

An Architecture for Situation-aware Evacuation Guidance in Smart Buildings

Holger Billhardt¹, Jürgen Dunkel², Marin Lujak¹, Alberto Fernández¹, Ramón Hermoso³, and Sascha Ossowski¹

Abstract. Smart Cities require reliable means for managing installations that offer essential services to the citizens. In this paper we focus on the problem of evacuation of smart buildings in case of emergencies. In particular, we present an abstract architecture for situation-aware evacuation guidance systems in smart buildings, describe its key modules in detail, and provide some concrete examples of its structure and dynamics.

1 INTRODUCTION

As cities in the 21st century are growing both in size and population, it is necessary to have reliable means to manage installations that offer essential services to the citizens (e.g., airports, train stations, sports centres, museums, and so on). Although there are already experts who design and manage such facilities, there is a lack of operational tools and knowledge to explore their functional limitations in a principled manner, to identify potentially dangerous situations (a crisis is always identified when it is too late), and to support decision-making in case of emergencies.

Recommendations or guidelines about what to consider and how to react do exist, but they can hardly be challenged or debated upon as they are often based on specific cases and experiences rather than strong general arguments. In practice, frequently it is up to human decision-makers to design and monitor an appropriate and timely course of action in response to a specific emergency.

Recently, it was proposed that, by bringing together works from the fields of Agent-Based Social Simulation (ABSS), Ambient Intelligence (AmI), and Agreement Technologies (AT), advanced methods and tools can be developed to address the aforementioned problem [1]. In particular, it has been suggested to use ABSS as a means for realistically modelling human crowds in large installations (taking into account both individual and herd behaviours, as well as their interplay); AmI techniques are adequate to model and simulate physical devices in smart spaces that capture relevant features of the situation (sensors) and provide decision-makers with the means to act upon it (actuators); while AT are used to explore intelligent strategies for managing such advanced installations as large-scale open distributed social systems.

In this paper, we focus on the problem of evacuation of installations of the aforementioned type in case of emergencies. In particular, we focus on smart buildings equipped with information processing, sensing and actuation facilities. In [2], for instance, a recommender system has been put forward that arranges personalized visits through a museum, 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 that take the current Context-aware Recommender Systems (CARS) that are discussed in detail in [3].

The present work aims at exploiting infrastructures of this type also for evacuation purposes.

The objective of an evacuation is to relocate evacuees from hazardous to safe areas or the areas where the life-threatening risk is minimal while providing them with safe routes. Present building evacuation approaches are mostly static and preassigned. Frequently, no coordination is available except for predefined evacuation maps. Still, due to the lack of the overall evacuation network information, there might be casualties caused by a too slow evacuation on hazardous routes. *Real-time* route guidance systems, which dynamically determine evacuation routes in inner spaces based on the imminent or ongoing emergency, can help reducing those risks. A dynamic, *context-sensitive* notion of route *safety* is a key factor for such recommendations, in particular as herding and stampeding behaviours may occur at potential bottlenecks depending, among other factors, on the amount of people who intend to pass through them. Furthermore, smart devices allow guidance to be *personalized*, taking into account, for instance, the specific circumstance of the elderly, disabled persons, or families. In such settings, an adequate notion of *fairness* of evacuation route recommendations is of utmost importance to assure the trustworthiness of the system from the standpoint of its users [4]: the guidance should not only achieve good overall performance of the evacuation process, but must also generate proposals for each of its users that each of them perceive as efficient. Finally, large groups of people may need to be evacuated so *scalability* plays a key role.

Therefore, we concentrate on real-time situation-aware evacuation guidance in smart buildings such that we keep track of the related fairness considerations among the paths assigned to individuals based on their mobility limitations, initial positions, respecting individual's privacy, and other evacuation requirements.

Section 2 describes in detail the particular problem that we are addressing, extracts requirements for the architecture, and provides a brief overview of the devices, methods and tools, mainly from the fields of AmI and AT, that we will use to address them. Section 3

¹ CETINIA, University Rey Juan Carlos, Spain, email: {holger.billhardt, alberto.fernandez, marin.lujak, sascha.ossowski}@urjc.es

² Computer Science department, Hochschule Hannover, Germany, email: juergen.dunkel@hs-hannover.de

³ Computer Science department, University of Zaragoza, Spain, email: rhermoso@unizar.es

outlines our abstract architecture, describes the structure and dynamics of its key modules in further detail, and provides some concrete examples to this respect. We conclude the paper with Section 4, describing lessons learnt and future lines of work.

