

A multi-agent framework for cloud-based management of collaborative robots

*International Journal of Advanced
Robotic Systems*
July-August 2018: 1–13
© The Author(s) 2018
DOI: 10.1177/1729881418785073
journals.sagepub.com/home/arx



Tooba Samad¹, Sohail Iqbal¹, Asad Waqar Malik¹,
Omar Arif¹ and Peter Bloodsworth^{1,2}

Abstract

This article presents a cloud-based multi-agent architecture for the intelligent management of aerial robots in a disaster response situation. In a disaster scenario, a team of highly maneuverable quadcopters is deployed to carry out surveillance and decision support in disaster-affected areas. In Pakistan, such events usually result from sudden unpredictable calamities such as earthquakes. The aim of this work is to develop a robust mechanism to autonomously manage and react to sensory inputs received in soft real time from an unstructured environment. The immediate goal is to locate the maximum number of trapped, injured people within a large area, and help first responders plan rescue activities accordingly. To evaluate the proposed framework, a number of simulations are carried out using GAMA platform to emulate a disaster environment. Subsequently, algorithms are developed to survey an affected geographical area through the use of small flight drones. The key challenges in this work are related to the combination of the domains of multi-agent technology, robotics, and cloud computing for effectively bridging the cyber world with the physical world. Therefore, the proposed work demonstrates the effective use of a limited number of drones to capture inputs from a disaster situation in the physical world, and such inputs are used for timely planning of rescue efforts. The results of fixed resource assignment are compared with the proposed reactive assignment strategy, and it clearly shows a significant improvement in terms of resource usage compared to traditional approach.

Keywords

Multi-agent systems, cloud computing, quadcopters, disaster management

Date received: 12 February 2018; accepted: 30 May 2018

Topic: Special Issue - Distributed Robotic Systems and Society

Topic Editor: Andrey V Savkin

Associate Editor: Christian Penalzoa

Introduction

Over the last few decades, technology has evolved significantly leading to the development of smart systems which have helped human beings to accomplish complicated tasks effectively. Disaster management is one such domain where innovative solutions can help save precious lives in the wake of natural catastrophes, especially earthquakes.

Owing to the highly unpredictable nature of tectonic activities, high-magnitude earthquakes cause substantial damage to buildings, roads, communication networks, and other infrastructure. This often results in significant loss of life and the destruction of livelihood. The main challenge

faced by first responders is to provide immediate assistance to trapped or injured people and transfer them to safer locations. However, the presence of an unstructured

¹ School of Electrical Engineering and Computer Science (SEECs), National University of Sciences and Technology (NUST), Islamabad, Pakistan

² Department of Computer Science, University of Oxford, Oxford, UK

Corresponding author:

Asad Waqar Malik, School of Electrical Engineering and Computer Science (SEECs), National University of Sciences and Technology (NUST), H-12, Islamabad 44000, Pakistan.

Email: asad.malik@seecs.edu.pk



environment spread over a large geographical area poses significant hurdles in planning rescue operations, especially with limited resources. Therefore, innovative technological means are sought to effectively survey an affected area in the shortest possible time.

For this purpose, one of the possible solutions can be from the field of robotics, where Yan et al.¹ proposed a multi-robot system for the collective accomplishment of predefined goals. However, several of the heterogeneous characteristics of the distributed systems have prevented the transition of the technology from academic institutes to real-world problems as mentioned by Ferrer.² Within academic settings, the focus has been mainly on synthetic toy world problems. The real-world problems, such as effective disaster management that requires robots, or small flight drones, to survey the disaster area and accordingly perform intelligent actions, have received little attention. Furthermore, such systems often have shortcomings when it comes to intelligent control and collaboration of multiple robots with common and conflicting goals as mentioned by Gzel et al.³ and Nikolaidis et al.⁴ Nevertheless, there were few research projects that aim to develop robotic tools which can assist during rescue operations, such as the FP7 ICARUS (<http://www.fp7-icarus.eu/>) project. Similarly, Amazon Robotics⁵ (formerly Kiva Systems) has developed a material handling multi-robot system applicable in an industrial setting.

Multi-agent systems can be highly beneficial in uncertain environments as they have the capability to manifest self-organization, self-steering, and other autonomous control paradigms. These are achieved through sharing of knowledge via common language platforms within the constraints of the specified system.⁶ Moreover, multi-agent systems can find solutions to problems without any significant human intervention, and their innate flexibility makes them more suitable as they can be modified and reconstructed without the need for detailed rewriting, one such solution is presented by Michael.⁷ Hence, these systems are self-recovering and to some extent failure resistant, as the components are quite redundant and possess self-managing features. The combination of agents with robotics provides unique opportunities to achieve advanced functionalities for the abovementioned application.

A common drawback for the deployment of such complex systems is that they require considerable computing resources, which increase the cost of the underlying infrastructure. Additionally, the installation of a multi-agent system on top of a robot-controlled mechanism can also result in additional power requirements for a small flight drone with limited battery life. Therefore, the incorporation of cloud computing technology into the existing scenario holds the key, and the integration of multi-agent systems with the cloud can result in the high performance of systems as well as the development of complex intelligent applications that can sense things from real world and also alter them remotely.⁸

Proposed solution

In this article, we propose a system that has been developed to merge robotics, cloud computing, and multi-agent systems in order to control and coordinate the actions of a group of robots based on sensory inputs. The proposed system is intended to survey a disaster-affected area, following the event of sudden calamities such as earthquakes, and help locate victims in a timely manner. Disasters like an earthquake, resulting in an unstructured environment, locating victims and providing them immediate assistance, become an onerous task to achieve by humans alone. Therefore, the proposed system uses a group of autonomously managed quadcopters to ascertain the actual state of the disaster site and help first responders plan rescue efforts accordingly. It also takes into account robotic limitations of battery time and flight duration to manage resources intelligently. The other salient features of the proposed system are as follows.

- Highly scalable, even when deployed on the local server—high availability mode is adopted to provide uninterrupted guidance to mobile agents.
- The proposed system is also evaluated over cloud environment.
- The disaster area is monitored through multiple coordinated quadcopters.
- The quadcopters are represented as agents which can collaborate with each other and they can also access a central server for efficient use of resources and overall planning.
- The quadcopters are equipped with a smart device that provides Bluetooth and cellular communication.
- Bluetooth is used to acquire the input from stranded people.
- Cellular communication is used between quadcopter to quadcopter and quadcopter to central server.
- The proposed system uses reactive quadcopter assignment over the disaster site. This helps in effective resource management as shown in the “Results” section. Moreover, the proposed system can utilize historical data for effective decision-making as explained in the “Quadcopter management” section.

