parts:

- module with user preferences and constraints,
- user situation awareness module, and
- route guidance module.

The *user preferences and constraints* module allows defining constraints such as disabilities (e.g., the use of wheelchair or vision impairment) as well as evacuation–related behavioral disorders (e.g., agoraphobia, social phobia, etc.), while the preferences include the affiliate ties with other users of the building. We assume that a user agent possesses complete local situation awareness of an evacuee that is deduced in the *User Situation Awareness* module and is sent to the closest SB agent together with user data including his/her preferences and constraints.

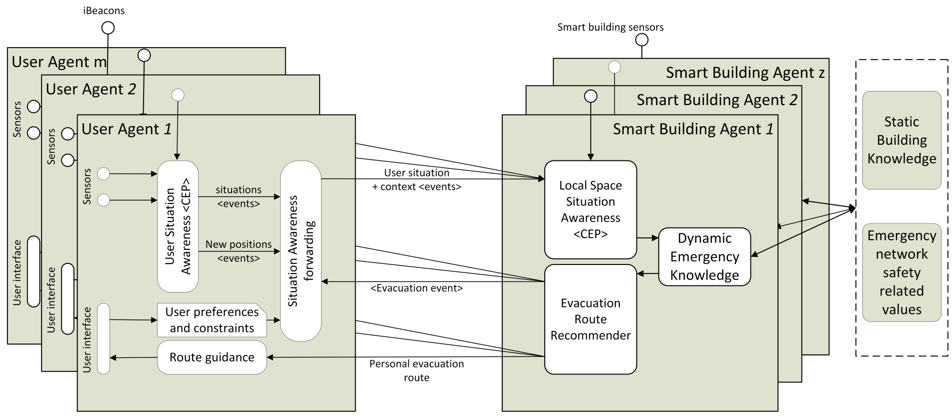

Fig. 1. Situation-aware real-time distributed evacuation route guidance architecture (ERGA). User agents $1, 2$, and m are located in the physical space of SB Agent 1 so that they are given route recommendations by SB Agent 1

The *user situation awareness* module exploits sensor data (from the smartphone and beacons installed in the building) and reasons about the behaviour and location of the user through local CEP processes. The presence of an evacuee together with the information derived from the situation awareness module and the individual preferences and constraints are passed to the local space situation awareness module in the closest SB agent. In order to assure privacy, only certain basic data about the user's situation should be forwarded to the SB agent (e.g., location, running events).

Every user agent informs the closest SB agent of its personal information, i.e., position, evacuation constraints, and preferences. Subsequently, the SB agent takes a role of evacuation origin agent and sends the route guidance to UA from its *evacuation route recommender* module described in detail in section 6. Finally, the *user interface* provides the user with personalized navigation guidelines for evacuation, helping him/her to leave the building in the way calculated by the SB agent.

Smart building agents The network of smart building agents is the central computational and perception part of the evacuation route guidance system. The situation awareness and decision making are distributed in the network of SB agents such that each agent is responsible of the semantic reasoning concerning the safety of its assigned physical space. Then, the qualitative safety status of the space is abstracted to normalized safety values in the range $[0, 1]$. Moreover, each SB agent is responsible of the evacuation route computation for the evacuees positioned in its physical space. It is the link between the distributed SB agent network and an individual evacuee (see Figure 1).

The SB agent stores its user agents' information as a part of its initialization since the frequency of changes in the user's description is likely to be very low. Furthermore, we assume that each SB agent has at its disposal the information regarding all evacuation network's layout, topology and safety. Furthermore, each SB agent is responsible of evaluating the cost functions of the evacuees within its physical space and thereby determining the optimal route for each such evacuee.

A single SB agent controls only its own physical space within its realm of influence, for example, whose maximal surface is up to l_{max}. If the space is larger than that, then the space is partitioned into regions controlled by more SB agents. Each SB agent has a corresponding region (Voronoi cell) consisting of all user agents closer to that SB agent than to any other SB agent within the same room. Each SB agent contains a local space situation awareness module that perceives the safety conditions of the physical space it controls through combining and analysing the events provided by the sensors of the smart space it represents and the individual user agents that are located within the physical space controlled by it. Moreover, each SB agent communicates with its neighboring SB agents and with the user agents present within its physical space.

The local space situation awareness module functions in cycles. At the first phase, the local building sensor data is fused with the data from the locally present user agents. Then, the normalized safety value is deduced through CEP events. This data is sent to a blackboard or alike globally shared data structure containing the overall network safety values and visible to all agents. Thus, the global situational awareness of the building is accessible to every SB agent by accessing the blackboard. Once the data regarding all the evacuees present within the physical space controlled by an SB agent is gathered by the SB agent, it starts the computation of the safe and efficient evacuation routes.

To relieve the communication load, each SB agent maintains a local copy of the evacuation network with the updated safety conditions. Only the changes in the local safety values and evacuee conditions are broadcasted while the detailed local space situation awareness information is exchanged only between neighboring SB agents, if necessary. In this process, CEP is used to filter irrelevant information and to generate higher level events. Individual user events are aggregated to detect events regarding groups of users, their distribution and density in the building.

