Delft University of Technology

Dynamic Computation of Time-Varying Spatial Contexts

Horvath, Imre; Li, Yongzhe; Rusak, Zoltan; van der Vegte, Wilfred; Zhang, G

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Imre Horváth[1]
Fellow ASME
Industrial Design Engineering,
Delft University of Technology,
Landbergstraat 15,
Delft 2628CE, The Netherlands
e-mail: i.horvath@tudelft.nl

Yongzhe Li
Industrial Design Engineering,
Delft University of Technology,
Landbergstraat 15,
Delft 2628CE, The Netherlands
e-mail: y.li-8@tudelft.nl

Zoltán Rusák
Industrial Design Engineering,
Delft University of Technology,
Landbergstraat 15,
Delft 2628CE, The Netherlands
e-mail: z.rusak@tudelft.nl

Wilhelm Frederik van der Vegte
Industrial Design Engineering,
Delft University of Technology,
Landbergstraat 15,
Delft 2628CE, The Netherlands
e-mail: w.f.vandervegte@tudelft.nl

Guangjun Zhang
State Key Laboratory of
Advanced Welding and Joining,
Harbin Institute of Technology,
92 West Dazhi Street,
Harbin 150001, China
e-mail: zhanggj@hit.edu.cn

Dynamic Computation of Time-Varying Spatial Contexts

There are many real-life processes whose smart control requires processing context information. Though processing time-varied context information is addressed in the literature, domain-independent solutions for reasoning about time-varying process scenarios are scarce. This paper proposes a method for dynamic context computation concerning spatial and attributive information. Context is interpreted as a body of information dynamically created by a pattern of entities and relationships over a history of situations. Time is conceived as a causative force capable of changing situations and acting on people and objects. The invariant and variant spatial information is captured by a two-dimensional spatial feature representation matrix (SFR-matrix). The time-dependent changes in the context information are computed based on a dynamic context information (DCI) management hyper-matrix. This humble but powerful representation lends itself to a quasi-real time computing and is able to provide information about foreseeable happenings over multiple situations. Based on this, the reasoning mechanism proposed in this paper is able to provide informative instructions for users who needed to be informed in a dynamically changing situation. This paper uses the practical case of evacuation of a building in fire both as an explorative case for conceptualization of the functionality of the computational mechanism and as a demonstrative and testing application. Our intention is to use the dynamic context computation mechanism as a kernel component of a reasoning platform for informing cyber-physical systems (I-CPSs). Our future research will address the issue of context information management for multiple interrelated spaces. [DOI: 10.1115/1.4034034]

1 Introduction

1.1 Introducing the Problem and the Objectives. Cyber-physical systems (CPSs) deeply penetrate into real-life processes with the objective of carrying out complicated tasks under varying circumstances. In order to be able to do this, they need sensing, reasoning, and actuation capabilities. When CPSs work in dynamic environments and/or under strongly varying circumstances, their control and reasoning mechanisms cannot be based on static system models. Even dynamically updated application models cannot provide the required information on time. For this reason, two novel ideas have been considered in system engineering: (i) reasoning based on real-time collected data and emergent requirements and (ii) using control strategies, instead of model-based control, to adapt the system to dynamic situations or personalized user needs. These are the roots of our research.

In order to make proper decisions on the operations and services provided by CPSs, we need to consider the context of their operation and servicing. However, if the system works in a dynamic environment and/or under strongly varying circumstances, then processing context information and using it in decision-making are not trivial problems [1]. This is a typical challenge for so-called I-CPSs, which implement a sensing–reasoning–actuation operation scenario, but their actuation functions are dedicated to informing people or systems (Fig. 1). What actuation means in this branch of CPSs is (i) generating individualized messages, (ii) forwarding them to specific clients, and (iii) monitoring and collecting their reactions. These functions are all strongly dynamic context-dependent, where the term "dynamic" refers to procedural state changes, rather than to ontological ones.

I-CPSs make it necessary to develop context-dependent control strategies to adapt their sensing, reasoning, and informing behaviors to various dynamic contexts. For this reason, the objective of our research is to develop a generic theoretical framework and a
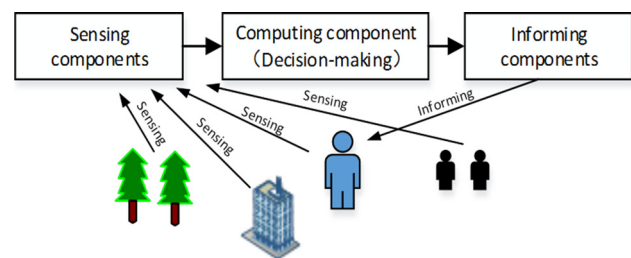

Fig. 1 Basic constituents of an informing CPS

computational mechanism for context data-driven control and adaptation strategy generation in run time. This reasoning mechanism will be the kernel of an embeddable computing platform that provides integral support for sensing, reasoning, and informing, and that can in turn be a standard central module of various application-specific I-CPSs. Though many results (including theories and computational approaches for dynamic context management) were presented, typically the current generation of I-CPSs cannot support a long-term monitoring and management of context changes and cannot develop operation control and adaption strategies run-time.

1.2 What is Proposed in This Paper? We propose a computational mechanism, which (i) collects context data from multiple sources, (ii) evaluates the actual situation of operation, (iii) develops a raw control strategy for problem solving, and (iv) refines the strategy by considering the influencing factors and foreseen changes. Context is usually referred to as any information related to people, places, or objects that is relevant for the operation of applications [2]. For our purpose, we considered context as a body of information dynamically created by a pattern of entities and relationships over a history of situations. Context data are aggregated by sensors and interpreted by the proposed reasoning mechanism, which also decides on what actions should be performed to achieve the application's objective [3]. Using DCI in the decision-making components of I-CPSs improves the "smartness" of a system as a whole. For instance, Lee et al. proposed a medical CPS, which is able to detect the physiological parameters and to use patient-related context information in decision support [4]. The highlights and the main objectives of our research can be summarized as follows:

- Making possible to use dynamically changing context in decision-making by application-specific smart CPSs.
- Handling dynamically changing spatial context information was in the focus of our research that also aimed at reducing the time of information input and processing.
- Development of a conceptual framework, which captures entities, relationships, attributes, and changes in space and time.
- Transferring the conceptual framework to a computational mechanism that can be used as a kernel of a reasoning platform for I-CPSs.
- Using evacuation of a building in fire as a practical case study throughout the completed research. It was used both as an explorative case for functional conceptualization of the computational mechanism and as a demonstrative and testing application.
- Generalizing the computational mechanism to be able to handle dynamically changing spatial context information in other application cases, such as protection in disaster, crowd management, and medical rehabilitation.

1.3 Structure of the Paper. In this paper, we concentrate on the reasoning mechanism proposed for dynamic spatial context computation. Consequently, we do not address the issues associated with conceptualization and implementation of the sensing, reasoning, and informing control platform, which nevertheless represents an integral part of our research and development work. Neither have we intended to discuss the issues of tailoring I-CPSs to specific applications. Section 2 gives a concise overview of the related research reported in the literature. Section 3 deals with the underpinning concepts and the elements of the computational mechanism proposed for dynamic context computation. Section 4 applies the proposed dynamic context computation mechanism into a real-life application context and validates its functionality and benefits through simulating a nonassisted and an assisted application scenario. Finally, in Sec. 5, we reflect on the progress

achieved so far and provide information about the planned follow up research.

2 Overview of the Related Literature

Though there was a library of seemingly relevant academic papers and digital documents available, a preview explored that most of them reported on results, which were not directly related to dynamic context computation for time-varying process scenarios. However, a deeper study of the related literature allowed us to identify three domains of interest, which provided useful information for our follow up research. These were (i) interpretation and representation of context information, (ii) DCI management and fast computation of dynamic contexts, and (iii) context-aware reasoning mechanisms and application systems. Below, we overview a representative subset of the related documents and expose the main findings.

2.1 Interpretation and Representation of Context Information. The term "context" is used variously in the literature, and it has been defined either too general or too specific, depending on the purpose [5]. There is no universally accepted definition of context. In its broadest interpretation, context is about any circumstance in which something happens. Technically, it refers to the setting of a thing or a process, i.e., the set of facts or circumstances that surround a situation or event. Context influences the outcome of a thing or a process that happens. In previous studies, context information has been defined as a set of information, which influences the realization of a certain objective with a particular manner. In the case of human beings, context is considered as a "state-of-the-mind" influencing interpretation and decision-making. In the case of smart (cognitive) systems, it is a model of the system that makes situated reasoning and decision-making possible. Context information has been defined by developers of context-aware systems as any information related to people, places, or objects that are relevant for operations of the systems [6]. Among others, Debes et al. tried to provide a scientifically defendable definition of term context [7]. They claimed that three classes of context can be distinguished: (i) user context, (ii) terminal context, and (iii) communication network context. Dey and Abowd classified context into three categories: (i) physical context, (ii) action-based context, and (iii) emotional context [8]. In a different paper, they defined context as any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves [9]. As descriptive parameters of context, features such as location, position, surrounding, environmental entity, identity, status, application, situation, and time, have been considered by many researchers [10].