The rest of the article is organized as follows. The second section gives a brief overview of existing multi-agent architectures and disaster management systems. The third section describes the proposed cloud-based multi-agent architecture. The details of simulations that are carried out using the GAMA platform are discussed in the fourth section. The methods used for the evaluation of the proposed system and the obtained results are described in the fifth section. Finally, our conclusions and future work are outlined in the last section.

Related work

This section is subdivided into two sections. The first subsection covers the current approaches adopted toward the

design of multi-agent systems, whereas the second subsection discusses the related research contributions. Hence, the proposed system is developed keeping in view the advantages and possible drawbacks of the approaches discussed below. This formed the basis for evaluating the developed system and obtaining meaningful results.

Types of multi-agent systems

Essentially, there are three main approaches to designing multi-agent architectures for control software as discussed by Mouad et al.⁹ These are briefly stated as follows.

Reactive versus deliberative. Architectures, which fall in the reactive category, implement a behavior that provides immediate response to a stimulus. In contrast to architectures based on deliberative behaviors, the response to each type of stimulus is rationalized keeping in view a set of predefined conditions. Therefore, a choice between the two designs is heavily dependent on the intended application.

Hierarchical versus nonhierarchical. Hierarchical designs present a layered architecture where upper layers are dedicated for decision-making, intermediate layers deal with supervision and control, and repetitive methods such as actuator control are handled by the lower layers. Nonhierarchical architectures often face shortcomings in terms of a lack of modularity, scalability, and decentralized control. Finally, hybrid architectures are a combination of hierarchical and nonhierarchical architectures.

Hybrid architectures are created from a combination of the existing architectures.

Centralized systems versus distributed systems. Centralized systems include a supervisor agent that takes over the leadership role and is responsible for decision-making, organizing, and planning for the whole team. This approach, however, often suffers from a singular point of failure. In contrast to a centralized approach, a distributed system is a collection of autonomous agents which collectively take part in the decision-making process.

Research contributions

The use of unmanned aerial vehicles (UAVs) to support first responder services has been increasing over the last decades. However, there are a number of problems that still need exploration. Some of the challenging areas in UAV management are path planning, coordination with other UAVs in real time, flight time limitations, and the effective management of UAVs for timely task completion. Similarly, Ren et al.¹⁰ have explored the consensus problem in multi-agent-based coordination. The consensus becomes more challenging under situations like variable communication delay. Ad hoc communication among UAVs to complete the designated task is another open research area. Olsson et al.¹¹ proposed a UAV-based ad hoc

communication network for monitoring and surveillance. Broadly, the UAV network still faces power-related issues. With limited power, computing capacity and storage often create hindrance to real-time applications. To overcome the limitations of computing capabilities within UAVs, powerful onboard computers have been installed in them but these have further reduced the available power, thereby reducing the maximum flight duration. Overall, the UAVs are used to cope with disaster situations; therefore, in certain situations, a significant amount of sensory data might need to be handled with limited resources. Moreover, retransmission due to noisy conditions, unexpected node failure, and limited power supply are some of the research challenges that need to be further explored.¹²

There are several different multi-agent systems that have been applied to control multiple robots. They follow different multi-agent architectures based on their application needs. The systems that require hard real-time, non-deterministic, and dynamic environments often utilize specialist multi-agent architectures such as ARTIS.¹³ ARTIS allows the modeling of agents within hard real-time constraints that lets them control the environment through a set of sensors and effectors in order to generate a response within hard temporal restrictions. Essentially, ARTIS is an extension of the blackboard model modified to meet hard real-time constraints.¹⁴

Similarly, Mosterman et al.¹⁵ proposed the IDEA architecture designed to support multi-robot applications, where each agent can change its role as a planner, a functional module, or a diagnostic system. In this architecture, each agent can monitor and communicate with other agents, but it does not rely on organizational rules to build agents. Ezequiel et al.¹⁶ proposed similar architecture based on the Prometheus methodology which supports the design of goal-oriented agents, covering a range of activities from requirement specification to detailed design. Their proposed architecture is a three-phase, agent-based design built using the Prometheus methodology and is used to analyze system functionality, which builds operation scenarios and designs the MAS coordination architecture after defining the types of agents through role-agent mapping.

Galindo et al.^{17,18} proposed an architecture for controlling robotic wheelchairs, which is both a modification and an extension of ACHRIN (hybrid) architecture. The components in ACHRIN are designed in a client-server fashion, which can make modification and expansion by incorporating the multi-agent system and implementing each component of ACHRIN as a separate autonomous agent. Several systems require multi-robot coordination and cooperation to achieve a particular goal in games like soccer.^{19,20} For this purpose, layered architectures have been proposed to facilitate coordination and cooperation of a team of robots where each layer provides services to upper layer. The controlling multi-agent system resides on a local server which gets information about field through visual sensors, makes decisions accordingly, and communicates its decision to the robots via effectors.

When a group of robots need to move in an unstructured environment while avoiding obstacles, coordination and cooperation issues are assigned the highest priority. In such scenarios, a hybrid architecture, which takes into account the advantages of hierarchical and reactive architectures to minimize the response time and to carry out planning tasks in order to achieve coordination through distributed and centralized aspects, is highly preferred.

Fierro et al.²¹ proposed a multi-robot coordination paradigm for managing cooperative behavior of finite state robots. It focuses on methodologies regarding ease of control for application-specific robotic systems. Unfortunately, the proposed paradigm works indoors within a structured environment and fails to address tasks involving communication and information sharing. The coordination strategies presented by Nieto-Granda et al.²² explore various strategies used by a group of mobile robots to map and explore uncertain/unstructured environment. The approach is based on shared resources, which are distributed among the team of robots rather than one single expensive machine. However, the robots have limited storage and processing capacity, which can fall short when used for complex applications. Du et al.²³ proposed a complex architecture for a mobile robot to facilitate an autonomous behavior. However, this architecture is for a single robot and does not address the coordination issues, which arise when working with multi-robot applications. Crooks and Sarah²⁴ presented a prototype to illustrate a post-disaster scenario for better planning and coordination of humanitarian relief efforts. It relies on agent-based modeling and geographic information obtained from crowd movements, along with other publicly available data sources. The novelty of the system is that it integrates raster data collected from several sources and vector data structures in the same simulation. The relevant research work successfully utilizes agent-based modeling and geographical information to facilitate rescue efforts; however, the crowdsourced component of information used by this system lacks structure and proper format as it relies on people in times of panic that may lead to inaccurate or partially accurate information.