When an SB agent detects an emergency situation, it sends the updated safety value of its physical space to the shared blackboard. This allows, on the one hand, to monitor the real-time situation of the building and, on the other hand, to trigger an evacuation process and to execute control actions in such a process. Each SB agent computes the routes for the evacuees that are momentarily within its physical space. In the computation of their optimal routes, it considers their preferences, requirements, and the evacuation conditions. With the known evacuees' and the space safety information, the SB agent's evacuation route recommender module can compute optimized evacuation routes for each locally present user agent. It does so by distributed computation and communication with the rest of the SB agents in a multi-hop fashion. In this process, the algorithm uses:

- Data regarding the building topology: Static information about physical elements in the building (e.g. rooms, corridors, floors, doors, etc.) and relation among them (e.g. the area of room A is 10 m^2; room A is next to room B and they are both at floor F). Topology knowledge is represented in such a way that it is sufficient to describe the building network by a digraph with weights and tags on the constituent nodes and connecting arcs as described in Section 4. Nodes and arcs are described by their type, surface, area, inclination, etc.
- Emergency ontology: This ontology contains general knowledge about emergency and evacuation scenarios, e.g., facts that people with strong affiliate ties should always

be evacuated together (for instance, families with children and persons with disability and their assistants), the appropriateness of certain routes for people with limited mobility in emergency situations, and the influence of certain events like fire and smoke on the security level of an evacuation space.

- Real-time evacuation situation: Contains the current physical space situation awareness of the SB agent itself as well as regarding the evacuees that are currently in the space represented by the SB agent. This information includes:
  - The number and distribution of people in the space,
  - Evacuation preferences and constraints of each person,
  - Evacuation network's safety values and topology regarding the evacuees' preferences and constraints.

During evacuation, the global safety situation of the building is dynamically updated in real–time in order to reflect the dynamically changing safety conditions of the building. In the same way, each SB agent recalculates the evacuation routes each time the changes in respect to the evacuees or the safety of the evacuation network occur (i.e., each time new events are detected).

5.2. CEP components

Both UA and SB agents analyse the incoming streams of events to deduce the current situation. In this subsection, we discuss the underlying event models and give some examples for appropriate rules for achieving situation awareness. To make the description more comprehensive, we simplify the event model and the corresponding rules.

CEP in the user agents User agent exploits sensor data and infers (i) the location and (ii) the behavior of a single user. To explain the CEP component in more detail, we will assume that user agent monitors two types of explicit (or atomic) events to achieve this type of situation awareness:

- *beaconEvent(beaconID)*: an iBeacon with a certain ID has been detected.
- *accelerationEvent(velocity)*: the smartphone is moving with a certain velocity.

The *beaconEvents* collected by a particular smartphone are used to derive the current position of its owner. The following CEP rule creates *enteringSpace* and *leavingSpace* events, meaning that the user is entering, respectively leaving certain space. These events can be considered as complex (or materialized) events. They carry the ID of the user and the related beacon ID.

CONDITION	beaconEvent AS $b_1 \rightarrow beaconEvent\ AS\ b_2$
	$\wedge b_1.id \neq b_2.id$
ACTION	CREATE enteringSpace(userID, $b_2.location$)
	CREATE leavingSpace(userID, $b_1.location$)

The rule describes the situation that a new *beaconEvent* b_2 has been read in the smartphone, where the beacon ID has changed. (Here the beacon ID, more precisely its minor ID, corresponds with a space representing that part of the building.)

Detecting a running user is another situation that must be forwarded to the SB agent, because many running users can indicate a panic situation. An appropriate CEP rule checks if the average velocity of a user is higher than 5 km/h considering a time window of 10 seconds:

CONDITION	accelerationEvent AS a [win:time:10sec]
	$\wedge$ average(a,velocity) $>$ 5 km/h
ACTION	CREATE runningEvent(userID)

If the condition is fulfilled, then this rule creates a *runningEvent* that contains the ID of the corresponding user.

CEP in the smart building agents The CEP component in the SB agent is responsible for deriving the situation in its physical space. For instance, it could receive and analyze the following atomic events: produced by the CEP rules running on the users smartphones.

- *enteringSpace(userID, space)*: a user with a certain ID has entered a certain space.
- *leavingSpace(userID, space)*: a user with a certain ID has left a certain space.
- *runningEvent(userID)*: a user with a certain ID is running.

Another kind of situation awareness describes the global situation. A first type of rules calculates the occupancy of the space controlled by the SB agent.

The following CEP rule calculates the number of persons located at a certain space by counting all entries and exits registered in that space during the last 15 minutes:

CONDITION	(enteringSpace AS e $\vee$ leavingSpace AS l)
	[win:batch:15min]
	$\wedge$ count(e) AS enters
	$\wedge$ count(l) AS exits
ACTION	CREATE occupancy(e.space, enters-exits)

The second type of rules infers a global behavior of the people present currently in the building. For instance, the next rule intends to detect a panic situation in the building:

CONDITION	runningEvent AS r [win:time:1 min]
	occupancy AS o
	$\wedge\ count(r) > o.value \cdot 0.2$
ACTION	CREATE panicEvent(r.space)

It groups all *runningEvent*s according to a time-spatial window. The grouping criterion is defined by a time interval of 1 minute. If more than 20% of the people staying in a space are running, a panic situation is indicated. Note that also other situations could be detected by appropriate CEP rules. For instance, a blocked staircase could be inferred if numerous persons could not continue their recommended evacuation path along the staircase. Furthermore, there are other sensors in the smart building that can be exploited to derive certain building states. For instance, the data from temperature and smoke sensors can be used to detect a fire situation in a certain space of the building. There are appropriate CEP rules that derive such situations as well. In the following subsection we show how sensors can be used to obtain a safety value for the space in which they are located.

Safety calculation Space safety is calculated by combining (aggregating) the values obtained from different sensors. A thorough description of this field can be found in, e.g., [26,57]. Since this is not a main topic of this paper, here we propose a way of calculating safety values in the case of fire so as to get a feeling of how sensor data and CEP rules can be combined.