Managing context data and information have multiple aspects, such as (i) data elicitation and ordering, (ii) construction of context models, (iii) using context information in reasoning about things and processes, and (iv) archiving context information. Strang and Linnhoff-Popien surveyed the state-of-the-art of context modeling [11]. Fernández-de-Alba et al. proposed a distributed blackboard framework for management and fusion of context information at different levels [12]. Malandrino et al. addressed the issues of aggregating context from distributed sources and deriving appropriate context-aware adaptation policies in the case of mobile users and ubiquitous web access and services [13]. Fuchs et al. proposed a metamodel to define context information and its associations based on concepts, such as devices, persons, and their properties and relationships [14]. In the practice, context information is typically processed, evaluated, or passed on by extensible markup language. Context information may be heterogeneous, incomplete, and imperfect [15]. Representation and handling context data having these features have been addressed in many research projects. For instance, Castro and Munz dealt with managing heterogeneous context data for smart spaces [16].

Henricksen and Indulska addressed the issues related to modeling and using imperfect context information [17], while van Kranenburg and Eertink dealt with the challenge of processing heterogeneous context information [18]. Development of context ontologies represents a specific strand of research. For instance, Gu et al. proposed an ontology-based context model to be used in intelligent environments [19]. Wang et al. proposed using OWL in ontology-based context modeling and reasoning [20].

2.2 DCI Management and Fast Computation of Dynamic Contexts. Changes in the objectives, environment, and circumstances of operation bring new entities and relationship into consideration, which result in changes of context information. Kokinov proposed a dynamic theory of context, which claims that context changes dynamically because of the inherent dynamics both of the memory-induced context (decreasing its influence with the course of time) and of the perception-induced context (continuously changing the perceived elements of the environment) [21]. This theory explains the internal context (formed by perception, memory, and reasoning in the human mind), rather than the dynamics of external context, which is established by the setting of and the interaction with the physical and social environment. Forsström and Kanter utilized ubiquitous sensors to enable capturing continuously evolving context information in mobile environments [22]. Euzenat et al. showed opportunities for using dynamic context management in pervasive applications [23], while Kim and Mahapatra reported on the application of dynamic context management for a low power coarse-grained reconfigurable architecture [24].

Brown interpreted context information as a set of elements of the user's environment that should be known by the computer [25]. Aktas et al. used dynamic metadata as context on context-aware systems [26]. Grossmann et al. proposed a method for efficient management of context information in the case of large-scale scenarios [27]. Taconet et al. developed a runtime model and a middleware for dynamic context management [28]. To support user-driven web integration in the personal web, Villegas et al. investigated the role of a dynamic context management infrastructure [29]. Jaroucheh et al. presented a product line-based dynamic context management approach for pervasive applications [30].

2.3 Context-Aware Reasoning Mechanisms and Application Systems. Our literature study gave us that theory and methodology development for dynamic context-aware reasoning mechanisms are still in its infancy and that, for this reason, only limited efforts have been made toward developing system and application-independent sensing, reasoning, and informing platforms for CPSs [31]. Nevertheless, there are important contributions. Kwon critically analyzed the potential roles of context-aware computing technology in optimization-based intelligent decision-making [32]. Brown et al. investigated how context-aware applications can make their way from the laboratory environments to the marketplace [33]. Anthony et al. developed a context-aware adaptation mechanism for the DYSCAS system, which facilitates the fusion (aggregation and interpretation) of the different pieces of data from changes in the context representation to trigger actions in certain actuators [34]. Dai and Xu discussed the application of context-aware computing for an assistive meeting system [35].

Baldauf et al. completed a focused survey of the recently developed context-aware application systems [36]. According to the interpretation of Vieira et al., context-aware application systems determine which user tasks are most relevant to a user in a particular context, considering history, preferences, behaviors, or conditions [37]. They presented an integrated approach to designing context-sensitive systems. Cortez discussed a CPS, which follows a disaster situation based on its situation awareness [38]. Various

problems and opportunities were addressed by Tanca et al. with regards to context-based personalization [3].

2.4 Main Findings. Computational processing of context information and using it in decision-making in highly dynamic processes are challenging tasks as shown by the recent publication. The core issue of dynamic context management is capturing the rapid, parallel changes of the surround information concerning mutually connected and/or isolated entities, which is needed for an optimal control of real-life processes. Considering their dynamically changing contexts, the high complexity of decision-making over processes makes it imperative to develop a proper framework for designers to be able to build decision-making components into I-CPSs. However, most of the found publications dealt with static or altering, rather than with dynamic and emergent context information (ECI). From a computational point of view, the studied context reasoning mechanisms have not been developed for handling high dynamics contexts and with the objective of achieving quasi-real time computation.

3 A Computational Mechanism for Dynamic Context Computation

3.1 Objectives of Capturing Dynamic Spatial Context Information. Our starting point was that context is a body of information dynamically created, maintained and/or/modified by a pattern of entities and their mutual relationships over a history of situations. The relationships of the entities are supposed to be of an each to others nature. It means that, for a particular entity, all the manageable pieces of information concerning other entities and their attributes and relationships form a context. In the case of a spatial arrangement of real-life entities, both the entities and their relationships can be metricized and expressed in measures, such as spatial locations, Euclidean distances, angular positions, and velocities. These can be used as attributes to characterize the spatial situation and relationships of entities. The question is how to capture, represent, and process these pieces of spatial information in a dynamically changing context? Obviously, not only the entities and their attributes may change in a dynamic situation but also the relationships and their attributes. We note that only on-site spatial information rather than geospatial information was considered in our research. The main requirements have been specified as follows.

The dynamic spatial context information processing mechanism should be:

- based on a computationally robust at the same time situation- and application-adaptive theoretical framework
- linearly computable, that is, linear increase of the entities and relations considered should not result in exponential increase in the context computation time
- able not only to describe observable situations but also to forecast changes and trends without the need for additional simulation tools
- able to receive data from a large number of (dynamically activated) sensors, evaluate the detected situation, and reason about how to achieve the specified objective
- coupled with a strategy development mechanism that converts the results of situation evaluation into action strategies for people on both individual and crowd levels
- connectable with a postprocessing mechanism that generates contents for the operation of an I-CPS

3.2 An Application Case as a Basis of Conceptualization of the Needed Mechanism. Many practical cases, which can be supported by dynamic spatial context management (e.g., assistive robotics-based home care, smart entertaining parks, smart failure detection, natural catastrophe management, etc.), have been considered in our research. From the set of possible application cases,
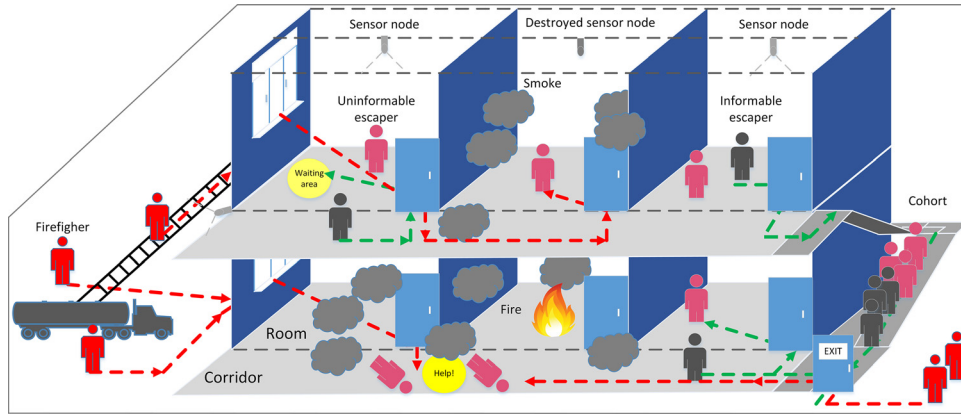
Fig. 2 Building in fire

we selected the case of building fire evacuation and used it as both an explorative case for conceptualization of the functionality of the computational mechanism and a demonstrative and testing application. Consider now the case of having fire within a building crowded with people. Figure 2 illustrates a possible real-life situation. The entities involved in this situation are: (i) fire, (ii) people, and (iii) exits of the building. The entity attributes are: (i) locations of the fire, (ii) locations of the people, (iii) locations of the exits, (iv) speed of moving of the people, (v) speed of motion of the fire, (vi) ability of a person to communicate digitally or verbally, and (vii) throughput capacities of the exits (number of people per second).

The relationships are: (i) remoteness of a person from the other people, (ii) remoteness of a person from the fire, (iii) remoteness of a person from the exits, (iv) remoteness of the fire from the exits, and (v) remoteness of an exit from the other exits. The relationship attributes are: (i) distances between a person and other people, (ii) distances between the people and the fire, (iii) distances between the people and the exits, (iv) distances between the fire and the exits, and (v) distances between an exit and other exits. For the sake of simplicity, let us assume that location information about the entities is collected by proper sensors placed in the building according to a considerate plan. The sensors are able to detect and communicate the data through a wireless sensor network (WSN) to a service computer at given points in time. The two research questions were: How to represent DCI concerning all the entities? and How to reason about the momentarily situations?