Predicting the movements of a large population during the disaster is a complex task. Chen et al.²⁵ proposed a system to predict the movement of people using cellular and geographical information system (GIS) data, which continuously update simulation with real-time data by tracing the movements of cell phone users. Once the data set has been generated, the system infers those areas, which people prefer or avoid, for predicting movements. The validation of the results is done using Manhattan distance technique to show generated predictions are valid, both internally and predictively. However, there exists the possibility of a significant compromise on the timeliness of the system, which relies heavily on the availability of adequate computational capacity under times of duress when large volumes of data are being received continually.

Agent-based modeling techniques can also be used to assess the resilience of a community in the event of a major

disaster. Boston et al.²⁶ proposed a system based on a virtual city. The scenario of a seismic event is used as input to show the damage sustained and interactions between people predisposed by shock and panic.

Modeling human behavior during typical situations remains a difficult task. In this respect, agent-based systems provide a paradigm to simulate human behavior in the environment created by real-world or synthetic data because it is important to study human and traffic movements within the affected area as discussed by Frank and Paul.²⁷ For this purpose, Okaya et al.²⁸ proposed a hybrid model to simulate populations' movement through a large number of agents in the affected areas without utilizing intensive computational resources. The model is capable of simulating the behavior of agents in wide areas through network, free-space, and integrated models. The data being used for simulation purposes can be considered comprehensive and accurate as they combine structured data from two computational models and corroborate it with an actual simulation. Moreover, the efficiency of the hybrid simulator makes the information readily accessible as less time is required to determine agent behavior and the results can be shared easily.

Ideally, emergency response requires preplanning and maximum possible information about an unexpected event prior to its occurrence, but an ever-present threat of potential events rarely follows such ideal patterns. Also, in times of great uncertainty, it is difficult to retain a formal command structure for emergency responders as roles and responsibilities keep evolving according to the situation. In fact, it is often a challenge to accurately track their positions and movements. For this purpose, Coates et al.²⁹ presented an integrated framework that has been developed with decision support and agent-based simulation to help practitioners related to disaster management plan effectively for emergency response. The approach illustrated here has been proven to be quite successful in terms of formulating a timely response because of significant computational efficiency, therefore the resulting operational models for rescue efforts can be shared readily. Moreover, the accuracy of the system can be verified through evidence-based corroboration of actual scenarios, thus making the system adaptable to any major events that might unfold.

Currión et al.³⁰ proposed an example of more conventional open-source software developed for disaster management and relief efforts. The designated officials, who are accountable for its accuracy, record the data being used in a structured format. Another useful feature of the system is that it provides access to information even in areas where there is little or no connectivity using distributed repositories at authorized locations. The data are recorded manually before being translated into an electronic format. This results in delayed response.

Based on the discussion thus far, effective decision-making aided by agent-based systems has significant merits; however, relatively evolved conventional systems can be equally effective in smarter planning of rescue efforts. In this

Table 1. Comparison of disaster management systems based on damage assessment, monitoring, multi-agent support, resource optimization, and fault tolerance.

Authors	Damage assessment	Monitoring	Multi-agent support	Resource optimization	Fault tolerance
Wada et al. ³²	Yes	No	No	Yes	No
Ueyama et al. ³³	No	Yes	Yes	Yes	No
Mosterman et al. ¹⁵	Yes	Yes	Yes	Yes	No
Ezequiel et al. ¹⁶	Yes	Yes	No	Yes	No
Petrovic et al. ³⁴	Yes	Yes	No	Yes	No
Erdelj et al. ³⁵	No	Yes	No	Yes	No
Crooks and Sarah ²⁴	No	No	Yes	Yes	No
Boston et al. ²⁶	Yes	No	Yes	No	No
Frank and Paul ²⁷	Yes	Yes	Yes	No	No
Othman et al. ³⁶	No	Yes	Yes	Yes	No
<i>Proposed System</i>	Yes	Yes	Yes	Yes	Yes

regard, Su et al.³¹ have presented a simulation model for planning the transport of patients to the nearest medical centers following the occurrence of a disaster. Their work is beneficial for victims of disaster as it considers a post-disaster scenario, where data are significantly comprehensive and accurate since all relevant information has been relayed by authentic sources. However, the completeness of the work may be argued, as methods to determine the location of victims still seem abstract. Moreover, efforts have been made to improve the timeliness of system results such that the waiting time for patients can be reduced and they can be shifted to nearby hospitals; however, it is the authors' opinion that a more detailed consideration is required for time taken by ambulances, especially in the case of the absence of infrastructure in the aftermath of disasters such as earthquakes. Finally, a comparison of other similar works is listed in Table 1. The comparison is drawn based on damage assessment, monitoring, multi-agent support, resource optimization, and fault tolerance.

The proposed system architecture

The proposed multi-agent framework follows a blackboard architecture, where a common or shared knowledge base blackboard is updated by a diverse set of sources; each knowledge source contributes to the solution using its partial knowledge. In other words, the blackboard model is designed to handle convoluted problems where solutions are typically the sum of many individual pieces of data. Blackboard architecture is an important approach of artificial intelligence where multiple sources (in our case, quadcopters) provide updates to a common knowledge database. It is a well-known concept used in robotics (refer to the work of Tobias et al.³⁷); however, in the domain of disaster management it has not been used to the best of the authors' knowledge.

Since the multi-agent system has to control a set of multiple robots in an unstructured environment during catastrophic events such as earthquakes, each quadcopter acts as a knowledge source, which periodically updates the shared repository (or blackboard) with its partial

knowledge about the victims of a disaster. The victim's information is conveyed to quadcopters through an android-based disaster application which uses a Bluetooth low-energy beacon to communicate to another beacon mounted at the top of quadcopter as shown in Figure 1. Using collective sensory knowledge of different quadcopters deployed for surveying disaster-affected areas, the multi-agent system fabricates a mental picture of the physical world and manages the group of robots in a way to maximize their utility based on optimal resource allocation principles. Essentially, the number of quadcopters assigned to a particular earthquake-hit area is directly dependent on the extent of damage so that accurate information about maximum possible victims can be collected and those in urgent need of humanitarian relief can be identified.