We assume that a fire is detected by smoke and temperature sensors. The number of sensors of each type in the same physical space usually depends on the size of the space. For simplicity, we assume that smoke sensors return a value in the range $[0, 1]$, (0 - no smoke and 1 - maximum smoke), while temperature (in centigrades) is in the range $[0, 100]$. We consider normal (safe) temperatures to be up to 40 centigrades so we normalize the temperature with the following formula:

$$T^{norm} = \begin{cases} 0 & \text{if } T^{real} \leq 40°C, \\ (T^{real} - 40)/60 & \text{otherwise,} \end{cases} \tag{8}$$

such that the resulting normalized temperature T^{norm} obtained from the real temperature readings T^{real} is in the range $[0, 1]$. The idea is to aggregate the values read by different sensors in the same space in such a way that the safety value decreases with the increase of the number of sensors that detect an incident.

We propose the aggregation function $f(x, y) = x + y - x \times y$ (also known as probabilistic sum), where $x, y \in [0, 1]$. This function is associative, commutative and the result of combining x with y is always higher or equal to x and y. This means that each sensor will support and increase the evidence of a lack of safety or jeopardy for evacuees. In this light, let jeopardy J be defined as $J = 1 - S$, with the range $[0, 1]$, where 0 represents full safety and 1 a hazardous area. For example, if two smoke sensors report values $ss_1 = 0.6$ and $ss_2 = 0.7$, and a temperature sensor detects $T^{real} = 50°C$ ($T^{norm} = 0.17$), the resulting safety is calculated as follows: based on the smoke sensors, jeopardy is $J(ss_1, ss_2) = 0.6 + 0.7 - 0.6 \times 0.7 = 0.88$. Combined with the temperature sensor, $J(ss, ts) = 0.88 + 0.17 - 0.88 \times 0.17 = 0.9$. Safety is then: $S(ss, ts) = 1 - J(ss, ts) = 1 - 0.9 = 0.1$.

We assume that information about temperature and smoke sensors is represented by the following type of events:

- *temperatureEvent(sensorID, value)*: temperature $value$ read by sensor $sensorID$.
- *smokeEvent(sensorID, value)*: smoke $value$ detected by sensor $sensorID$.

The following rule detects and normalizes an abnormal high temperature (higher than $40°C$) identified by a sensor. Note that, for some reason, different SB agents might define different thresholds and/or normalization functions.

```
CONDITION temperatureEvent AS t
          ∧ t.value > 40°C
ACTION    CREATE highTemperature((t.value - 40)/60)
```

The next type of rule obtains the safety value by aggregating smoke and high temperature events according to the function described above, considering a time window of 30 seconds:

```
CONDITION (highTemperature AS t ∨ smokeEvent AS s)
          [win:time:30sec]
ACTION    CREATE safety(1 - window(*).aggregate(0, (result, value) =>
          result + value − result × value))
```

The *aggregate* enumeration method takes an expression providing the initialization value (0) and an accumulator expression. The return value is the final accumulator value, i.e. the safety of the space controlled by the SB agent.

6. Evacuation route recommender module

Finding evacuation routes in a building considering the impact of congestion and envy-freeness is coupled spatially. The objective is a robust and scalable decision-making support that will align the choices of the routes by self–interested evacuees with the goal of overall evacuation system efficiency considering crowdedness. To achieve that objective, we propose a distributed decision–making approach that maintains consistency in all of the routing decisions while distributing the overall route computation process among SB agents that compute routes with a minimum of synchronizations.

With that scope, we implement an adapted version of the evacuation route recommender module that was presented in [35] and described briefly in the following. The recommender module is made of two parts: the routes' safety optimization and the routes travel time system optimization considering congestion and fairness (see Fig. 2).

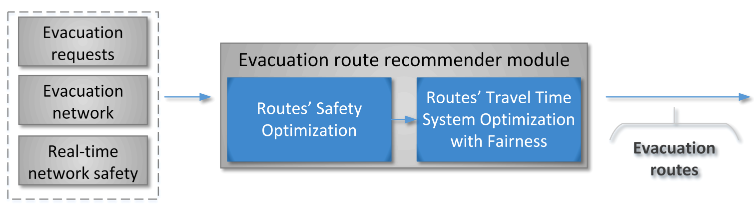

Fig. 2. Evacuation route recommender module

Routes' safety optimization In this part, each SB node representing an origin $o \in O$ with evacuation requests $R_o > 0$ computes a set $\bar{P}_o$ of k simple safest paths, (the paths with minimum jeopardy) that are also efficient (acceptable in terms of travel duration).

To discourage the usage of unsafe arcs with $S_a < S^{cr}$, in the preprocessing step before computing the paths, we multiply the travel time of unsafe arcs by M^{-S_a}, where M is a very large number. In this way, the unsafe arcs will be included in the shortest paths only if there is no alternative path composed of safe arcs. Moreover, the number of the unsafe arcs in proposed paths will be minimal and their safety value will be maximal. Then, a possible algorithm to use for the computation of k fastest simple paths is, e.g., Yen's algorithm [55]. The found set of safest efficient paths is ordered in a non-decreasing order of free-flow travel time and to obtain set $\bar{P}_o$ of available (simple) safe efficient paths, the paths within a predefined upper bound $\alpha \geq 1$ in respect to the fastest path are selected.

Routes' travel time system optimization with fairness The route's travel time system optimization with fairness considers possible crowdedness and is divided into two phases, Figure 3. At the first phase, Nash social welfare maximization problem (3)-(7) with included envy-freeness constraints is decomposed to obtain subproblems that will be optimized independently locally by each SB agent through local computations and mutual communication of relevant data with other SB agents. The decomposition is done at four levels, one level for each shared constraint and the routing solution for all evacuees is achieved by an auction-like primal-dual optimization method among the SB agents through the exchange of the local variables until an acceptable system–wide solution is found. The details on the optimization approach can be found in [33]. However, for the self-completeness of this work, we bring its short description in the following. At the beginning of the evacuation route's duration optimization, every route is available and every evacuee located at his/her initial position is uncommitted with respect to the evacuation routes. Following initialization, every SB agent with evacuees present within its physical space is fully aware of: i) its evacuees with their evacuation constraints and preferences and ii) entire network structure with arcs' travel times and safety values.