3.3 Structured Representation of Spatial Entities, Attributes, and Relations. As the conceptual framework of modeling dynamically changing contexts and data structuring, we considered an "attributes ← entities ↔ relationships → attributes" pattern. Based on this pattern, we conceptualized a two-dimensional SFR-matrix for a structured representation of spatial entities, attributes, and relations. The SFR-matrix is set out by rows and columns that arrange the entities in a structured manner. The principal arrangement of an SFR-matrix is shown in Fig. 3. It is a rectangular array of size $(s+1)(s+1)$, where s is the total number of entities captured, and $s \leq N$, and i and j are the running indices. The organization of the contents of the matrix is as follows: The first row, as well as the first column, contains the identifiers of the various entities included. The elements of the first column are generated by a transposition of the first row. The cells $(s \geq 2)$ in the main diagonal $(i = j)$ store all the entity-reflective information, while the cells outside the main diagonal $(i \neq j)$ store the connectivity information.

Entities are clustered as persons (P_p), exits (E_q), and fires (F_r). Within the matrix, they form blocks. The cells in the main diagonal capture the so-called "attribute profile" of the entity. For a person, this includes the profile attributes (ID and type), the sensed

attribute (location and status), and the derived attributes (speed of motion, target exit, estimated arrival time at the target exit, and danger coefficient). For an exit, it includes the entity ID and location, the sensed attributes (locked or not status and the number of people in jam), and the derived attributes (the number of people who selected the exit as a target, the number of people that can pass through per minute, and when the exit will be occupied by fire). Actually, these cells include multiple one-dimensional arrays that carry the data about the spatial features of the entities. As shown in Fig. 3, the upper triangular submatrix is divided into five blocks, which represent the distances DP-P, DP-E, DP-F, DE-E, and DE-F among the entities of different types. The contents of all the blocks are concurrently used in the dynamic context computation process.

In terms of the contents of the SFR-matrix, an invariable and a variable part can be differentiated. The invariable part is filled in as default of initial values (e.g., to describe the original arrangement of the space and the exits). Therefore, some preliminary computations are already done before fire is detected by field sensors, and the actual context computation mechanism is operationalized. Every exit is given an identity number, and the location of all the exits is recorded. Based on their location data, the momentarily distances (D) between the reference (central) points of any pairs of entities can be computed. The distances are expressed as scalar values in the relevant cells of the upper triangular submatrix, though vector quantities could also be considered. The cells in the lower triangular submatrix can be used for storing: (i) the distances calculated in a preceding situation and (ii) the distances

Fig. 3 SFR-matrix

calculated in a succeeding situation, or storing the distances forecasted over multiple steps. This means, all the relative distances can be easily computed and stored this way.

Entities may become involved in or excluded from a given situation. For instance, if an entity is not any more present in a given situation (e.g., a person successfully escaped, an exit became unusable, a fire has been extinguished, etc.), then it is labeled as inactive in the corresponding cell of the main diagonal and excluded from any further computation. Contrarily, if an entity becomes involved or reactivated in a given situation, its label is changed or set accordingly in the main diagonal.

In order to help reasoning about a sensible overall evacuation strategy (OES) (keep everyone safe during evacuation) and to handle the SFR-matrix conveyed information in the process of reasoning about the person's individual evacuation strategies, called personal escape routes (people must be informed to stay away from the fire at a safe distance and guided out), a priority list is constructed and used. The rationale of setting up the priority list is the momentarily distances between related actors (i.e., people, exits, and fire fronts). In addition to supporting OES generation, the priority list provides the highest opportunity of escaping for those, who are most threatened by the fire. Toward this end, it needs to be evaluated: (i) which exit will be occupied by the fire, (ii) with what estimated speed of motion, the fire is propagating, and (iii) at what time, the fire can be expected to arrive at the close-by exit. In these computations, the distances between the people and the fire, DP-F, and the distances between the fire(s) and the exits, DE-F, are used. The individual having the longest escape route will be put to the top of the priority list, in accordance with the formula used to calculate the danger coefficient.

3.4 Time-Related Management of Context Information.
Dynamic context management assumes that the situation is changing minute wise, which implies the need for considering time as an additional aspect of computation. Context information can theoretically be: (i) static (SCI), (ii) altering (ACI), (iii) dynamic (DCI), and (iv) emergent (ECI). In practice, they cannot always be separated so sharply. SCI can be captured in a robust information structures and has a limited influence on the change of the thing or process it is associated with. An example of it is an internal arrangement of a building. ACI typically changes or fluctuates over a substantial period of time and may have significant and lasting influence on the thing or process in question. This can be exemplified by the regular changes of the seasons. DCI changes at a rapid pace related to the rapid changes of a situation. An example of it is the traffic situation on a highway during rush hours. While these can be captured in predefined information structures or models, ECI is the most difficult to manage because of the unpredictable changes of the situation. Though an emerging context situation may have radical influence on reasoning, its forecasting is uncertain, as well as the information pattern, which describes it. We intended to develop a mechanism that is fully fledged for handling DCI and partially fledged for managing ECI.

As a logical basis of the DCI handling mechanism, we developed the concept of spatial feature representation hyper-matrix (SFRH-matrix). This is virtually a linear array of time-sequenced temporally interlaced SFR-matrices, as shown in Fig. 4. Computational processing of the hyper-matrix involves updating and recomputing the contents of the SFR-matrix at given Δt increments. The changes of the values in the corresponding cells of the temporally (or logically) subsequent SFR-matrices indicate certain trends of the changes (e.g., two persons are furthering from each other, or the fire is approaching a nearby exit). Based on these trends, the time-dependent content of the hyper-matrix can be used for forecasting. The information obtained by this makes it possible to discover incidental events that potentially happen in a sequence of situations. This affordance of the SFRH-matrix is exploited in the development of the overall and individual evacuation strategies.
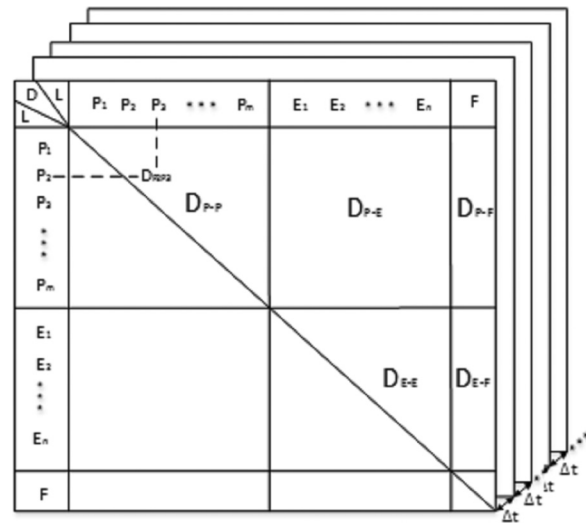

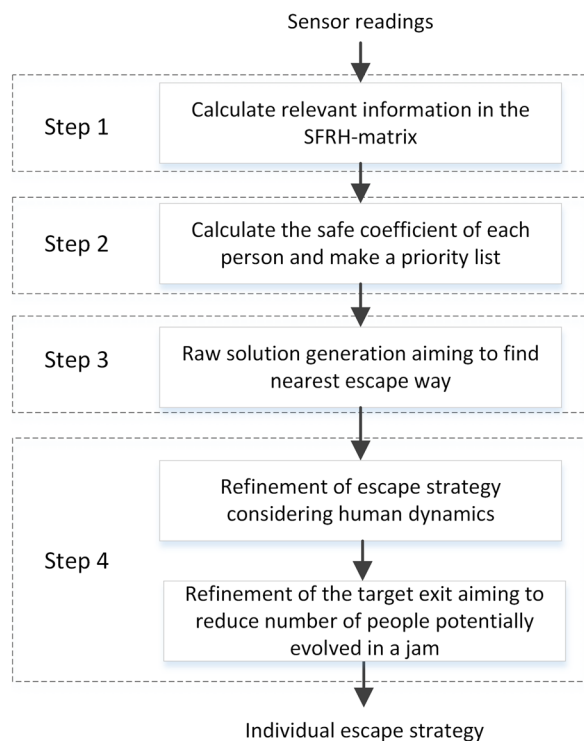**Fig. 4 Hyper-matrix for DCI management**

Time-shifted update and recomputation of the DP-E and DE-E distances in the SFR-matrices provide information about the progression achieved by operationalization a strategy. For instance, the number of persons walking toward the same exit can be deduced from the changes of DP-E distances at Δt_i and Δt_{i+1} points of time. If it raises the thread of jam forming or lack of throughput capacity, the strategy can be changed and certain people can be informed to change their route to another exit, if possible. Based on the changes of distances, the speed of motion can be computed for each person as well. With these, the total evacuation time and individual escape times can be computed. Furthermore, the time-constant DE-E distances can be used to decide on which exit should be targeted by which person. The changes of the DP-P distances over multiple SFR-matrices can be used to see if a person approaches another one in time, or moving away. This affordance is used to check if an informed person approaches a person-in-need according to the provided instruction, or if there is a threat of jam forming. If there is no change in the speed of moving in the subsequent SFR-matrices, then a person may have become injured or hindered by some incidental obstacle. The intensity of congestion forming (the number of people involved in a forming jam) can also be detected based on the DP-P distances. Determining the identifiers of the involved people, preventive measures can be applied to reduce the jam at reasoning about the persons–exits relationships. It can also be predicted when a jam will happen in time. Note that this is a computational requirement of lowest level priority.