Proposed framework of multi-agent collaborative robots

The proposed system can have a real-life implementation to facilitate first responder services. Figure 1 shows the proposed framework and summarizes its use. The mobile application communicates with quadcopters through Bluetooth. A similar device is installed on every quadcopter that provides communication with the backend multi-agent system through Internet or ad hoc network.

The developed multi-agent framework embodies the following components.

- Environment monitoring
- Quadcopter management
- Immediate response planning
- Blackboard consolidation

Environment monitoring

During catastrophic events, a timely response can save many lives and minimize the post-disaster damage. Therefore, the monitoring agent of the multi-agent system repeatedly monitors the environment for disaster situations and

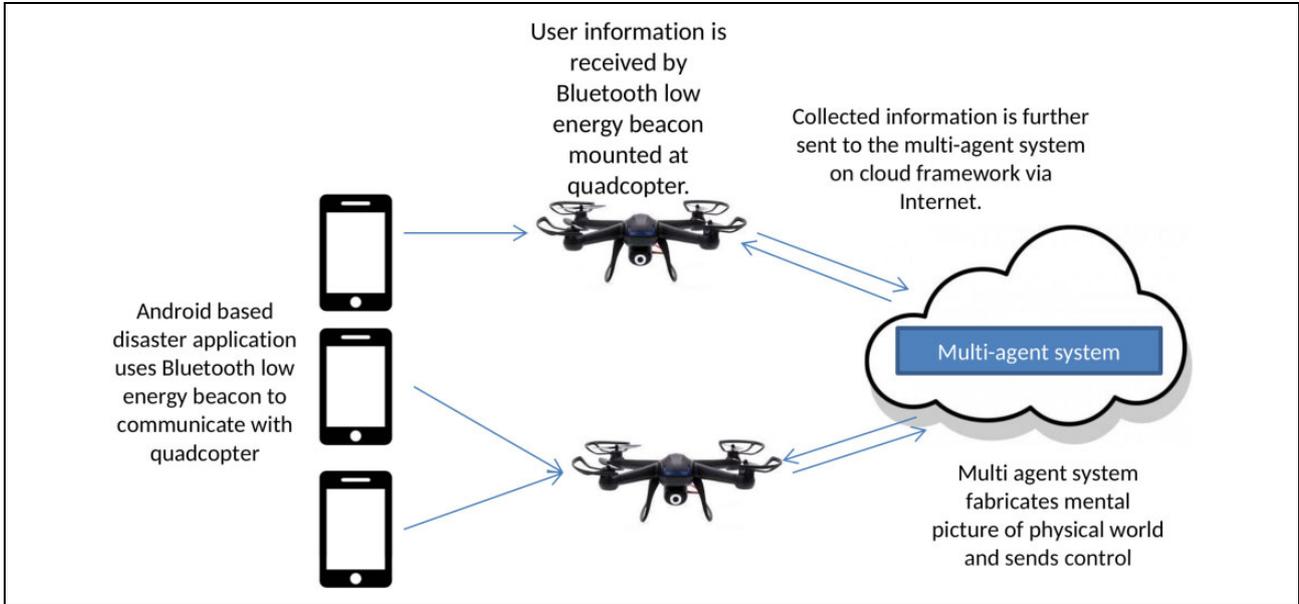


Figure 1. Proposed framework of multi-agent collaborative robots.

makes decisions accordingly such as initialization of a quadcopter management agent.

Quadcopter management

The goal of the quadcopter management agent is to utilize the available resources effectively for capturing the data about victims of an earthquake. It achieves this by remotely surveying the affected area through sensory robots (quadcopters) while addressing constraints such as limited battery life. Therefore, this module involves decisions regarding quadcopter initialization and creation of associated quadcopter agents to virtually represent each individual robot (quadcopter) from the physical environment. It also manages quadcopters during its survey by assigning survey zone to its associated quadcopter agent and controls the movement of the quadcopter between adjacent plots in a specific zone.

Quadcopter initialization. Whenever a disaster occurs, the environment monitoring agent initiates a quadcopter management agent which further needs to initialize quadcopters in a given area. The quadcopter management agent determines the number of quadcopters required in an affected area based on either of the following approaches.

1. The first approach considers previously recorded data for earthquakes in terms of magnitude, affected population, and number of quadcopters utilized for effective surveillance of an area post-disaster. If the historical record for an earthquake of similar magnitude exists in the database, the following equation will be used to determine adequate number of quadcopters required for survey in the possible disaster scenario faced currently

$$P_{(qc+1)} = P_{qc} + \left(r \times \frac{Q_n}{\Delta t} \times P_{qc} \right) \quad (1)$$

where $P_{(qc+1)}$ is the proportion of the required quadcopter to be sent, P_{qc} is the proportion of quadcopters sent, r is the rate of population growth, Q_n is the number of quadcopters sent, Δt is the number of years between the previous and current occurrences of the disaster, and N is the number of entries found in the history record.

When the historical data contain multiple records of an earthquake of similar magnitude with varying number of quadcopters deployed each time (based on the damage sustained by a community), the following equation will be used to accurately determine the current requirement

$$P_{(qc+1)} = \frac{1}{N} \left\{ \begin{array}{l} \left[P_{qc1} + \left(r \times \frac{Q_{n1}}{\Delta t_1} \times P_{qc1} \right) \right] \\ + \left[P_{qc2} + \left(r \times \frac{Q_{n2}}{\Delta t_2} \times P_{qc2} \right) \right] \\ \dots \left[P_{qcn} + \left(r \times \frac{Q_{nn}}{\Delta t_n} \times P_{qcn} \right) \right] \end{array} \right\} \quad (2)$$

The previously mentioned formulas are based on logistic models. We have used these models for prediction purposes because they not only incorporate the population growth rate but also consider the carrying capacity of the existing population.

2. The second approach is used to initially assign an adequate number of quadcopters for surveying a disaster-affected area in the absence of any historical record for an earthquake of similar magnitude. Since the previously recorded data of similar magnitude are not available, the closest higher magnitude earthquake data are used to populate the parameters in equation (1).