2 EVACUATION GUIDANCE IN EMERGENCY SITUATIONS

A pedestrian route recommender system for smart spaces that recommends the safest routes to pedestrians and simultaneously optimizes conflicting objectives of finding the social optimum and minimizing individual path travel times in steady state conditions while considering people flow and fairness was presented in [4].

The system considers the influence of stress on human reactions to the recommended routes and iteratively ponders user 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 was studied in [5]. Here, Lujak et al. model self-concerned and social group behaviour via individual and team reasoning. The recommended routes take in consideration the affiliate ties to guarantee evacuee's compliance with the routes.

If real-time infrastructure information is available to evacuees and they can negotiate their routes, it becomes possible to provide a selection of safe fair routes considering individual safety requirements. Therefore, we assume that the building and evacuees are monitored by a strategically positioned network of sensors. The monitoring permits us both to recognize the evacuees' behavior in respect to the suggested route and time window as to perceive the congestion and safety conditions of the infrastructure. Furthermore, we assume that the people flow demand (i.e., evacuation requests) is known at the beginning of the time window of evacuation. This can be achieved based on the number of persons detected by the sensor network in the building.

The aim of the architecture is, thus, to safely evacuate all the evacuees' demand on (temporally) efficient and safe routes and if not possible, then evacuate as many people as possible within the allotted time period. To this aim, we should find optimal paths toward safe exits that maximize evacuees' safety and minimize their evacuation cost considering critical crowd density and flow and thus avoiding the crowdedness conditions that might result in panic. The path cost can reflect different factors, such as the evacuation time or cost incurred because an evacuee is too close to a hazard (e.g., fire, smoke).

In the case of contingencies, the architecture should reroute evacuees towards safe exits and, thus, propose evacuation routes that are adaptive to unpredictable safety drops in the evacuation network.

As a continuation of the works mentioned previously that mathematically model the safe evacuation problem and propose a scalable and robust optimization method applicable in real world, in this paper we propose an architecture that uses necessary sensory, localization, semantics, and processing technologies that can provide real time situation awareness and evacuee guidance based on individual requirements.

2.1 Technologies

2.1.1 Indoor location infrastructure

A. Localization with landmarks

A prerequisite for intelligent routing guidance is a detailed knowledge about the current localization of all persons in the building: First, the routing algorithm must know about the occupancy of each space in a building for calculating an appropriate route. Secondly, the precise position of each person is necessary for providing her with individualized routing recommendations taking her specific constraints into account.

There are various technological approaches to localize persons in buildings:

- **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 doesn't yield good accuracy: the distance between a mobile phone 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** technology has recently been introduced to support indoor navigation [6]. An iBeacon device uses Bluetooth LE 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 location technologies

	#Sender	#Reader	Accuracy
<i>WiFi</i>	few senders per floor	1 reader per person	low
<i>RFID</i>	1 sender per person	1 reader per room	medium
<i>Beacon</i>	many senders per room	1 reader per person	high

Table 1 summarizes the characteristics of the different technologies that are applicable for indoor localization. It states the superior accuracy of iBeacon technology: there are as many readers as users, and each building section can be equipped with so many

beacons that a dense net of landmarks is given. Furthermore, some of our former projects proved that iBeacons provide sufficient localization accuracy [7][8]. Therefore, we applied beacon technology in our scenario, i.e. all sections of the buildings contain a sufficient number of iBeacons that cover completely the space in the building.

B. User smartphones:

The personal smartphones of the users play two different roles: they serve as readers of the iBeacon signals and they can exploit their built-in sensors to derive more details about the current situation of its particular user.