Because the fire may spread in arbitrary directions in space, the boundary of its fire front should be taken into consideration in computation, rather than a factious central (reference) point. Information about the feature points of the boundary is to be provided by temperature sensors. The distances between the fire front and other entities are taken as the minimum distance between the concerned entity and the nearest feature point of the fire front. No matter if the fire front is represented by an open chain of points or by a close loop of points, the location and distance information can be used in two ways to provide information about: (i) if the fire front is growing or shrinking and (ii) if the fire front approaches a given location in the space. Even the size of a particular SFR-matrix can be changed in time and managed dynamically. The size of the matrix can be increased or decreased by inserting or deleting rows and columns, respectively, any time, before it is recomputed. The identifier of the concerned new entity/entities (person, exit, and fire) is added to the first row/column if the entity is a newly detected one, or removed, if not existing any more. These affordances help increase adaptability and optimize computational time. In order to facilitate correct registration

and decision-making, a classification action takes place whenever a new person is detected to get involved in the real-life situation. For the sake of procedural simplicity, presumed is that the information about possessing an operational mobile phone is collected from the people at entering the building.

3.5 Generating Evacuation Strategies and Escape Routes Based on the SFRH-Matrix. Four principles have been set up for the design of the reasoning mechanism: (i) keep everyone away from fire, (ii) evacuate all the people in the shortest possible time, (iii) reduce the number of people involved in a jam, and (iv) let each person escape in the shortest route. These principles have different levels of priority in the order of mentioning. There might be a conflict between these four situational rules under certain circumstances. In those situations, the decision should be made according to the principle with higher priority. The workflow of the reasoning mechanism used to develop personalized evacuation strategies is shown in Fig. 5. The strategy computation includes four steps. Step 1 calculates relevant information in the SFRH-matrix, step 2 deals with the evaluation of the situation of each person, step 3 with development of a raw strategy, and step 4 with refinement of the strategy. An in-depth explanation on the operation of the reasoning mechanism is given below.

Step 1: The values regarding the spatial features defined in the SFRH-matrix are calculated based on the sensor readings. Calculation of the information in the SFRH-matrix includes two parts: calculation of distances between any two entities and calculation of information in the attribute profiles of people and exits. In the upper triangular submatrix, the distances DP-P, DP-E, and DE-E represent the length of a route a person should walk along between two positions in a building, while the distances DP-F and DE-F represent the length of the route of fire. Normally, they are longer than the length of the direct lines, since the blocks and corners in the building need to be considered. Focusing on the reasoning mechanism, we calculate the direct distance between any two entities in the space using the SFRH-matrix. In addition, considering the imprecise results and accidental errors of indoor positioning of people, we employed Karman filter to remove the fluctuation on the values of people locations in different increments of time. The cells in the lower triangular submatrix are used to store the predicted distances of the next increment of time.

A typical attribute profile used to represent the attributes of a person is shown in Fig. 6(a). As an example, P_p, where $p \in \{1,\ldots,m\}$, is the identification number of a person and is reported by indoor positioning devices for processing in the corresponding attribute profile. In our case, the IP addresses of mobile phones are used as the identification number of the users. The speed of moving of people can be calculated based on the distance changes between two subsequent increments of time

$$V_{P_p} = \frac{\Delta L_{P_p}}{\Delta t} \tag{1}$$

where V_{P_p} is the speed of motion of a person, P_p, and ΔL_{P_p} is the location change of P_p. Thus, the status of a person can be derived based on setting a threshold on the speed of motion, either standing or moving. Target exit of P_p can be known by comparing the direction that the person is moving toward and the direction where an exit is. If a person is moving toward the same exit over three subsequent situations, then this exit is regarded in the reasoning as the target exit, E_{Target}, of this person. Thus, the identification of the relevant exit involves context-dependent reasoning, and the result is the number of an exit through which the person is supposed to escape.

The possibility of obtaining information concerning human dynamics is another important aspect and advantage of the proposed computational mechanism. Human attitude and behavior should be considered in the context-dependent reasoning operation, since the users might neglect, obey, partly obey, or disobey the instructions given to them. Therefore, the people are classified according to if and how information and instructions can be communicated to them. Thus, we sorted people in the space into four categories, including (i) skillful informable people (e.g., firefighters or guards), who can help not-informable people, (ii) informable people, who are escaping according to the obtained instructions, (iii) not-informable people, who are not aware of the fire and unadvised about the evacuation, and (iv) not-informable people, who are escaping based on their own judgment.

Skillful people are continuously accessible, make decisions on their own, and can send requests to the system in order to get more information about a local situation or a particular region in the building, or inform it about entity-related or relationship-related situational changes. Informable people are supposed to have identifiable and operating mobile device with them and to obey the instructions communicated to them. Not-informable people (i) are not supposed to have operating mobile device with them, (ii) are not able to act according to the instructions, or (iii)

Fig. 5 Reasoning mechanism used for developing personalized evacuation strategies

Fig. 6 Typical attribute profiles used in the SFRH-matrix: (a) attribute profile of an exit and (b) attribute profile of a person

may even disobey the provided instructions. Therefore, only the skillful and the informable persons are regarded as active partners in the decision-making about the overall and the individual evacuation strategies. The reasoning mechanism is equipped with the capability of deciding upon if a person is informable or not-informable, based on the possibility of sending information to them and comparing the reactions on the instructions given to him with the expectations, while continuously monitoring the actual spatial changes of the person. Therefore, if a person is informable, then he might be informed to help any not-informable person, who is considered as a help recipient of the informable person.

A typical attribute profile of an exit is shown in Fig. 6(a). The status of an exit is obtained by sensors deployed in the doors of the exits and it indicates if the door is locked, blocked or not. The number of people involved in a jam next to it is computed based on their known momentarily locations relative to the nearby exit. It is recalculated in every time increment, unless no one approached the "boundary people" in a jam. As examples, the identification number, E_q, is a designated value in the attribute profile, the coordinates of the locations are measured, and the
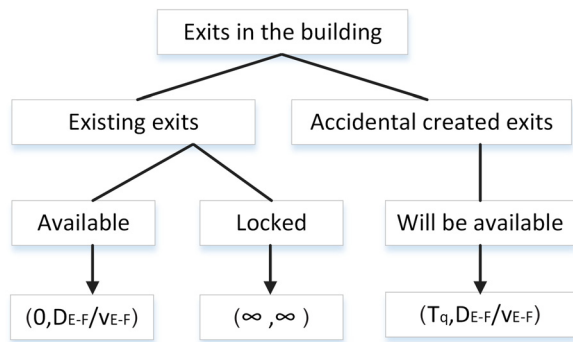
number of people that can pass through per minute, Num_q, is predefined by the system designers. The available period, ap_q, indicates the time window, which is available for a person to escape through an exit, E_q. There are two types of available exits: (i) existing available exits and (ii) incidentally created exits. The latter can be a window with a ladder outside. For an existing exit, the available time window indicates the duration of time before the fire will occupy a particular exit. This is calculated by the below formula:

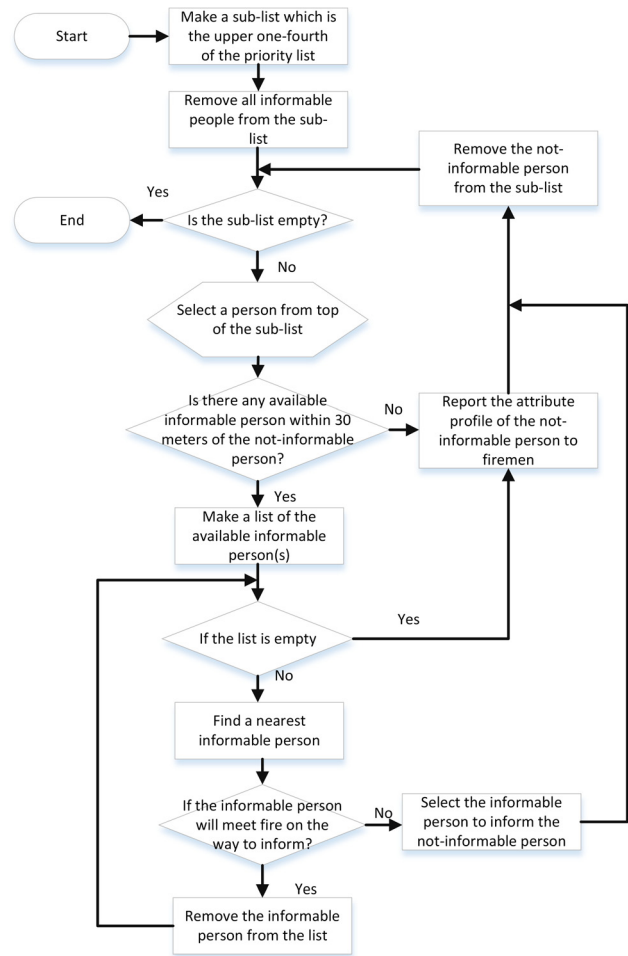$$ap_q = \left[0, \frac{D_{E_q} - F}{V_{F_q}} \right] \qquad (2)$$