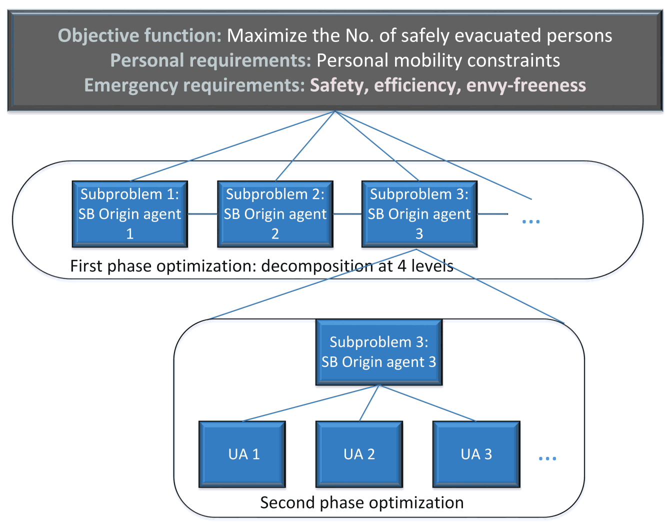

Fig. 3. Two phases of the distributed evacuation route computation

Based on the evacuation requests R_{SB} expressed in terms of person flow per time unit, each SB agent tries to achieve a sufficient number of shortest paths considering free flow (no congestion). Thereafter, each SB agent propagates its evacuation requests to the intermediate nodes of the chosen paths. These requests are implemented in the form of a message and are sent by SB agents in a multi-hop fashion through intermediate nodes up to the fictitious sink node.

In the following, we describe the decomposition at four levels. At the first level, SB agents define arcs' dual variables (Lagrange multipliers that can be seen as arcs' prices) to handle flow arc capacity constraints. The prices of the evacuation network's arcs are adjusted based on the SBs' overall demand on the routes influencing congestion. At the second level, SB agents with present evacuees act as the evacuees' evacuation origin agents and negotiate with other evacuation origin agents the assignment of envy-free paths.

Each evacuation origin agent has a local information of the flows along the paths of other origins that share with it the same arcs and/or nodes. At the third level, the subproblem regarding the consistency of this local information is solved within each cluster of the paths sharing one or more arcs and/or nodes. Thus the computation of the main problem (3)-(7) is distributed into $|O|$ subproblems, each one related to an SB agent with present evacuees. At the fourth level, the Lagrangean relaxation of problem (3)-(7) resolves the constraint on distribution of demanded flow R_o over paths $k \in \bar{P}_o$, for each SB agent representing evacuation origin $o \in O$.

The SB agents on the intermediate nodes of the paths of other SB agents compute the travel times due to the congestion on their outgoing arcs as soon as they receive new evacuation requests and resolve conflicts in terms of the prices in a distributed way as explained in [33].

In each iteration, each evacuation origin agent calculates optimal evacuation paths based on the set of momentary prices for each level of decomposition. The dual variables converge to their dual optimal values as long as the lower level problems are solved on a faster time scale than the higher level ones so that all the problems at a lower level have converged at each iteration of a master problem [33].

After the route assignment is made for all evacuation requests by evacuation origin agents at the first phase of the optimization model, each such SB agent decides, at the second phase, of the assignment of its evacuees to the routes assigned to it at the first phase. This is done based on relevant personal characteristics and social welfare parameters that guarantee fairness of the assignment of the evacuees to the routes through an iterative auction, see Figure 3. The negotiation through auctions at the second phase is local between each evacuation origin agent and the evacuees momentarily present at that origin, similar to [33].

To avoid the movement of the congestion from one point to another (see, e.g., [52]), the evacuation route is re-computed for each user every time he/she changes his/her physical location with a related SB agent or the safety conditions of the evacuation path change. The resulting route guidance is personalized considering the relevant user's factors and the influence of the evacuees' routes on each other.

7. Case study

We show the functioning of the architecture on a simple assumed evacuation scenario presented in Figure 4. Given is an evacuation network with 9 SB agents (nodes of the graph) that represent 5 different physically limited spaces (halls) $A, \ldots, E$ and an evacuation staircase F. This layout can be a representation, for example, of a movie multiplex with 4 different movie theaters.

As can be seen from Figure 4, halls A, B, and D and staircase F are represented only by one SB agent since their total surface is lower than l_{max}. However, due to a too large size, halls C and E are represented by two and three SB agents, respectively. When needed, we will denote the part of spaces represented by different SB agents as, e.g., e_5 (area of hall E managed by SB agent 5).

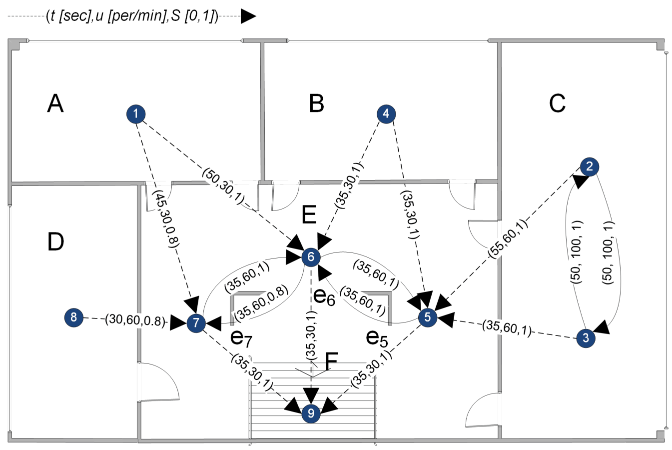

Fig. 4. Example of an evacuation network modelling from the smart building floor plan. Arc labels consist of three numeric values: travel time t_a [sec], capacity u_a [♯ per/min] and safety S_a [0,1]

A separate arc in each direction is created for every two communicating neighboring evacuation spaces and, correspondingly, SB agents. For simplicity, in Figure 4, we draw just outgoing arcs from halls $A, \ldots, D$ to the main hall E, and similarly, from hall E to staircase F. The capacity of every arc is $30p/min$ except for the arcs connecting space D with E whose capacity is $u_{87} = 60p/min$ and of the arcs connecting space E with the staircase F, $u_{59} = u_{69} = u_{79} = 30p/min$, Figure 4.