- *Beacon reader for localization:* In smartphone operating systems such as iOS and Android, the capability of reading iBeacon signals is already integrated. In ranging mode, a smartphone estimates the proximity to an iBeacon according to the three proximity ranges:
 - IMMEDIATE: [0, 0.5m]
 - NEAR: [0.5m, 2m]
 - FAR: > 2m
 Each room is equipped with several iBeacons with non-overlapping ranges. As soon as a user approaches an iBeacon within the predefined range (e.g. NEAR) the smartphone triggers an event carrying the iBeacon ID. Then the smartphone knows that it is near that iBeacon and can forward this information to a server that coordinates emergency situations. An iBeacon ID is hierarchical 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 several works on how to use phone-based sensors for performing activity recognition. For instance, the authors in [9] applied different machine learning techniques, such as decision trees, logistic regression and neural networks to classify accelerometer data as certain activities. In our scenario, the current behavior of the users is crucial to detect panic situations, e.g. the situation that most persons in a room are running.

Furthermore, the smartphones serve as an individualized communication channel to each user to provide personalized routing guidance.

C. Further Sensors and Infrastructure

Further sensors are necessary for achieving situation awareness in the emergency recommender system. In particular, these sensors can be used to detect unexpected events in the building. 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 server also provides 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.

Furthermore, building operators can specify current incidents that could be detected automatically.

2.1.2 Complex Event Processing (CEP)

A key issue 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 sections 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 build-up smoke, or panic related herding and stampeding behaviors.

Apparently, such situational knowledge cannot be predefined, but must be inferred by exploiting live data. Usually, live-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 dynamic knowledge with a high change frequency. In emergencies, these streams of events must be evaluated in real-time to achieve situation awareness.

Considering a solitary event is usually of no significance, because it represents just a single incident in the physical world. For instance, it is of no importance if a single person is running, but if all persons in a room are running it could indicate a panic situation.

Complex event processing (CEP) is a software technology to extract the information value from event streams [10], [11]. CEP analyses continuous streams of incoming events in order to identify the presence of complex sequences of events, so called event patterns. The main goal of CEP is to extract a domain-specific meaning out of the observed streams of simple fine-grained and uncorrelated events. Instead, according to the key idea of CEP, a set of fine-grained simple events must be correlated to a single complex event with a significant meaning [10]. For instance, a panic event can be inferred, if the smartphones of nearly all visitors in 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:

Operators	
$\wedge, \vee$	Boolean operator for events or constraints.
NOT	Negation of a constraint
->	Sequence of events (e1 -> e2 meaning e1 occurred before e2).
Timer	Timer(time) defines a time to wait Timer.at(daytime) is a specific (optionally periodic) point of time
.within	defines a 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) [10]. 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. Event processing agents communicate with each other by exchanging events.

3 IN-DOOR EMERGENCY MANAGEMENT ARCHITECTURE

In this section we present an abstract architecture and describe the different components comprising it. Then, we give some details and examples of the CEP and Route recommender modules.

3.1 Abstract Architecture

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

The general operation dynamics of the system is based on two modes: standard mode and evacuation mode. In standard mode, the system continuously monitors the current state of the building, trying to detect a possible emergency scenario. If such a situation is

detected (e.g., an emergency event is detected through complex event processing), the system alerts some human operator who can activate an evacuation process and the system enters in evacuation mode. In this mode, the situation of the building is still monitored and an evacuation route recommendation algorithm is executed, which provides individualized route guidance to the people that are currently in the building.

The system consists of two main parts: User Agents (UA) and Emergency Manager (EM), as well as a set of Sensors that are located at different points in the infrastructure.

User Agent (UA)

The user agent manages and stores all the information that is related to a particular user (a person that is currently located in the building under consideration). The UA is executed as an app on the smartphone of each user. Here, we assume that people that enter the building have either downloaded and run such an app on their smartphones, or they have been provided with some Smartphone like device that runs the app when they entered the building.

The UA contains three parts: a preference module, a user situation awareness module and a recommendation interface. The *preferences and constraints module* allows the user to specify certain preferences or constraints regarding evacuation scenarios; e.g. certain handicaps that imply to a restricted mobility of the person (wheelchair, blind, etc.). This information is entered during the configuration of the UA and is stored locally in form of RDF² data. RDF is a standard data model for knowledge representation commonly used on the semantic Web.