Fig. 9 Flowchart of the algorithm used to refine the escape strategy considering human dynamics
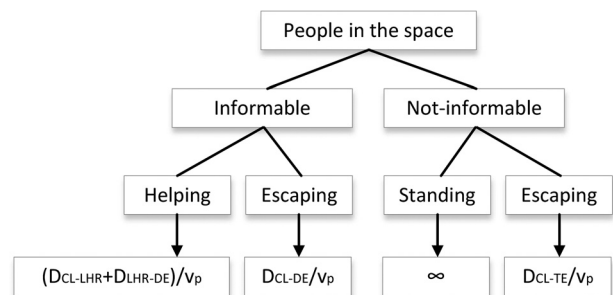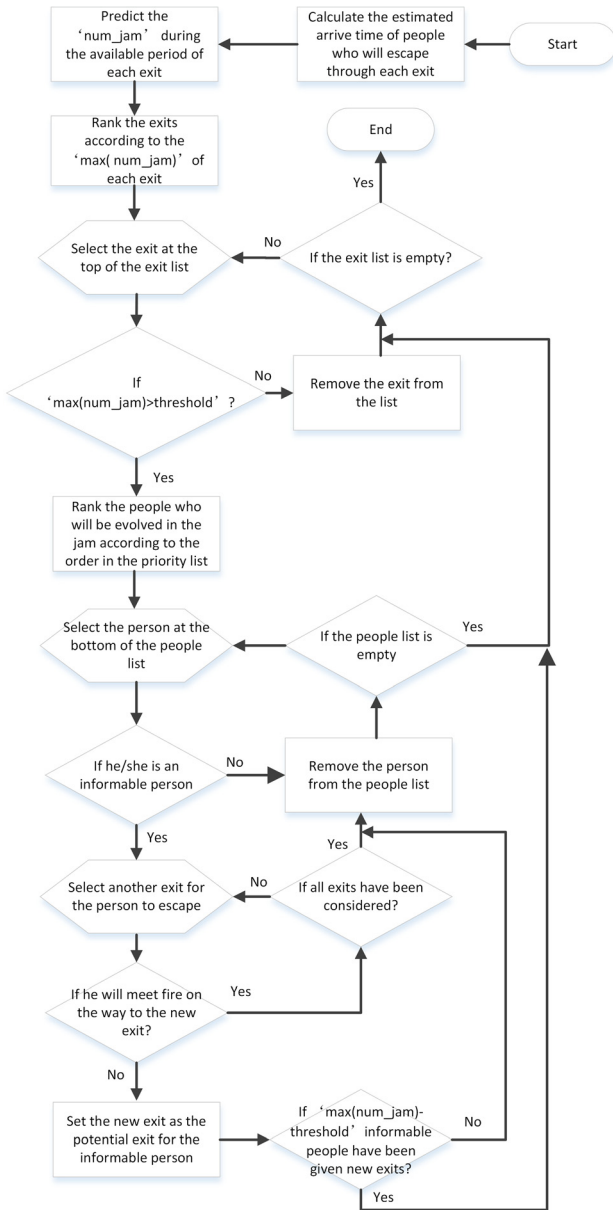

Fig. 7 Time periods available for using different types of exits
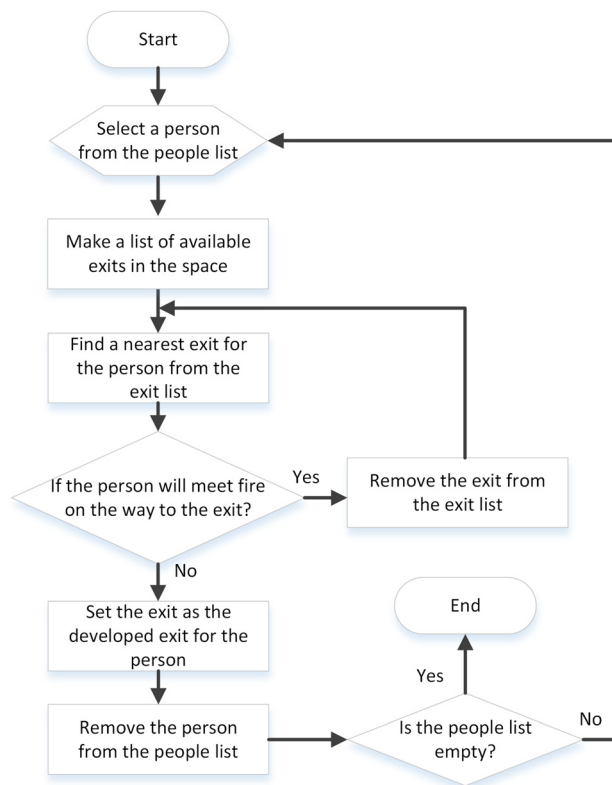
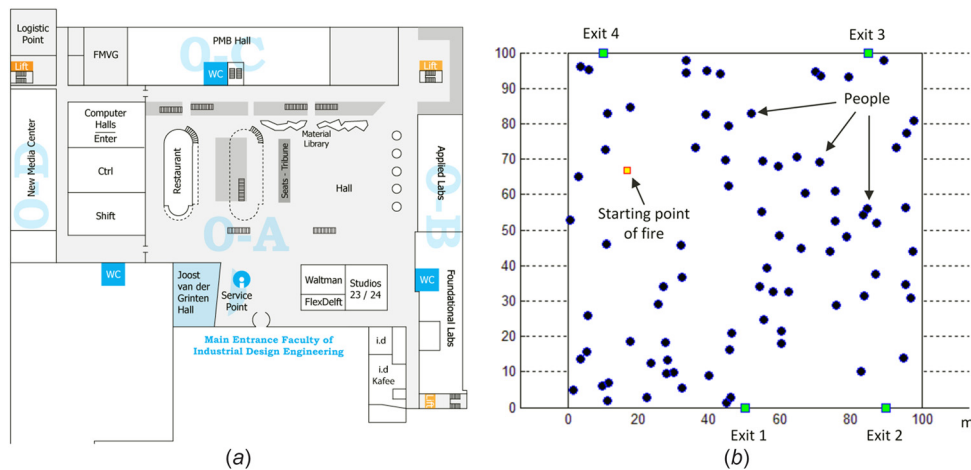Fig. 8 Flowchart of the algorithm used to develop raw strategy

Fig. 10 Time needed by the different category of people to escape in different situations

Fig. 11 Flowchart of the algorithm used to reduce the number of people involved in a jam

where D_{E_q-F} refers to the distance between E_q and the fire front, which can be queried from the SFRH-matrix, and V_{F_q} is the speed of motion of fire front toward E_q. For an incidentally created exit, the available time window represents the duration of time between the particular point in time (i.e., time increment) when the exit is created and the point in time when the fire occupies this exit

$$\mathrm{ap}'_q = \left[t_q, \frac{D_{E_q-F}}{V_{F_q}} \right] \tag{3}$$

where t_q represents the time when the exit is created. For instance, when firemen intend to build an accidental created exit by using a scale ladder on a window from outside of a building, the firemen can estimate the location of the new exit and the time when the exit will be available. An overview of the available period of time for each type of exits is shown in Fig. 7.

Step 2: Indoor evacuation aims at providing personalized escape strategy for people in different situations. A danger coefficient, DC_p, is used to evaluate the situation of a person. It provides the possibility to give priority to people who are in a bigger danger. It is obvious that the closer a person is to an exit, the safer he is, while the closer he is to the fire, the more danger he will be in. Therefore,

$$DC_p = \frac{D_{P_p-E_{\text{target}}}}{\min(D_{P_p-F})} \tag{4}$$

where $\min(D_{P_p-F})$ is the distance between P_p and the nearest fire front, and $D_{P_p-E_{\text{target}}}$ is the distance between P_p and the target exit of the concerned person. Note that the above equation is proper for evaluation of changing situations in a 2D space. This is one of the reasons why a priority list is set up based on the calculation of DC_p. The person with the highest DC_p value is put at the top of the priority list, while the person with the lowest DC_p is put at the bottom of the priority list.

Step 3: The objective of step 3 is to develop a raw solution, which is calculated based on the basic layout of the space and the information about entities in the space. The entities include people, available existing exits, accidentally created exits, and fire. Obviously, the raw strategy is not optimal since: (i) the possibility of jam formation, (ii) human dynamics, and (iii) proliferation or extinction of fire are not considered. Nevertheless, the raw strategy can be used as the basis of a first action plan and can be a subject of a follow up refinement. Procedurally, each person is guided to a relevant exit. This happens with the consideration of the shortest route for them. The strategy should avoid that any person walks toward fire. It entails that it has to be predicted if a person

Fig. 12 Floor plan of the considered building: (*a*) physical arrangement of the space and (*b*) the starting place of fire

will "meet the fire" on the way to the designated exit. The prediction can be made by comparing the distances of the person to the nearest position of the fire front at different time increments. Figure 8 shows the reasoning mechanism used for the generation of the raw strategy. In this computation, all the available exits are taken into account, in the order of the distances to the person, until a suitable exit is found. If no exit can be found for the person to escape safely, the attribute profile of the person is reported to firemen, since the person might already be surrounded by the fire in that situation. It is assumed that the firemen will take over the responsibility of rescuing the person. After receiving confirmation from a fireman, the information related to the person is removed from the SFRH-matrix in the next context computation step.