We assume that the critical safety for survival is $S^{cr} = 0.5$. Initially, during the normal operation of the building and before any incident started, safety of all halls was intact and its value was 1, i.e. $S_A = S_B = S_{C_2} = S_{C_3} = S_D = S_{E_5} = S_{E_6} = S_{E_7} = S_F = 1$, which translates into the value of the safety of all the arcs in the network equal to 1.

Let us assume that, due to a malfunction on an electrical installation, a fire began at hall A. We assume there are several smoke (ss) and temperature (ts) sensors installed in different parts of the space controlled by each SB agent. In particular, agent SB 1 responsible of hall A contains smoke sensors ss_1^1 and ss_1^2, and temperature sensors ts_1^1 and ts_1^2 that detect smoke and high temperature reading the following values: $ss_1^1 = 0.2$, $ss_1^2 = 0.3$, $ts_1^1 = 50°C$, and $ts_1^2 = 40°C$. These data are introduced into the event stream as: [..., $smokeEvent(ss_1^1, 0.2)$, $smokeEvent(ss_1^2, 0.3)$, $temperatureEvent(ts_1^1, 50)$, $temperatureEvent(ts_1^2, 40)$, ...].

CEP rule that detects abnormal high temperature is triggered only once with an event from sensor ts_1^1. It generates and includes the following event into the stream: $highTemperature(0.17)$.

Sequentially, another CEP rule obtains the jeopardy value by aggregating values from smoke sensors: $J(ss_1^1, ss_1^2) = 0.2 + 0.3 - 0.2 \times 0.3 = 0.44$. The resulting value is combined with the temperature sensor, $J(ss_1, ts_1^1) = J_A = 0.44 + 0.17 - 0.44 \times 0.17 = 0.54$. Safety of hall A is computed as $S_A = 1 - J_A = 1 - 0.54 = 0.46$.

Therefore, shortly after the incident, the safety of hall A fell to 0.46 (below S^{cr}) and the building must be evacuated. The fire starts to extend to the neighboring halls B, D

and e_7 (since the walls and doors are not fire–proof). Thus, the safeties of these halls are starting to decrease and momentarily their values are $S_B = 0.8$, $S_D = 0.7$ and $S_{E_7} = 0.8$ (obtained by CEP rules analogously to the previous example). The rest of the halls maintain their safety value intact, including areas e_5 and e_6 (since the fire propagates from the left part of E and is not detected in these parts of the E hall).

Let us assume that there are 300 evacuees in hall A; 400 evacuees in hall B; 700 in hall C, 100 in hall D, and 300 in hall E. Furthermore, let us assume that evacuees are distributed uniformly randomly in each hall. The maximum evacuation time of the whole building given by the evacuation coordinator is 30 minutes. This means that the evacuation request of agent SB 1, R_1, is 10 persons/min, for SB 2, R_2= 11.7 persons/min, R_3=11.7 persons/min for SB 3, R_4=26.8 persons/min for SB 4, and R_8=3.4 persons/min for SB 8.

Evacuation route recommendation should be given to the evacuees such that they all evacuate safely considering personal and evacuation constraints introduced previously. SB agents jointly compute safe, efficient and envy-free routes in the following way. Firstly, each SB agent with present evacuation requests finds safe routes. Since in Figure 4, all arcs are with safeties $S_a > S_{cr}$, at this level, all physical paths are possible.

At the next step, we perform routes' travel time system optimization with fairness. If we assume that the tolerance factor for the temporal efficiency is 15%, i.e., $\alpha = 1.15$, then initially each SB agent finds a set of temporally efficient paths from the set of safe paths. In more detail, safe temporally efficient paths for SB 1 are $p_{11} = ((1,6),(6,9))$ with path's free flow duration $t_{11} = 80$ sec and $p_{12} = ((1,7),(7,9))$ with the duration $t_{12} = 85$ sec. There is only one efficient path for SB 2, $p_2 = ((2,5),(5,9))$ with the duration $t_2 = 90$ sec, and one such path for SB 3, $p_3 = ((3,5),(5,9))$ with the duration $t_3 = 70$ sec. Moreover, SB 4 contains two safe efficient paths, $p_{41} = ((4,5),(5,9)$ and $p_{42} = ((4,6),(6,9))$, both with free-flow duration $t_{41} = t_{42} = 70$ sec. Agent SB 8 has only one efficient path $p_8 = ((8,7),(7,9))$ with the duration $t = 65$ sec.

SB agents with evacuation requests send the requests for their safe efficient paths to all the SB agents on the intermediate paths' nodes. Then SB agents decompose the problem at four levels and compute dual prices for their outgoing arcs based on the congestion, capacity, consistency, and envy-freeness constraints as described in Section 6. In particular, initially, the accumulation of the paths' passage requests are satisfied from the congestion point of view everywhere except for arc $(5,9)$ that has capacity 30, and the accumulated requests on the paths of agents SB 2, SB 3, and SB 4 are 30.1 pers/min. This is why the flow of agent SB 4 needs to decrease on path $((4,5),(5,9))$ at the cost of path $((4,6),(6,9))$ that has an increase of 7.5 persons per minute.