The *user situation awareness module* exploits sensor data (from the smart phone and beacons installed in the building) and reasons about the behaviour and location of the user (through local CEP processes). This derived information is passed to the situation module in the EM. In order to assure privacy, the amount of information provided to external components is different in standard and in evacuation mode. In standard mode, only certain basic data about the user's situation are forwarded to the EM (e.g., location, running events). In case of the activation of an evacuation

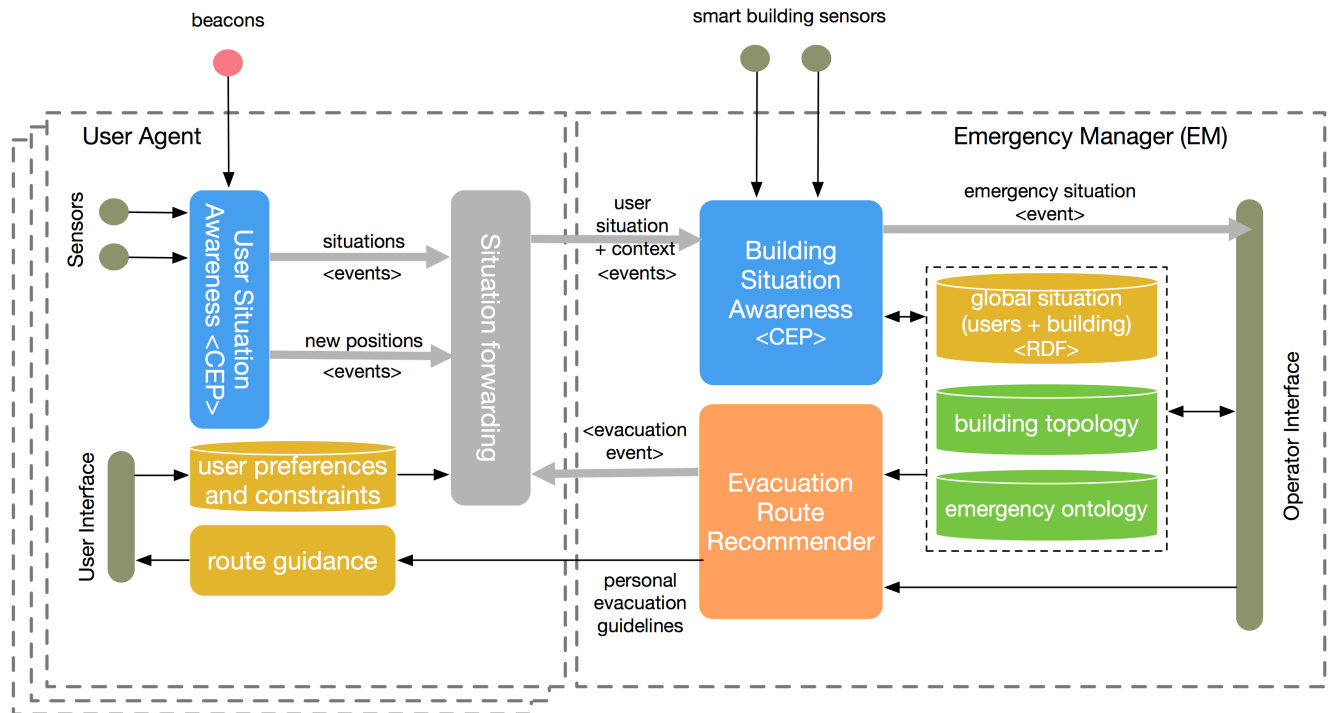


Figure 1. Overall architecture of the evacuation guidance system

(e.g., the EM broadcasts an evacuation event to all user agents), more detailed events are detected and also the preferences and constraints regarding user mobility are passed to the EM. That is, we consider that an emergency situation prevails upon privacy issues.

Finally, the evacuation mode will also trigger the recommendation interface. This interface provides the user with personalized navigation guidelines for evacuation, helping her to leave the building in the way it was calculated for her by the evacuation route recommender.