Quadcopter management during survey. After the initialization of quadcopter agents, individual aerial robots are assigned to their respective survey zone which consists of multiple adjacent plots. During the survey, the local agent of a quadcopter periodically updates its associated agents with current location. It also checks for messages and sends acknowledgment to the sender. Based on location updates from the local agent of a particular quadcopter, associated agent also manages the movements between different plots in assigned zone and marks a particular zone as “OK” if all plots have been surveyed. Apart from managing quadcopters assignment and movement, this component also oversees the allocation of more robots to areas with greater extent of damage based on the count of distress beacons broadcasted by the victims. The maximum number of message violations in a particular zone is denoted by m , where m is initially set to 3. As soon as the threshold is crossed, the quadcopter management agent allocates new quadcopters in that particular zone and assigns a new set of plots to them which have not been surveyed yet. The inspiration of using abovementioned features was taken from Queensland State Disaster Management Plan (Queensland State Disaster Management Plan Reviewed September 2016. Available online at <http://www.disaster.qld.gov.au/cdmp/Documents/Queensland-State-Disaster-Management-Plan.pdf>) (September 2016).

Immediate response planning

Another important aspect of the system pertains to the support it offers in the planning of immediate response efforts in the disaster area. This component uses hierarchical clustering technique to group victims into different spatial clusters. This grouping is performed on the sensory information (victim location) received from victims. After clustering the data, this component assigns priority values to each cluster based on the count of trapped, injured, or both trapped and injured victims (depending on the type of distress signal received by a surveying quadcopter). The clusters with higher priority more likely require immediate help. Furthermore, this component uses Dijkstra’s

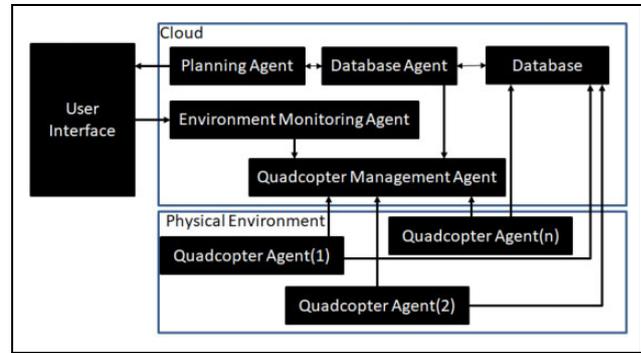


Figure 2. The proposed system architecture.

algorithm to calculate the shortest distances to the centroid of high-priority clusters, which helps rescue teams to plan their movements accordingly.

Blackboard consolidation

This component plays a central role in the proposed system. A central database is continuously updated by a group of quadcopters based on the respective data received by them. Information collected by quadcopters from their survey zones includes the location of the victim along with their status (injured, trapped, both, or safe), date and time, magnitude of earthquake, and ID of associated distress beacon based on the sender’s IMEI number. This information is accessible to all other agents in the proposed system for management and control purposes. Finally, control information, clustering results, and shortest paths are stored in the database to facilitate the movement of first responders in the disaster scenario.

Figure 2 depicts the flow of information between various components of the cloud-based multi-agent framework explained earlier. The user interface module is used to input the parameters signifying the occurrence of an earthquake in order to initialize the survey of a disaster-affected area. Conversely, upon completion of the survey, the results are sent back to the user interface for planning rescue activities accordingly.

Fault tolerance

One of the benefits of using an agent-based system is fault tolerance. When an agent fails, another nearby agent can take over its tasks. This failure can easily be identified through ad hoc communication or monitoring at the central server. In the proposed framework, the role of the central blackboard is critical as it is continuously controlling and monitoring the quadcopters. Therefore, the central server is deployed in high availability (HA) mode, where one server is designated as a master node and two servers act as secondary servers. The HA is defined as the ability that keeps system functioning after software or hardware failure.³⁸ In the case of master server failure, one of the secondary

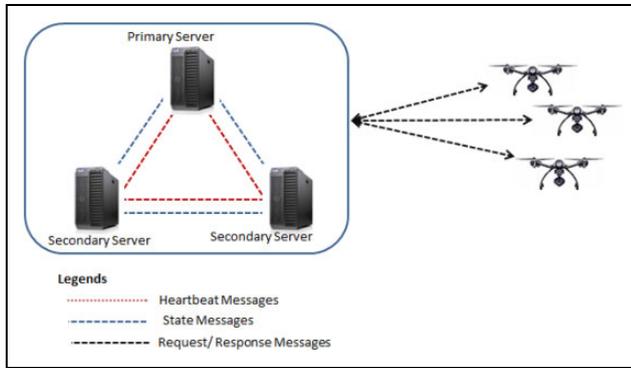


Figure 3. The fault tolerance deployment model.

servers takes its place and system continues its execution uninterrupted. The failure between the servers is detected through the exchange of heartbeat messages (as shown in Figure 3). Having multiple servers working in HA mode, one of the core issues is to keep the states of all the servers synchronized with each other. In our proposed approach, only the primary server can receive requests and generate a response or assign tasks to agents. However, to keep the states synchronized with other secondary servers, the primary server sends the received requests to all the secondary servers. After receiving a response from all, it generates a response or assigns a task to the quadcopter. In this way, all the servers are synchronized and in the case of failure, one of the secondary servers becomes the primary server. The decision of selecting a secondary server to replace the primary one is based on priority which is assigned at setup time. The fault tolerance deployment model is shown in Figure 3.

Fault tolerance inside the cloud environment

Inside the cloud, fault tolerance is managed in a similar fashion as mentioned earlier but automatically by the cloud providers. The cloud providers offered two types of recovery services, that is, fault and disaster tolerance. In disaster tolerance, the replica of services is hosted inside geographically distributed data centers. Should one of the data centers go down, the services can be accessed from a different region and this is achieved through domain name system (DNS). The main issue in deploying replicas across geographically distributed data centers is data synchronization. Most of the cloud providers offer either eventual or strong consistency. In the eventually consistent model, there is a chance that the secondary might not be up to date when failure occurs, whereas strong consistency ensures that all the sites are updated before generating a response to the user.³⁹ In our proposed framework, our focus is fault tolerance, which is achieved in Amazon EC2 through the load balancer and auto-scaling features. In this case, if one of the instances goes down, the load balancer automatically boots up another instance, thus keeping the number of instances to a predefined number.