Step 4: The objective is to refine the solutions developed in step 3 considering the dynamic spatial context information. It includes two parts: refinement of the escape solutions considering human dynamics and reducing the number of people potentially involved in a jam. In both cases, the objective of shortest distance should be sacrificed for a longer, but more secure route to another exit. Figure 9 shows the reasoning mechanism used to refine the escape route proposal considering human dynamics. The dynamically changing context information is used to refine the raw strategy, and the strategy refinement is actually done based on the order in the priority list. Human dynamics derived from the dynamic spatial context information is also used to refine the raw escape solutions. This is needed since the system should not be loaded by generating and sending instructions to not-informable people. It is possible to know if a person is informable, or not, based on the attribute profile of people. Assumed is that an informable person can be informed to move to a given place, and that he can inform not-informable persons who are around the escape route proposed to him.

In order to remove each person from any dangerous situation, people with a higher value of danger coefficient should be given attention and priority. Only people who are placed in the upper one-fourth of the priority list are considered as such in each computation. In the next step of the process, all the informable people are removed from the sublist since they are supposed to be informable by the system. Thus, the objective of the computational is to find a suitable informable person for each of the not-informable people in the sublist. The most suitable informable person is determined and is given the informing task. It is predicted if the informable person will meet fire while he is engaged with informing since it is unethical to lead him in a situation that may turn to be dangerous.

Reducing the number of people potentially involved in a jam is another important issue during the planning. Toward this end, it is necessary to predict if a jam is being formed at the exits during the available period of time. If, at an exit, the number of people in a jam at a given point in time is larger than a predefined threshold, the exit is considered overcrowded during that period. The prediction of a jam is made based on the analysis of the time needed for each person to escape. These escape times for each category of people are shown in Fig. 10, where D_{CL-LHR} is the distance between the current location of the informable people and the location of the help recipient; D_{LHR-DE} is the distance between the location of the help recipient and the exit defined for them to escape; D_{CL-DE} is the distance between the current location of escaping informable people and the location of the developed exit for him; and D_{CL-TE} refers to the distance between the current location of a not-informable person and the target exit of him.

Therefore, it is possible to count how many people are supposed to select the same exit as target. For each exit, a list is made to include the people who are supposed to select that particular exit. The people in the list can be arranged according to their estimated arrival time to the exit. The variable $Num_{arrive_{q,t}}$ is used to capture the number of people that are assumed to arrive at E_q between t and $t+1$. Based on the number of people that can pass through the exit per minute, Num_q, it is possible to predict if a jam is being formed at an exit at any time (by adding the people who have arrived and removing the people who have passed

through an exit within the available time period). Then, the time period in which a jam with the largest number of people is formed can be calculated for each exit. Using this information, the exits can be ranked and the number of people involved in the biggest jams can be reduced.
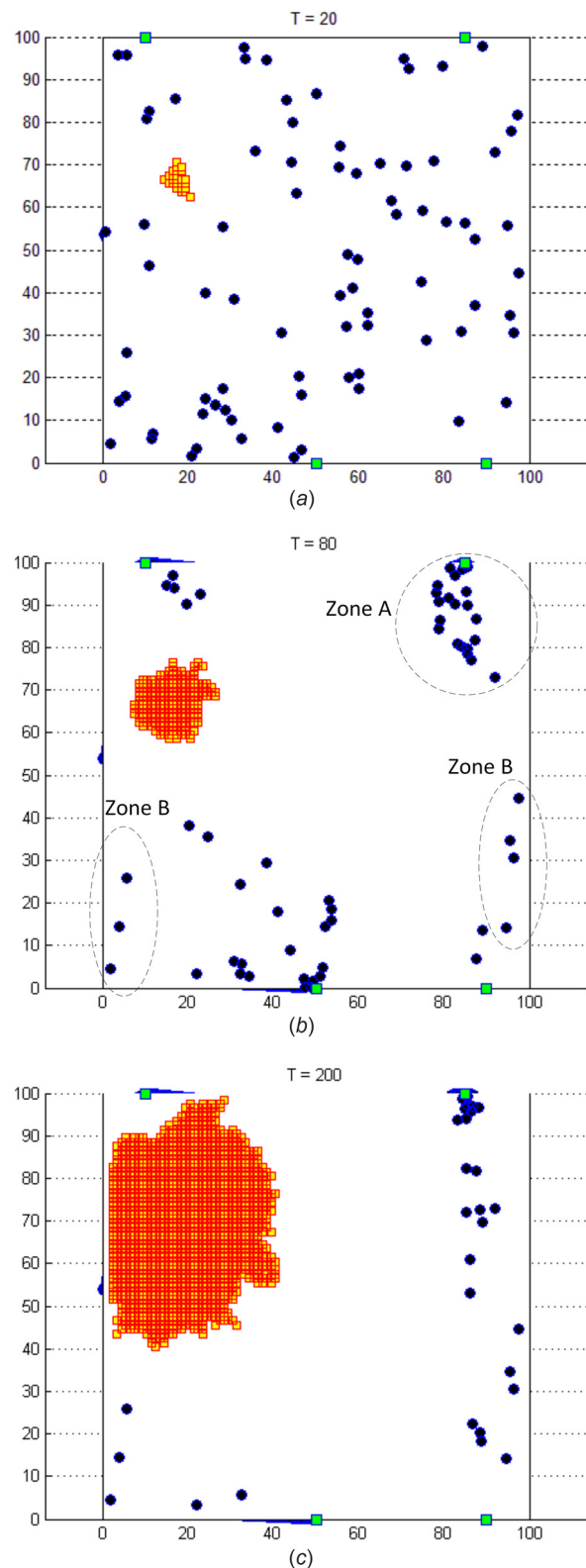
Fig. 13 Simulation of the indoor fire situations without informing by the system at (*a*) *T* = 20 s, (*b*) *T* = 80 s, and (*c*) *T* = 200 s
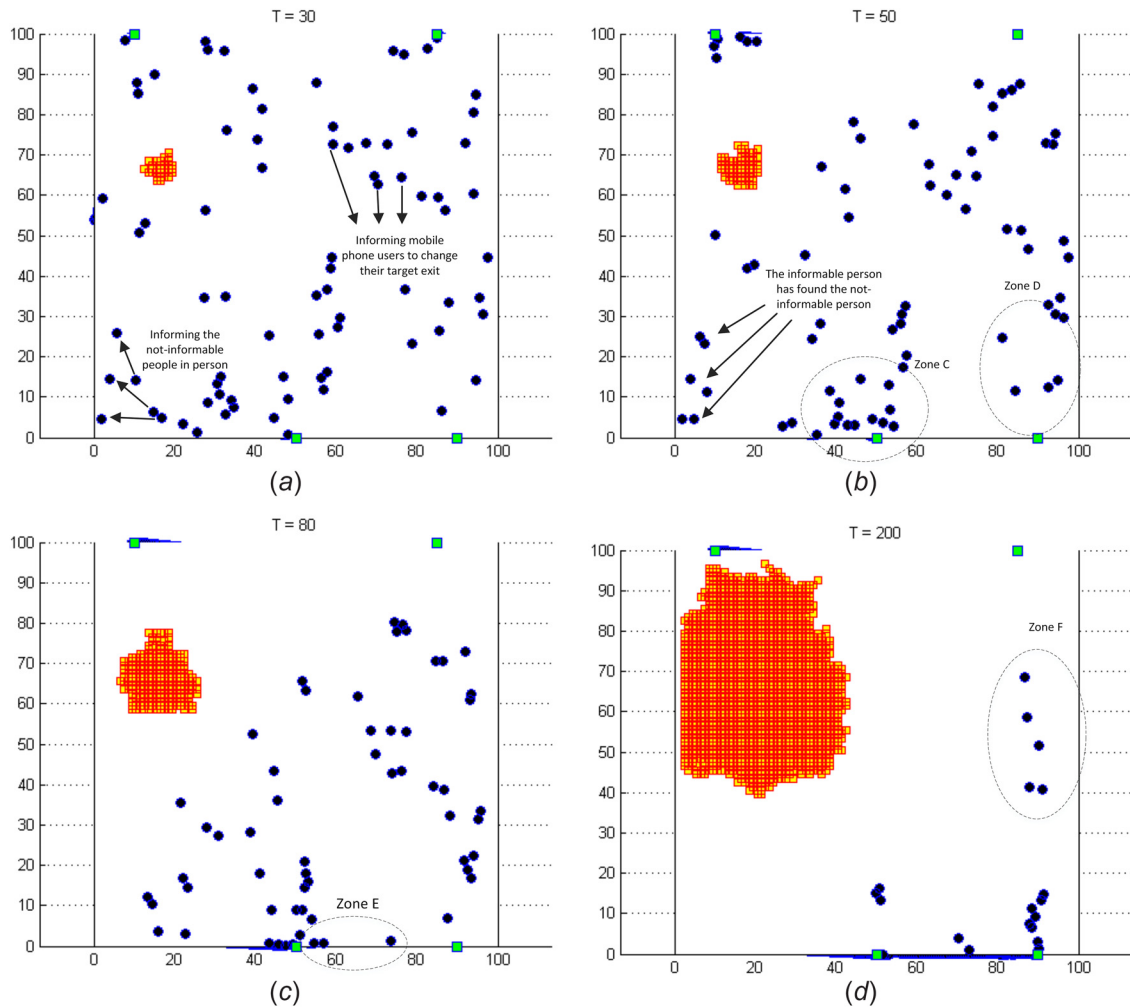
Fig. 14 Simulation of the indoor fire situations with informing by the system at (*a*) *T* = 30 s, (*b*) *T* = 50 s, (*c*) *T* = 80 s, and (*d*) *T* = 200 s

Afterward, a judgement is made concerning if an exit will be overcrowded. This is done by comparing the largest number of people in a jam and the predefined threshold. The flowchart of the algorithm used to reduce the number of people involved in a jam is shown in Fig. 11. If an exit seems to be overcrowded at a future point in time, the informable people involved in the jam are to be instructed to choose another exit as target. The relevant informable people are selected according to the reverse order in the priority list. They will be informed about another nearby exit, which is safe.