Emergency Manager (EM)

The emergency manager is the central part of the system. A *building situation awareness* module combines and analyses the events provided from the individual user agents with data from smart building sensors and generates information about the global situation of the building. This information is stored in the data model as RDF data. In this process CEP is used to filter irrelevant information and to generate higher level events. Especially in the case of the user events, individual data is aggregated to detect events regarding groups of users as well as identifying the density of the distribution of users in the building.

When the building situation awareness module detects an emergency situation, an alert is sent to the operator interface. This interface allows, on one hand, to monitor the situation of the building and, on the other hand, to trigger an evacuation process and to execute control actions in such a process (e.g., specifying blockage of parts of the building). If an evacuation process is initiated, the system enters evacuation mode and the *evacuation route recommender* [4] is executed. The module sends an evacuation event to all user agents informing them about the situation. Then it starts to calculate individual evacuation routes for all users. In this process, the algorithm uses three types of data:

- Data regarding the *building topology*: Static information about physical elements in a building (e.g. rooms, corridors, floors, doors, etc.) and relation among them (e.g. room A is 10 m², is next to room B and both are in floor F). In general, we use the term section to refer to physical elements. Topology knowledge is represented in such a way that is sufficient to describe the building network by a digraph with weights and tags on the constituent nodes and connecting edges. A node refers to some physical area (e.g., a room, a hall, a segment of a large corridor or floor, or some other open space). An edge connects two adjacent nodes and, thus, represents a way to move from one node to another. An edge represents, e.g., a passage, walkway, corridor, staircase, and alike. Nodes and edges are described through their type, surface, area, inclination, etc.
- *Emergency ontology*: This static 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, The influence of certain events like fire and smoke on the security level of an edge or node for evacuation purposes, etc.
- *Global situation*: Contains the current situation of the building itself as well as regarding the people that are currently in the building. This information includes:

- The distribution of people in the building (e.g., number of persons in each node and edge)
- Momentary positions, evacuation preferences, and mobility constraints of each person.
- Information on nodes and edges that are blocked for evacuation, and the reason for blockage. Possible reasons are fire, smoke and panic (that can be detected through the situation awareness module) and others (as specified by an operator).

During evacuation, the global situation of the building is dynamically updated in order to reflect the situation in each moment. In the same way, the guidance algorithm controls continuously the viability of the current evacuation strategy. If changes occur (e.g., new events are detected) that may violate that viability, then the evacuation route recommender recalculates new guidance data for each user.

In the following two subsections we describe in more detail the CEP component deployed in the user and building situation modules, and the principal functioning of the evacuation guidance algorithm.

3.2 CEP Components

Both agent types, User Agent (UA) and Emergency Manager (EM) analyse the incoming streams of events to understand the current situation. In this subsection, we will discuss in some detail the underlying event models and give some examples for appropriate rules for achieving situation awareness. To make the description more comprehensive, we will simplify the event model and the corresponding rules.

3.2.1 CEP in the User Agent

The UA 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 the UA monitors two types of explicit (or atomic) events to achieve this type of situation awareness:

- `beaconEvent(beaconID)`: an iBeacon with a certain ID³ has been detected
- `accelerationEvent(velocity)`: the phone is moving with a certain velocity

(i) The `beaconEvents` collected by a particular phone are used to derive the current position of its owner. The following CEP rule creates `enteringSection` and `leavingSection` events, meaning that the user is entering, respectively leaving a 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 b1 → beaconEvent AS b2
           ^ b1.id <> b2.id
ACTION:    CREATE enteringSection(userID, b2)
           CREATE leavingSection(userID, b1)
```

³ Note that the `beaconID` is structured and includes, among other information, the ID of a certain section or room.

The rule describes the situation that a new `beaconEvent` `b2` has been read in the phone, where the beacon ID has changed. (Here the beacon ID, more precisely its minor ID, corresponds with a section of a building)

(ii) Detecting a running user is another situation that must be forwarded to the Emergency Manager, 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 5 seconds:

```
CONDITION accelerationEvent As a[win:time:10sec]
    ^ average(a.velocity) > 5 km/h
ACTION      create runningEvent(userID)
```

If the condition matches, then the rule creates a `runningEvent` that contains the ID of the corresponding user.

3.2.2 CEP in the Emergency Manager

The CEP component in the Emergency Manager is responsible for deriving the global situation in the building. For instance, it could receive and analyze the following *atomic events*: produced by the CEP rules running on the users' smartphones.

- `enteringSection(userID, sec)`: a user with a certain ID has entered section `sec`.
- `leavingSection(userID, sec)`: a user with a certain ID has left section `sec`.
- `runningEvent(userID)`: a user with a certain ID is running.

Another kind of situational knowledge describes the *global* situation. A first type of rules is calculating the occupancy of different sections in the building. This data is used as input for a situation-aware routing recommendation algorithm.

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

```
CONDITION:
    (enteringSection AS e ∨ leavingSection AS l)
        [win:batch:15min]group_by(e.sec)
        ^ e.sec = l.sec
        ^ count(e) AS entries
        ^ count(l) AS exits
ACTION CREATE occupancy(e.sec, entries - exits)
```

The second type of rules tries to infer a global behavior of the people currently staying 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]
    group_by(r.sec)
    ^ count(r) > r.sec.occupancy * 0.2
ACTION: CREATE panicEvent(r.sec)
```

It groups all `runningEvent` according to a time-spatial window. The grouping criterion is defined by the section, where the `runningEvent` have occurred, and a time interval of 1 minute. If more than 20% of the people staying in the room are running, a panic situation is indicated.

Note that also other situation 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 sensor 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.

3.3 Evacuation Route Recommender Model

An evacuation route recommender model was presented in [4]. For the self-completeness of this work, we describe it briefly in the following. The model is made of the optimization and human factor module. Furthermore, the optimization module is made of the *Routes' safety optimization* component and the *Routes' travel time system optimization with fairness* component, Figure 2.

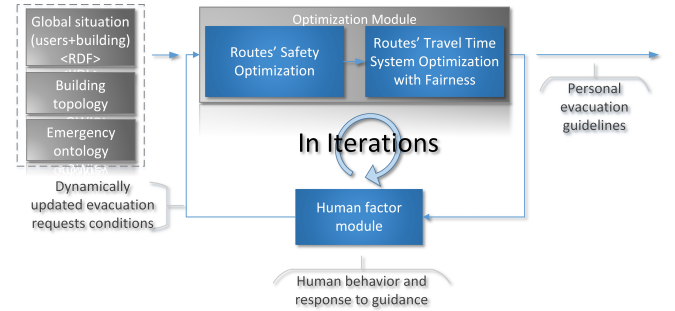


Figure 2. Evacuation route recommender model

Our objective is not only to find routes with satisfied minimal safety conditions since it may occur in hazardous situations that no such route exists. Thus, with the objective to increase the chances of survival, in the *routes' safety optimization*, we need to find routes that maximize Nash social welfare of the safety of the routes. We opt for this choice since it gives the best compromise between the optimization of the evacuees' utilitarian and egalitarian social welfare. Therefore, the safety optimization problem maximizing Nash product of the safeties of the constituent edges of evacuation paths is to be solved.

To facilitate scalability and robustness of the system in the evacuation of large premises, a distributed approach to this route safety optimization problem can be applied, as presented in [5].

Since we treat a highly computationally complex problem, the implementation of this distributed approach to our proposed architecture adds scalability by enabling the computation of the overall routing solution in parallel computation processes where each process is responsible of the computation of an evacuation route for a group of users with similar preferences and constraints in the same section of the building. The solution of the safety optimization model is a connected graph that assures the maximization of routes' safeties.

The basic idea of the module for the routes' travel time system optimization with fairness is as follows. The *route's travel time optimization with fairness* is divided into two layers. On the upper layer, Nash social welfare maximization problem with included envy-freeness and fairness constraints is decomposed to obtain a subproblem that can be optimized individually locally by the

processes described previously. The details on the optimization approach can be found in [12].

Moreover, based on the total demand expressed in terms of person flow per time unit, each process tries to achieve a sufficient number of shortest paths considering fairness for all its evacuees. The processes compute a sufficient number of shortest paths for their evacuees through, e.g., k-shortest path routing algorithm [13]. The prices of networks' edges are adjusted based on the overall processes' demand on the routes influencing congestion on the highly demanded arcs.