Final route flows for SB agents are as follows. SB 1, 4.7 per/min on path p_{11}, and 5.3 per/min on path p_{12}. SB 2, SB 3, and SB 8 agents direct all their evacuation requests on their single paths. SB agent 4, 14,7 per/min on path p_{41} and 12,1 per/min on path p_{42}. Based on these values and the individual evacuation requirements of the evacuees, in the second phase, each SB agent assigns the routes it was assigned on the first phase to the evacuees present in its physical space.

8. Conclusions

In this paper, we presented a distributed evacuation route guidance architecture (ERGA) for large smart buildings. Our approach takes into account the evacuees' current location and building safety obtained by a smart building sensor network and personal mobile devices to recommend the best evacuation route for each localized evacuee. The ERGA always computes and recommends the most efficient safe paths for the evacuees. As the evacuees move and change their position, the recommendations change based on their momentary positions independently of whether the evacuees follow the recommended routes or not.

The organizational structure of the proposed architecture includes a network of smart building agents and user agents. Each SB agent is responsible of the computation of a personalized evacuation route for each user agent within the physical area under its control. Since, here, we deal with a highly computationally complex problem, the implementation of the proposed distributed sensing and computation in the SB agents adds scalability and robustness to the route computation. The evacuation route optimization approach first considers the safety of evacuation routes and then their temporal efficiency. Therefore, first, a set of the safest routes is found and then, the routes that are temporally efficient are selected within this set. By temporally efficient, we mean the routes that are acceptable from the point of view of their total travel time knowing the walking speed.

To implement this architecture in a real-world context, we propose the usage of the following technologies: iBeacons and smartphones for obtaining real-time evacuees' and building safety information, CEP for complex event processing, and a distributed optimization algorithm for efficient route computation. With the usage of iBeacon technology for the recognition of the evacuees, only the evacuees with a smartphone and the installed app will be recognized by the system. However, if additional sensors are used in the building, e.g., cameras, then the system could recognize all the evacuees and the related congestion level in different areas of the smart building as long as there is sufficient visibility within the building. We use CEP rules for updating the number of pedestrians in an area. The precision of this technology is related with the precision of the supporting people tracking technologies, in particular, iBeacons. To increase the tracking accuracy, further sensors are needed.

Our proposal addresses the computational complexity of managing the huge amount of data that can be continuously generated in a large installation. On the one hand, users' smartphones process events perceived from the infrastructure and forward only relevant high level events to the network of SB agents. Moreover, the decision of running the user agent on personal smartphones facilitates dealing with users' privacy concerns. To guarantee the efficiency of the evacuation, an important issue is the percentage of the evacuees that use the ERGA app on their smartphones during the evacuation. For best evacuation results, the evacuees should be familiar with smartphone navigation and should follow the ERGA recommendations.

In the future work, first we plan to perform a sensitivity analysis evaluating to which extent the results vary based on the level of accuracy in the iBeacon signals' reading; then, we plan to test our architecture in a simulated sufficiently complex large installation scenario where we will evaluate the correctness of CEP rules and the route recommendation algorithm in different settings. Then, for the proof of concept, we intend to deploy a field test in a University-like building.

Acknowledgments. This work has been partially supported by the Autonomous Region of Madrid through grant "MOSI-AGIL-CM" (P2013/ICE-3019) co-funded by EU Structural Funds FSE and FEDER, "SURF" (TIN2015-65515-C4-4-R (MINECO / FEDER)) funded by the Spanish Ministry of Economy and Competitiveness and FEDER funds, the Excellence Research Group GES2ME (Ref. 30VCPIGI05) co-funded by URJC and Santander Bank, and by an STSM Grant from COST Action IC1303: AAPELE.

References

1. Social modelling of ambient intelligence applied to large installations (mosi-agil), `https://www.gsi.dit.upm.es/mosi/`
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Recommender systems handbook, pp. 217–253. Springer (2011)
3. Alam, S.J., Geller, A.: Networks in agent-based social simulation. In: Agent-based models of geographical systems, pp. 199–216. Springer (2012)
4. Avery, W.H., Soo, J.: Emergency/disaster guidelines and procedures for employees. CCH Canadian Limited (2003)
5. Bessis, N., Dobre, C.: Big data and internet of things: a roadmap for smart environments. Springer (2014)
6. Bruns, R., Dunkel, J.: Towards pattern-based architectures for event processing systems. Software: Practice and Experience 44(11), 1395–1416 (2014)
7. Chen, P.H., Feng, F.: A fast flow control algorithm for real-time emergency evacuation in large indoor areas. Fire Safety Journal 44(5), 732–740 (2009)
8. Cho, H., Ji, J., Chen, Z., Park, H., Lee, W.: Measuring a distance between things with improved accuracy. Procedia Computer Science 52, 1083–1088 (2015)
9. Cocking, C., Drury, J., Reicher, S.: The psychology of crowd behaviour in emergency evacuations: Results from two interview studies and implications for the fire and rescue services. The Irish Journal of Psychology 30(1-2), 59–73 (2009)
10. Crociani, L., Vizzari, G., Bandini, S.: Conflicting tendencies in pedestrian wayfinding decisions: a multi-agent model encompassing proxemics and imitation. In: Bazzan, A., et al. (eds.) In Proc. of the 9th Int. Workshop on Agents in Traffic and Transportation (ATT 2016) co-located with IJCAI 2016. CEUR Workshop Proceedings, vol. 1678, pp. 1–8 (2016)
11. Cugola, G., Margara, A.: Processing flows of information: From data stream to complex event processing. ACM Computing Surveys (CSUR) 44(3), 15:1–15:62 (2012)
12. Deng, Z., Yu, Y., Yuan, X., Wan, N., Yang, L.: Situation and development tendency of indoor positioning. China Communications 10(3), 42–55 (2013)
13. Drury, J., Brown, R., González, R., Miranda, D.: Emergent social identity and observing social support predict social support provided by survivors in a disaster: solidarity in the 2010 chile earthquake. European Journal of Social Psychology 46(2), 209–223 (2016)
14. Drury, J., Novelli, D., Stott, C.: Psychological disaster myths in the perception and management of mass emergencies. Journal of Applied Social Psychology 43(11), 2259–2270 (2013)
15. Drury, J., Reicher, S.D.: Crowd control. Scientific American Mind 21(5), 58–65 (2010)
16. Dunkel, J., Fernández, A., Ortiz, R., Ossowski, S.: Event-driven architecture for decision support in traffic management systems. Expert Systems with Applications 38(6), 6530–6539 (2011)
17. Filippoupolitis, A., Gelenbe, E.: A distributed decision support system for building evacuation. In: 2009 2nd Conference on Human System Interactions. pp. 323–330. IEEE (2009)
18. Fisk, C.: Some developments in equilibrium traffic assignment. Transportation Research Part B: Methodological 14(3), 243–255 (1980)
19. Guest, J., Eaglin, T., Subramanian, K., Ribarsky, W.: Visual analysis of situationally aware building evacuations. In: Proc. of IS&T/SPIE electronic imaging. vol. 8654. SPIE (2013)
20. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407(6803), 487–490 (2000)