4 Validation of the Proposed Dynamic Context Computation Mechanism

4.1 Simulation of Decision-Making Based on DCI in a Practical Case. The building of the Faculty IDE was selected as a place of the hypothetical fire, and its ground floor as the basis of spatial context information computation and evacuation strategy development. We disregarded managing the people on the higher floors. The floor plan of the ground floor is shown in Fig. 12(*a*). For the modeled indoor fire situation, a space of 100 m × 100 m, 80 people, and four exits were considered. The break out point of fire in the space is as shown in Fig. 12(*b*). Speed of motion of people was considered to be between 0.5 and 0.9 m/s, obeying normal distribution pattern. The locations of people were generated randomly. The number of people that can pass through each of the exits is 60 people per minute.

In accordance with the floor plan, the location coordinates of the exits were taken as (50 m, 0 m), (90 m, 0 m), (85 m, 100 m),

and (10 m, 100 m), relative to the origin placed in the left lower corner. Exit 3 was supposed to be locked, but it was assumed that people were not aware of this. Other assumption was that 60 people were cellphone users (i.e., they could potentially be informed by the system) and that ten of them neglect the instructions given by the system. This means that 20 people were considered not-users. Both the cellphone users who neglect the instructions and the not-users were treated in the simulation as not-informable people. It was also assumed that, whenever the system could not inform a not-informable person directly, it might be able to inform an informable person and ask him to inform a not-informable person verbally. At time (*T*) = 1 s, when the fire started in the simulation, the people did not know the starting location of the fire, and it was recognized by them just when it was closer to them than 10 m. According to the used scenario, the people were supposed to do different things in the building. Twenty of them were assumed to walk on the ground floor. Furthermore, the scenario assumed that ten people were busy in the laboratories. They were warned to escape from the rooms when the fire alarm was switched on at *T* = 20 s. Hearing the fire alarm, most of the people started moving toward the nearest exit.

4.2 Simulation of the Situation Without Informing by the System. As shown in Fig. 13(*a*), when the fire alarm started at *T* = 20 s, many people moved toward the nearest exit 3, which was, however, locked. When they reached exit 3 and learnt that it was locked, they started to move toward the second nearest exit, which was exit 2. As shown in Fig. 13(*b*), a jam was formed in

zone A (indicated by the ellipse) at $T = 80$ s by the people who came from the locked exit 3. At the same time, several people were standing in zone B, since they did not realize the burst out of the fire. Therefore, at $T = 200$ s, there were still a lot of people in the space, which was rather unfavorable from the evacuation point of view.

4.3 Simulation of the Situation With Context-Dependent Informing by the System.
Figure 14(a) shows that the system started managing the dynamic spatial context information and informing the mobile phone users about the optimal escape routes at $T = 30$ s. The information about the optimal exits was computed and sent to the mobile devices of each accessible cellphone user. In addition, aiming at helping the people who were busy in the laboratories, several informable people were selected with the objective of informing them according to the priority list and the distances between them. Figure 14(b) shows that informable people managed to find and inform the assigned not-informable people directly at $T = 50$ s. Then, they started to move to the nearest safe exit, which was exit 1 in this case. At the same time, a jam was formed by people who wanted to escape at exit 1 (zone C), while the place at exit 2 (zone D) was almost empty. Therefore, the system selected several informable people and instructed them to change their target exit from exit 1 to exit 2. The threshold for the number of people at judging if a serious jam was in formation was set to seven people. Therefore, as shown in Fig. 14(c), three people were instructed to move toward exit 2 at $T = 80$ s in order to reduce the number of people involved in the jam. As it can be seen in Fig. 14(d), most of people have escaped from the space already at $T = 200$ s. The people in zone F were the mobile phone users, who neglected the instructions from the system and moved to exit 2, or the not-informable people.

5 Discussion, Reflections, and Follow Up

5.1 Discussion of the Result and the Findings.
The proposed reasoning mechanism is able to provide information and instructions for people who need to be informed in dynamically changing situations. Its reasoning and informing extends to not-informable persons, who may be informed or instructed through informable persons. In addition, the reasoning mechanism is able to predict if jam is being or was formed at the proximity of an exit and to reduce the number of people potentially involved in a jam. The objective of the conducted simulations was to generate different strategies under different circumstances using DCI. During the simulations, the SFR-matrixes were incrementally recomputed according to the data of changing situations. The time needed for computation varied significantly in the different situations. Evidently, the chosen (sampling) time step between two subsequent recomputations had to be bigger than the time needed for the completion of a cycle of information sensing, decision-making, and personalized informing. The attribute profiles of the people were generated based on the initial context information (orientation, speed, etc.). For each person, four local words (LWs) were generated. Based on the contents of the LWs, the reasoning mechanism figured out if the motions of the people would cause a jam and if a person can arrive at the target exit before the jam is formed. It also checks if the route is disturbed by fire. This way, the reasoning mechanism can choose the best exit for the concerned person. The individual escape route options were recorded at every sampling time for each person. Kalman filter was used to estimate the indoor locations of people using the biased positional information that is sensed in real-life situations. Originally, the standard deviation between the detected values and the real values was 0.60 m. After using the Kalman filter, the standard deviation between the estimated values and the real values is 0.32 m.

A comparison between the number of people computed at each exit with and without using the algorithm for jam prediction and planning is shown in Figs. 15(a) and 15(b), respectively. It can be seen that the number of people who are involved in the jam at exit
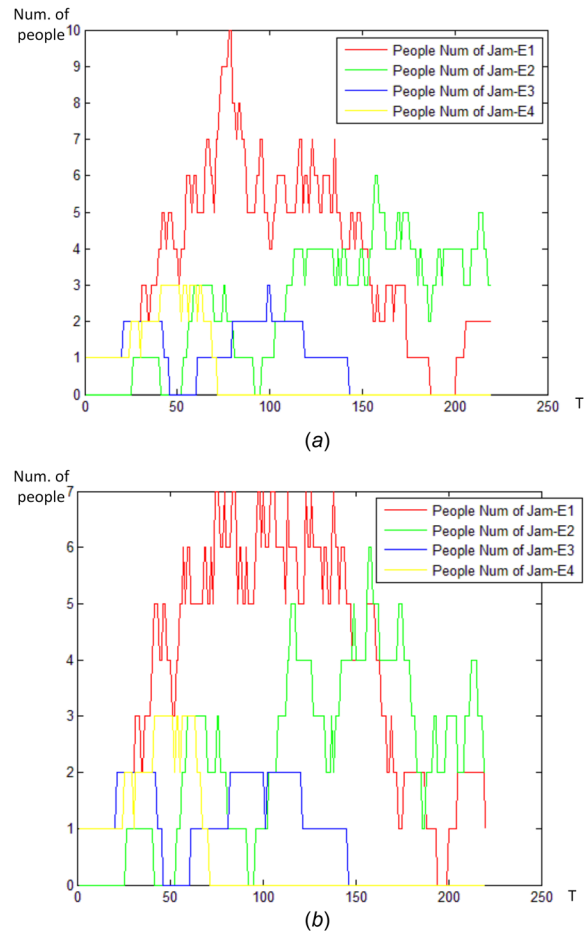
Fig. 15 The number of people trapped in a jam at various exits: (a) without reducing the number of people potentially involved in jam and (b) with application of strategy refinement in step 2

1 between $T = 70$ s and $T = 85$ s is reduced. Several informable people were instructed to move toward the second nearest exit, i.e., exit 2, which increased the number of people involved in the jam at exit 2 between $T = 105$ s and $T = 120$ s, but it was still smaller than the threshold.

5.2 Reflection on the Proposed Approach.
There are many processes in real life that are strongly influenced by the changes of the embedding environment, such as collaboration of robotic agents or evacuation of a building in case of fire. Making decisions about and controlling these processes require dynamic context computation. This is, however, a challenging task if it should be made in quasi-real time. The SFR-matrix proved to be an effective information model and computational enabler of processing DCI. If the evacuation guiding system works well, then it informs the largest possible number of people about the best escape strategy and routes and achieves that they leave the building in shortest time, while also helping others. Since all the biases increase the total evacuation time, the strategy needs to consider the expectable escape time. Therefore, together with the other context information, the needed escape time can also be used to reason about and making decisions on the best individual strategies. Another influential factor is the probability of a particular happening or of a situation to come forth. These probabilities cannot be determined exactly, but some rough estimates can be provided based on the analysis of the current situation, prediction of the behavior of the included objects, or a forecasting model created based on the previously collected information. One recognized limitation is that no consideration is given to context information

management and strategy development for multiple floors within a building.