The prices are Lagrange multipliers that are calculated through a distributed dual-decomposition of the primal evacuation problem. On the other hand, each process calculates shortest paths to the set of safe exits with updated edges' prices, envy-freeness prices, consistency dual prices, and user demand distribution over routes' prices and thus decides upon the amount of users to be routed on each of the assigned routes.

After the route assignment is made for all evacuation requests on the first level of the optimization model, each process decides, on the second level, of its users' assignment to the routes assigned to it on the first level, based on relevant social welfare parameters that guarantee fairness of the assigned routes to its users through an iterative auction. While the negotiation for the assignment of the routes among different processes on the first level includes the communication among processes when they share the same arc(s), the negotiation through auctions on the second level is local between each process and its users and considers a fair assignment of the available routes based on the users' individual evacuation preferences and mobility constraints.

4 CONCLUSIONS

In this paper we have presented an abstract architecture for situation-aware evacuation guidance in smart building. The system provides an individual evacuation route recommendation to each user of a smart large installation. The proposal takes into account the current location and building state obtained through sensors and personal mobile devices, as well as human factors in emergencies.

We described the architecture and the main technologies proposed to implement it, namely, iBeacons and smartphones for obtaining live building information, CEP for efficiently event processing, and a distributed optimization algorithm for route recommendation.

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 emergency manager. On the other hand, we proposed a distributed evacuation route recommendation algorithm. Moreover, the decision of running the user agent on personal smartphones facilitates dealing with private information.

In the future we plan to test our architecture in a simulated scenario where we will evaluate the correctness of CEP rules and the route recommendation algorithm in different settings. Then, we will deploy a field test in a University building.

ACKNOWLEDGEMENTS

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) funded by the Spanish Ministry of Economy and Competitiveness, and through the Excellence Research Group GES2ME (Ref. 30VCP1G105) co-funded by URJC and Santander Bank.

REFERENCES

- [1] MOSI-AGIL project: <https://www.gsi.dit.upm.es/mosi/> (2016).
- [2] R. Hermoso, J. Dunkel, and J. Krause, 'Situation Awareness for Push-Based Recommendations in Mobile Devices', *19th Intern. Conf. on Business Information Systems (BIS) conference*, (2016).
- [3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, pages 217–253. Springer, 2011.
- [4] M. Lujak and S. Ossowski, 'Intelligent People Flow Coordination in Smart Spaces'. *Post-Proceedings of the 13th European Conf. on Multi-Agent Systems / 3rd Int. Conf. on Agreement Technologies*, Rovatsos M. et al. (Eds.), LNAI vol. 9571, 34–49, (2016).
- [5] M. Lujak, S. Giordani, S., and S. Ossowski, 'Distributed Safety Optimization in Evacuation of Large Smart Spaces'. *The 9th International Workshop on Agents in Traffic and Transportation (ATT 2016 @ IJCAI 2016)*, (2016).
- [6] Apple Inc., *iBeacons* <https://developer.apple.com/ibeacon/>
- [7] W. Zimmermann, *Indoor Navigation with iBeacon Technology*, Master thesis, Hannover University of Applied Sciences and Arts, 2016. [in German]
- [8] M. Brown, *Introducing iBeacon Technology at SXSW 2015*. <http://www.sxsw.com/news/2015/introducing-ibeacon-technology-sxsw-2015> (2015).
- [9] J.R. Kwapisz, G.M. Weiss, and S.A. Moore, '2010: Activity Recognition using Cell Phone Accelerometers', *ACM SIGKDD Explorations*, **12** (2), 74–82, (2010)
- [10] D. Luckham, *The Power of Events*. Addison-Wesley, 2002.
- [11] O. Etzion and P. Niblett P. *Event Processing in Action*. Manning, 2010.
- [12] M. Lujak, S. Giordani, and S. Ossowski, 'Route guidance: Bridging System and User Optimization in Traffic Assignment', *Neurocomputing*, **151** (1), 449–460, (2015).
- [13] J.Y. Yen, 'Finding the k shortest loopless paths in a network', *Management Science*, **17** (11), 712–716, (1971).