21. Hermoso, R., Dunkel, J., Krause, J.: Situation awareness for push-based recommendations in mobile devices. In: Abramowicz, W., et al. (eds.) Business Information Systems: 19th Int. Conf., BIS 2016, Proc. pp. 117–129. Springer International Publishing (2016)
22. Ijaz, F., Yang, H.K., Ahmad, A.W., Lee, C.: Indoor positioning: A review of indoor ultrasonic positioning systems. In: Advanced Communication Technology (ICACT), 2013 15th International Conference on. pp. 1146–1150. IEEE (2013)
23. Incel, O.D., Kose, M., Ersoy, C.: A review and taxonomy of activity recognition on mobile phones. BioNanoScience 3(2), 145–171 (2013)
24. Jahn, O., Möhring, R.H., Schulz, A.S., Stier-Moses, N.E.: System-optimal routing of traffic flows with user constraints in networks with congestion. Operations Research 53(4), 600–616 (2005)
25. Kashevnik, A., Shchekotov, M.: Comparative analysis of indoor positioning systems based on communications supported by smartphones. In: Proc. FRUCT Conf. pp. 43–48 (2012)
26. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: A review of the state-of-the-art. Information Fusion 14(1), 28–44 (2013)
27. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter 12(2), 74–82 (2011)
28. Liu, J., Chen, R., Chen, Y., Pei, L., Chen, L.: iparking: An intelligent indoor location-based smartphone parking service. Sensors 12(11), 14612–14629 (2012)
29. Luckham, D.C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Pub. Co., Inc., Boston, USA (2001)
30. Lujak, M., Giordani, S., Ossowski, S.: Fair route guidance: Bridging system and user optimization. In: 17th Int. IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 1415–1422 (Oct 2014)
31. Lujak, M., Billhardt, H., Ossowski, S.: Optimizing emergency medical assistance coordination in after-hours urgent surgery patients. In: Bulling, N. (ed.) Multi-Agent Systems: 12th European Conference, EUMAS 2014, Prague, Czech Republic, Revised Selected Papers. Springer International Publishing (2015)
32. Lujak, M., Billhardt, H., Ossowski, S.: Distributed coordination of emergency medical service for angioplasty patients. Annals of Mathematics and Artificial Intelligence 78(1), 73–100 (2016)
33. Lujak, M., Giordani, S., Ossowski, S.: Route guidance: Bridging system and user optimization in traffic assignment. Neurocomputing 151, 449–460 (2015)
34. Lujak, M., Giordani, S., Ossowski, S.: On avoiding panic by pedestrian route recommendation in smart spaces. In: 2016 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom). pp. 1–5 (June 2016)
35. Lujak, M., Ossowski, S.: Intelligent people flow coordination in smart spaces. In: Rovatsos, M., Vouros, G., Julian, V. (eds.) Multi-Agent Systems and Agreement Technologies: 13th European Conference, EUMAS 2015, and Third International Conference, AT 2015, Athens, Greece, December 17-18, 2015, Revised Selected Papers. Springer International Publishing (2016)
36. Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R.R., Handziski, V., Sen, S.: A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In: Proceedings of the 14th Int. Conf. on Information Processing in Sensor Networks. pp. 178–189. ACM (2015)
37. Manual, T.A.: Bureau of public roads. US Department of Commerce (1964)
38. Mawson, A.R.: Understanding mass panic and other collective responses to threat and disaster. Psychiatry 68(2), 95–113 (2005)
39. Moussaïd, M., Kapadia, M., Thrash, T., Sumner, R.W., Gross, M., Helbing, D., Hölscher, C.: Crowd behaviour during high-stress evacuations in an immersive virtual environment. Journal of The Royal Society Interface 13(122), 20160414 (2016)
40. Ng, T.M.: From “where I am” to “here I am”: accuracy study on location-based services with IBeacon technology. HKIE Transactions 22(1), 23–31 (2015)