5.3 Follow Up Research. In our further research, we will focus on the implementation and validation of the sensing, computing, and actuating (SCA) platform. As a first step, we are going to decontextualize the above presented reasoning mechanism and to make it capable to provide dynamic special context computation and reasoning in an application semantics independent manner. Afterward, we aim at implementing a functional prototype of the generic kernel of the SCA platform. After that, the external integration of the SCA platform by using a commercial WSN manager, an application server (APP-S), and Android smart phones will be implemented. Consequently, considering the crowd management strategy, the person oriented instruction editor functionality, and message scheduling and communication functionalities will be implemented in the form of software agent components. We intend to develop a software for Android smart phones, which will be able to collect information about the indoor locations of people, to process this and other personal information, and to execute scheduled informing actions toward the users. On the validation side, our follow up research will include actions for functional and usability validation of the individual components and the SCA platform as a whole. As a complement of the digital simulation, real-life application testing will be conducted. The planned internal validation will focus on the assessment of the underpinning concepts, theories, and methods, in order to optimize the operation of the dynamic spatial context reasoning mechanism and the SCA platform. Through external validation, the effectiveness of the SCA platform will be tested in another application context. As a possible application of the dynamic spatial context reasoning mechanism, a smart marketplace navigation system will be considered.

References

[1] Buchholz, T., Küpper, A., and Schiffers, M., 2003, "Quality of Context Information: What It is and Why We Need It," 10th International Workshop of the HP OpenView University Association (HPOVUA'01), Vol. 2003.

[2] Da Rocha, R. C., and Endler, M., 2012, Context Management for Distributed and Dynamic Context-Aware Computing, Springer Science & Business Media, London, UK, pp. 1–92.

[3] Tanca, L., Bolchini, C., Quintarelli, E., Schreiber, F. A., and Orsi, G., 2011, "Problems and Opportunities in Context-Based Personalization," VLDB Endowment, 4(11), pp. 1–4.

[4] Lee, I., Sokolsky, O., Chen, S., Hatcliff, J., Jee, E., Kim, B., and Venkatasubramanian, K. K., 2012, "Challenges and Research Directions in Medical Cyber-Physical Systems," Proc. IEEE, 100(1), pp. 75–90.

[5] Gwizdka, J., 2000, "What's in the Context," Computer Human Interaction, Vol. 2000, pp. 1–4.

[6] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P., 1999, "Towards a Better Understanding of Context and Context-Awareness," Handheld and Ubiquitous Computing, Springer, Berlin, pp. 304–307.

[7] Debes, M., Lewandowska, A., and Seitz, J., 2005, "Definition and Implementation of Context Information," Second Joint Workshop on Positioning, Navigation and Communication, pp. 63–68.

[8] Dey, A. K., and Abowd, G. D., 1997, "CyberDesk: The Use of Perception in Context-Aware Computing (Extended Abstract)," Workshop on Perceptual User Interfaces, Oct. 19–21, pp. 26–27.

[9] Dey, A. K., and Abowd, G. D., 2000, "Towards a Better Understanding of Context and Context-Awareness," Workshop on What, Who, Where, When, and How of Context-Awareness, CHI.

[10] Schmidt, A., Beigl, M., and Gellersen, H. W., 1999, "There is More to Context Than Location," Comput. Graphics, 23(6), pp. 893–901.

[11] Strang, T., and Linnhoff-Popien, C., 2004, "A Context Modeling Survey," First International Workshop on Advanced Context Modelling, Reasoning and Management, Nottingham, UK.

[12] Fernández-de-Alba, J. M., Fuentes-Fernández, R., and Pavón, J., 2015, "Architecture for Management and Fusion of Context Information," Inf. Fusion, 21(1), pp. 100–113.

[13] Malandrino, D., Mazzoni, F., Riboni, D., Bettini, C., Colajanni, M., and Scarano, V., 2010, "MIMOSA: Context-Aware Adaptation for Ubiquitous Web Access," Pers. Ubiquitous Comput., 14(4), pp. 301–320.

[14] Fuchs, F., Hochstatter, I., Krause, M., and Berger, M., 2005, "A Metamodel Approach to Context Information," Third International Conference on Pervasive Computing and Communications Workshops, Los Alamitas, CA, IEEE, pp. 8–14.

[15] Henricksen, K., Indulska, J., and Rakotonirainy, A., 2002, "Modeling Context Information in Pervasive Computing Systems," Pervasive Computing, Springer, Berlin, pp. 167–180.

[16] Castro, P., and Munz, R., 2000, "Managing Context Data for Smart Spaces," IEEE Pers. Commun., 7(5), pp. 44–46.

[17] Henricksen, K., and Indulska, J., 2004, "Modelling and Using Imperfect Context Information," Second Annual Conference on Pervasive Computing and Communications Workshops, Orlando, FL, Mar. 14–17, IEEE, pp. 33–37.

[18] van Kranenburg, H., and Eertink, H., 2005, "Processing Heterogeneous Context Information," International Symposium on Applications and the Internet Workshops, pp. 140–143.

[19] Gu, T., Wang, X. H., Pung, H. K., and Zhang, D. Q., 2004, "An Ontology-Based Context Model in Intelligent Environments," Communication Networks and Distributed Systems Modeling and Simulation Conference, pp. 270–275.

[20] Wang, X. H., Zhang, D. Q., Gu, T., and Pung, H. K., 2004, "Ontology Based Context Modeling and Reasoning Using OWL," Second Annual Conference on Pervasive Computing and Communications Workshops, Orlando, FL, Mar. 14–17, IEEE, pp. 18–22.

[21] Kokinov, B., 1995, "A Dynamic Approach to Context Modeling," LAFORIA, 95(11), pp. 199–209.

[22] Forsström, S., and Kanter, T., 2012, "Enabling Continuously Evolving Context Information in Mobile Environments by Utilizing Ubiquitous Sensors," Mobile Networks and Management, Springer, Berlin, pp. 289–303.

[23] Euzenat, J., Pierson, J., and Ramparany, F., 2008, "Dynamic Context Management for Pervasive Applications," Knowl. Eng. Rev., 23(1), pp. 21–49.

[24] Kim, Y., and Mahapatra, R. N., 2009, "Dynamic Context Management for Low Power Coarse-Grained Reconfigurable Architecture," 19th ACM Great Lakes Symposium on VLSI, Boston, MA, ACM, New York, pp. 33–38.

[25] Brown, P. J., 1996, "The Stick-e Document: A Framework for Creating Context-Aware Applications," Electron. Publishing, 8(2–3), pp. 259–272.

[26] Aktas, M. S., Fox, G. C., and Pierce, M., 2005, "Managing Dynamic Metadata as Context," International Computational Science and Engineering Conference, Istanbul, Turkey.

[27] Grossmann, M., Bauer, M., Hönle, N., Käppeler, U. P., Nicklas, D., and Schwarz, T., 2005, "Efficiently Managing Context Information for Large-Scale Scenarios," Third International Conference on Pervasive Computing and Communications, Kauai, HI, Mar. 8–12, IEEE, pp. 331–340.

[28] Taconet, C., Kazi-Aoul, Z., Zaier, M., and Conan, D., 2009, "Ca3M: A Runtime Model and a Middleware for Dynamic Context Management," On the Move to Meaningful Internet Systems: OTM 2009, Springer, Berlin, pp. 513–530.

[29] Villegas, N. M., Müller, H. A., Muñoz, J. C., Lau, A., Ng, J., and Brealey, C., 2011, "A Dynamic Context Management Infrastructure for Supporting User-Driven Web Integration in the Personal Web," 2011 Conference of the Center for Advanced Studies on Collaborative Research, Toronto, ON, Canada, IBM, Riverton, NJ, pp. 200–214.

[30] Jaroucheh, Z., Liu, X., and Smith, S., 2010, "CANDEL: Product Line Based Dynamic Context Management for Pervasive Applications," 2010 International Conference on Complex, Intelligent and Software Intensive Systems, Krakow, Poland, Feb. 15–18, IEEE, pp. 209–216.

[31] Dey, A. K., Abowd, G. D., and Salber, D., 2001, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," Hum. Comput. Interact., 16(2), pp. 97–166.

[32] Kwon, O., 2006, "The Potential Roles of Context-Aware Computing Technology in Optimization-Based Intelligent Decision-Making," Expert Syst. Appl., 31(3), pp. 629–642.

[33] Brown, P. J., Bovey, J. D., and Chen, X., 1997, "Context-Aware Applications: From the Laboratory to the Marketplace," IEEE Pers. Commun., 4(5), pp. 58–64.

[34] Anthony, R., Chen, D., Pelc, M., Persson, M., and Törngren, M., 2009, "Context-Aware Adaptation in DYSCAS," Electron. Commun. EASST, 19, pp. 1–15.

[35] Dai, P., and Xu, G., 2008, "Context-Aware Computing for Assistive Meeting System," 1st International Conference on Pervasive Technologies Related to Assistive Environments, Athens, Greece, ACM.

[36] Baldauf, M., Dustdar, S., and Rosenberg, F., 2007, "A Survey on Context-Aware Systems," Int. J. Ad Hoc Ubiquitous Comput., 2(4), pp. 263–277.

[37] Vieira, V., Tedesco, P., and Salgado, A. C., 2011, "Designing Context-Sensitive Systems: An Integrated Approach," Expert Syst. Appl., 38(2), pp. 1119–1138.

[38] Cortez, R. A., 2009, "A Cyber-Physical System for Situation Awareness Following a Disaster Situation," Realtime Systems Symposium, Washington, DC, IEEE.