In cloud deployment model, all the data are stored in a shared location. The quadcopters are connected with the central server through a smart device that provides cellular connectivity for communication with the server. The cellular network continues to operate even after a calamity. Moreover, with the integration of drones and the availability of cheap hardware, the creation of dynamic, small cellular networks to facilitate disaster areas is also possible where cellular network becomes unavailable.⁴⁰

Experimental results

In order to simulate a disaster environment following an earthquake event, a suitable simulation platform was required to obtain credible results. Table 2 presents the simulators that are considered good choices for performing agent-based simulations. Only the GAMA and Gazebo simulators support the multilevel agent-based modeling. Since GAMA supports data mining, a feature that is absent in Gazebo, it was our natural choice for performing the simulations. All the simulations for the validation of our proposed approach are performed in the GAMA simulator. Moreover, GAMA also provides a multi-scale control.¹⁷

GAMA (<https://github.com/gama/platform/gama/wiki>, Online; accessed 1 January 2018) software is used for implementing the prototype due to its distinctive features, including the support for a multilevel agent-based modeling with respect to time, behavior, and inherent support for data mining tools. Figure 4 shows the disaster site that has been modeled in GAMA. Based on readily available quadcopter prototypes, the key assumptions are made for the purpose of simulation, as listed in Table 3. The parameter values listed are based on the average of prominent UAVs, commercially available (<http://www.amazon.com>), including MJX Bugs 2W with a flight time of 18 min (<http://www.mjxrc.net/upload/201707/12/201707121030082363.pdf> Online; accessed 1 January 2018) and DJI Phantom 3 Pro has a flight time of 23 min (<https://www.dji.com/phantom-3-pro/info> Online; accessed 1 January 2018).

When an earthquake hits a particular region, the agents divide the area into an equal number of zones. This is shown in Figure 5, which is composed of an equal number of plots. Agents allocate quadcopters in different zones for data collection and control the movement of those quadcopters within the plots. The basic method for data collection is that quadcopters receive data (using Bluetooth low-energy beacons placed on them) sent by victims, using a disaster app installed on their android phones and this application also uses Bluetooth low-energy beacons for communication purposes. The data received by the quadcopters include information about the location of the victim (longitude, latitude, and altitude), status of a particular victim, date and time, magnitude of earthquake, and ID of associated distress beacon based on the sender's IMEI number.

Table 2. Comparison of various simulation tool kits, justifying selection of GAMA.

Simulation name	Agent-based simulation	Multilevel agent support	GIS	Data mining support	Three-dimensional visual support
Netlogo	Yes	No	Yes	No	Yes
MASON	Yes	No	Yes	No	Yes
Robocode	Yes	No	Yes	No	Yes
USARSim	Yes	No	No	No	Yes
Robocup	Yes	No	Yes	No	Yes
Gazebo	Yes	Yes	Yes	No	Yes
ABDiSE	Yes	No	Yes	No	No
GAMA	Yes	Yes	Yes	Yes	Yes

GIS: geographical information system.

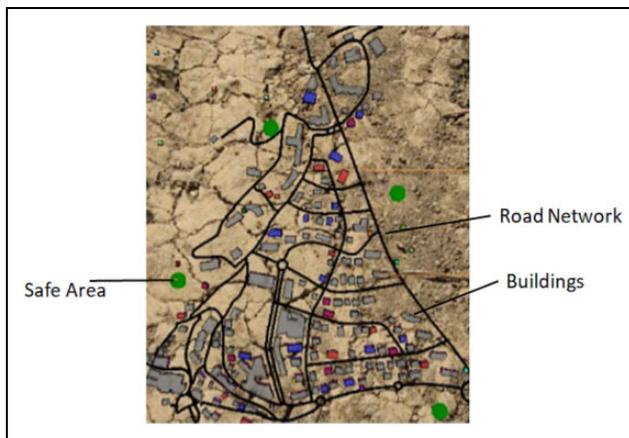


Figure 4. Disaster site modeled in GAMA. Gray: buildings, black: road networks, and green: safe areas.

Table 3. Simulation parameters.

Assumption	Values
Speed of quadcopter	10 m/s
Total flight time	20 min
Distance traveled during total flight time	12,000 m
Number of messages handled by single quadcopter	20 messages/s

During the survey, quadcopters are managed through a multi-agent system and the collected data are used in planning and to produce and prioritize clusters in order to identify people who require immediate help. Identification of victims needing immediate assistance can improve rescue efforts ensuring a timely response to those in need, which can increase the chances of survival. Figure 6 shows the simulation of the various degrees of damage that occur with different earthquake magnitudes.

Evaluation and results

As previously mentioned, we used GAMA to simulate a disaster-affected city. The area was divided into an equal number of zones to get information about the population in disaster-affected areas. Each zone is assigned a quadcopter for gathering information from it.

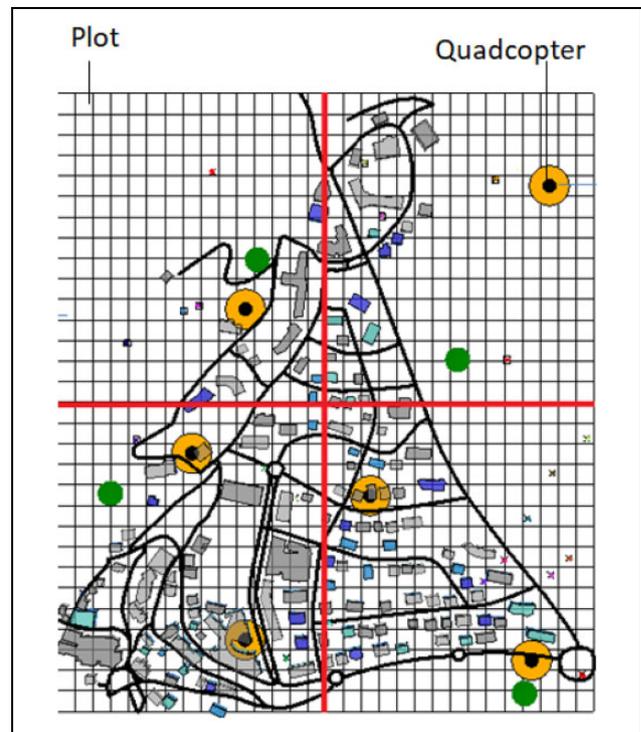


Figure 5. Quadcopters in different zones of disaster site.