41. Ossowski, S., Sierra, C., Botti, V.: Agreement technologies: a computing perspective. In: Ossowski, S. (ed.) Agreement Technologies, pp. 3–16. Springer Netherlands, Dordrecht (2013)
42. Patterson, O., Weil, F., Patel, K.: The role of community in disaster response: conceptual models. Population Research and Policy Review 29(2), 127–141 (2010)
43. Ronchi, E., Nilsson, D., Modig, H., Walter, A.L.: Variable message signs for road tunnel emergency evacuations. Applied ergonomics 52, 253–264 (2016)
44. Sadri, F.: Ambient intelligence: A survey. ACM Computing Surveys (CSUR) 43(4), 36 (2011)
45. Schneeweiss, C.: Distributed decision making. Springer Science & Business Media (2012)
46. Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J.: A survey of online activity recognition using mobile phones. Sensors 15(1), 2059–2085 (2015)
47. Suryotrisongko, H., Ishida, Y.: Emergence of cooperation as the impact of evacuee's solidarity. In: 2011 IEEE Int. Symp. on Safety, Security, and Rescue Robotics. pp. 265–271. IEEE (2011)
48. Susi, M., Renaudin, V., Lachapelle, G.: Motion mode recognition and step detection algorithms for mobile phone users. Sensors 13(2), 1539–1562 (2013)
49. Teymourian, K., Paschke, A.: Enabling knowledge-based complex event processing. In: Proceedings of the 2010 EDBT/ICDT Workshops. pp. 37:1–37:7. ACM, New York, NY, USA
50. Tubbs, J., Meacham, B.: Egress design solutions: A guide to evacuation and crowd management planning. John Wiley & Sons (2007)
51. Ustev, Y.E., Durmaz Incel, O., Ersoy, C.: User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal. In: Proc. of the 2013 ACM Conf. on Pervasive and Ubiquitous Computing Adjunct Publication. pp. 1427–1436. UbiComp '13 Adjunct, ACM (2013)
52. Varga, L.Z.: On intention-propagation-based prediction in autonomously self-adapting navigation. vol. 16, pp. 221–232 (2015)
53. Wada, T., Takahashi, T.: Evacuation guidance system using everyday use smartphones. In: Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on. pp. 860–864. IEEE (2013)
54. Wernz, C., Deshmukh, A.: Multiscale decision-making: Bridging organizational scales in systems with distributed decision-makers. European Journal of Operational Research 202(3), 828–840 (2010)
55. Yen, J.Y.: Finding the k shortest loopless paths in a network. Management Science 17(11), 712–716 (1971)
56. Yim, J.: Design of the recommendation module for context aware vod museum guide android app. International Journal of Software Engineering and Its Applications 7(2), 273–285 (2013)
57. Zervas, E., Mpimpoudis, A., Anagnostopoulos, C., Sekkas, O., Hadjiefthymiades, S.: Multi-sensor data fusion for fire detection. Information Fusion 12(3), 150–159 (2011)

Marin Lujak is an Associate Professor at IMT Lille Douai. He graduated Mech. Eng. at the University of Zagreb in 2006 and received a PhD degree in Management Eng. in 2010 from the University of Rome "Tor Vergata", Italy. From 2011 to 2016, he was a post-doc researcher at the University Rey Juan Carlos in Madrid, Spain. He has written more than 40 journal, workshop, and conference papers on the topic of multi-agent coordination and distributed multi-agent optimization in various industrial environments. He has participated in more than 10 research projects and has been a member of the organizing and programme committees of multiple international workshops and conferences.

Holger Billhardt received his PhD in Informatics from the Technical University of Madrid in 2003. Since 2001, he works at the University Rey Juan Carlos in Madrid in the Artificial Intelligence Group, where he is currently an Associate Professor of Computer Science. His recent research has been concerned with multi-agent systems, in particular with enforcement mechanisms for regulating the behaviour of software agents in open and dynamic environments. Prof. Billhardt is author or coauthor of more than 80 research papers and has participated in 19 nationally or internationally funded research projects and contracts. He has also been member of the organizing and programme committee of several international workshops and conferences.

Jürgen Dunkel is a professor at the Hanover University of Applied Sciences. His areas of research include software engineering and software architecture as well as Internet technologies. He focuses on agent-based systems and e-learning and conducts research on topics such as semantic sensor networks and complex event processing and mobile reading devices for the localization of passive UHF transponders. Jürgen Dunkel is a member of various committees, including the European Conference on Web Services (ECOWS) and the International Conference on Computer and Information Sciences (ICCIS).

Alberto Fernández received the PhD degree from the University Rey Juan Carlos in 2007. He is an Associate Professor at the University Rey Juan Carlos in Madrid (Spain), where he is member of the Artificial Intelligence Group and the Centre for Intelligent Information Technologies (CETINIA). During the last years, he has worked on the fields of coordination in multi-agent systems, semantic Web, and self-adaptive systems. He has applied his research to real world problems, in particular to transportation management. He has published more than 80 research papers in international journals and conferences, and has participated in more than 20 research projects. He has been member of the organizing committee and the programme committee of several international workshops and conferences.

Ramón Hermoso is an Assistant Professor at the University of Zaragoza (Spain). Formerly, he worked as a Senior Research Officer at the University of Essex (UK) and as an Assistant Professor at the University Rey Juan Carlos in Madrid (Spain). He obtained his PhD in 2011 and his research interests span from trust and reputation mechanisms in multiagent systems to mechanism design for complex systems and social networks. He is an author of several publications in journals, books and international conferences, and he have participated in more than 15 research projects, funded by both national and international institutions.

Sascha Ossowski is a Full Professor of Computer Science and the Director of the Centre for Intelligent Information Technologies (CETINIA) at University Rey Juan Carlos in Madrid. Formerly, he was an HCM/TMR research fellow at the AI Department of Technical University of Madrid. He obtained his MSc degree in Informatics from the University of Oldenburg (Germany) in 1993, and a PhD degree in Artificial Intelligence from UPM in 1997. He has authored more than 200 research papers, focusing on the application of Artificial Intelligence techniques to real world problems in various domains. He is co-editor of more than 20 books, proceedings, and special issues of international journals.

Received: October 14, 2016; Accepted: January 10, 2017.