Since different zones have different distress message rates owing to varying proportions of disaster-affected people, each zone is categorized as high, low, or medium based on the rate of distress messages received. Here, we have considered three cases. In the first case, all zones within the disaster site were kept high priority which means that the message rate of each zone is kept between 2500 and 6000. In the second case, all zones are marked as medium priority because the message rate of all zones is kept between 0 and 2500. Similarly, the third case involves all zones with low priority which means that the rate of all zones observes message rate between 0 and 1000. During the experiments, the message rate for each zone is recorded for a period of 20 min, and the management of quadcopters is done accordingly.

Once victims' data have been collected, trends at different earthquake magnitudes can be observed. In order to aid

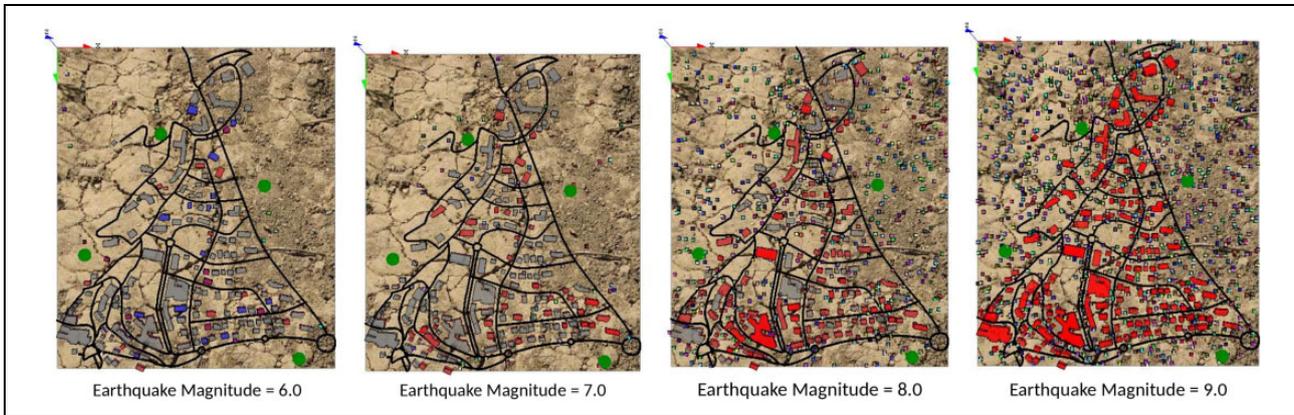


Figure 6. Degree of damage at different earthquake magnitudes.

timely rescue operations, clusters are formed within the affected population and the priority of each cluster is calculated based on the severity of devastation. Essentially, the clusters with high priority include people who require immediate help as compared to those with lower priority needs.

Evaluation criteria

The following two approaches are adopted in order to evaluate the management of collaborative robots using a cloud-based multi-agent system. Unstructured/uncertain environment due to varying message rates observed in each zone. Since our basic goal is to gather the maximum possible information about the affectees of an earthquake, we have compared the results using the following two approaches for assignment of quadcopters in all three cases mentioned earlier.

Fixed/equal assignment. In this approach, the number of available quadcopters is equally divided among the number of zones to locate victims without taking changes to the physical environment into consideration. For instance, assuming eight quadcopters are available then two are assigned to each of the four zones.

Reactive/intelligent assignment. Reactive or intelligent assignment of quadcopters is based on two components. First, the estimates of an initial number of quadcopters by considering previous data and second, autonomously increasing the number of quadcopters required to survey a particular zone if the incoming rate of distress messages exceeds the message handling threshold of the aerial surveillance drone. Furthermore, the performance of the designed multi-agent system was evaluated through deployment on a local machine and Amazon EC2.

Results

In Figure 7, the message rates of a particular zone are shown over a 20-min time period to analyze how successfully the multi-agent system developed for the cloud

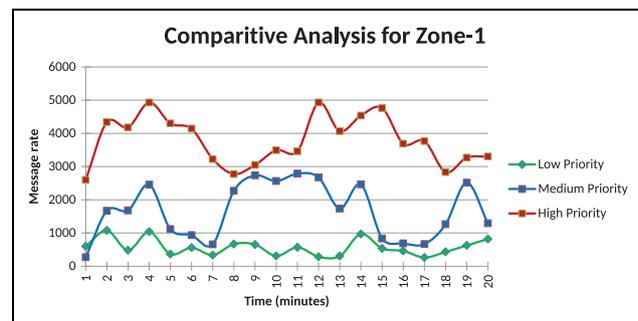


Figure 7. Message rates for zone 1.

responds to an uncertain environment. Based on the message rate, quadcopters are sent using two approaches fixed assignment (shown in red) and reactive assignment (shown in green), as shown in Figures 8 to 10. According to the assumptions made earlier, we know that one quadcopter has processing capacity to handle 1200 messages/min. It can be seen in the graphs that using fixed assignment, we have assigned two quadcopters in each zone which denotes that 2400 messages/s are being handled throughout a 20-min time frame in a particular zone. Alternatively, reactive assignment enhances the capacity of the system to handle more messages by increasing the number of quadcopters when the message rate increases. Essentially, our goal is to effectively manage a group of quadcopters, thereby locating the maximum number of victims. When zone 1 is set to a high message rate, the fixed assignment of quadcopters resulted in underutilization of resources (less than the required number of quadcopters was deployed for the survey), which led to loss of messages from trapped or injured victims. In comparison, reactive assignment in the proposed cloud-based multi-agent system increased the number of quadcopters directly in accordance with the incoming message rate, such that all distress calls from victims could be accurately recorded. Similarly, we can see under- and overutilization caused by fixed assignment when zone 1 is set to medium and low message rates, respectively. Figure 11 shows the comparative analysis of

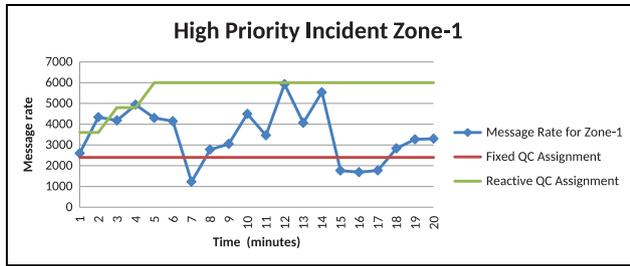


Figure 8. Quadcopter assignment in a zone with high message rate.

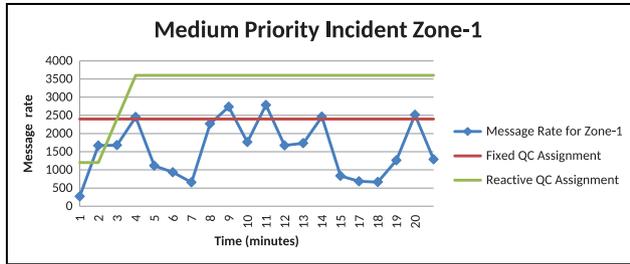


Figure 9. Quadcopter assignment in a zone with medium message rate.

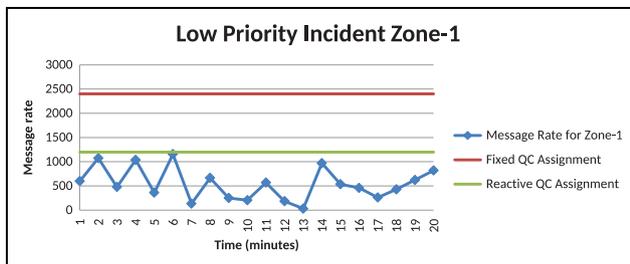


Figure 10. Quadcopter assignment in a zone with low message rate.

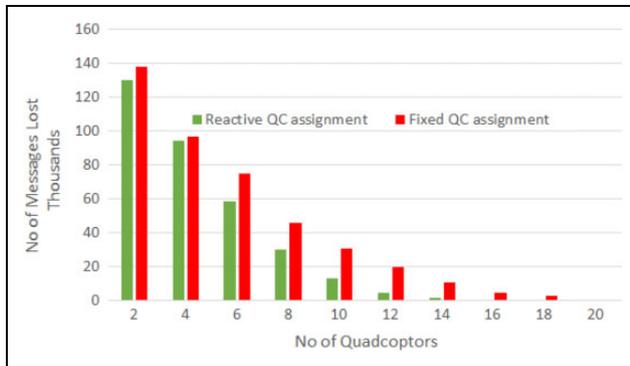


Figure 11. Comparative analysis of fixed versus reactive QC assignment.

fixed versus reactive QC assignment over the disaster site consisting of four zones. Each zone is randomly assigned a high, medium, or low message rate. The available quadcopters are assigned to different zones using fixed and reactive assignment. The simulation is executed for 20 min

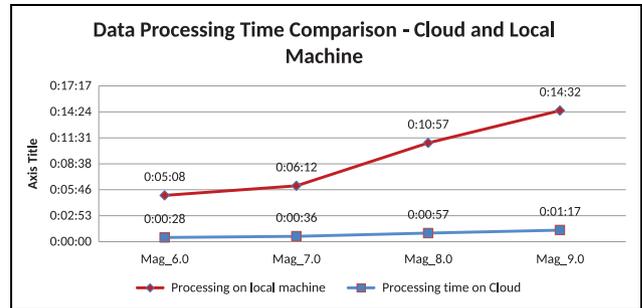


Figure 12. Performance analysis.

and the number of messages lost from all four zones is recorded for both the fixed and reactive QC assignments. The experiment is repeated 1000 times and the average number of messages lost for a different number of quadcopters available is depicted in the figure. The figure shows that the number of messages lost per quadcopter using reactive assignment is about 5% less than the fixed assignment. As a result, by using more than 14 quadcopters, no messages are lost in reactive assignment unlike fixed assignment. Furthermore, the number of quadcopters required to record all messages is less for reactive assignment than for fixed assignment, hence demonstrating efficient management of available resources. By requiring fewer quadcopters to cover the disaster site, reactive QC assignment also helps in reducing the computing resources that are necessary. Therefore, the results prove that the intelligent management of a group of quadcopters based on sensory inputs is highly beneficial in surveying the unstructured environment resulting from a disaster situation. Furthermore, the performance of the system is also analyzed when using a laptop with core i3 processor and Amazon EC2 services. Figure 12 shows that the proposed system takes less processing time on the cloud and has linear time complexity. In contrast, on the local machine, the simulation not only takes more time but also increases as a quadratic function of earthquake magnitude.

Conclusion and future work

This article research endeavor is focused toward the design and development of a multi-agent framework for the intelligent management of collaborative flight drones or quadcopters, such that they can carry out a quick yet effective survey of a disaster site in the aftermath of earthquakes for which early warning systems are still ineffective. The proposed system can help to timely assess the scale of damage such that immediate rescue efforts can be planned to prevent the loss of human life as a first priority.

In order to accomplish this, various approaches toward the design of multi-agent systems are surveyed to determine the methodology that circumvents most architectural shortcomings. Here, key assumptions are made regarding the speed, battery life, and message handling

capacity of a flight drone in order to develop novel algorithms for planning and prediction, thereby determining an adequate number of quadcopters to survey a disaster-affected area in minimal time. Additionally, the simulated results from the survey are used to identify affected communities such that limited rescue resources could be coordinated in the most effective manner to provide immediate assistance to victims.

Subsequently, the developed system has been repeatedly tested using fixed and variable number of quadcopters for surveying a disaster site for different earthquake magnitudes (simulated using the GAMA platform). Based on the results, the designed system is able to evaluate the input received in real time from the simulated environment and accordingly determine an adequate number of quadcopters for the survey.

The use of the Amazon EC2 platform in this context helped ensure the reliability of the developed system in terms of avoiding a single point of failure and provide features like auto-scaling. Through effective use of cloud services, the developed system can be accessed (after necessary authentication procedures) from anywhere in the proximity of the disaster site through Internet connection.

Based on the conclusions drawn thus far, future efforts may be concentrated on the realization of the scenarios simulated in this study pertaining to the use of actual quadcopters to conduct mock surveys in the subsections of real cities. This can invariably assist in the validation of experimental results. Moreover, the cloud-based multi-agent frameworks for managing robots in the physical world can also be used for structural inspection for dilapidated homes, remote monitoring of farms to reduce crop damage, locating dengue mosquito/larval habitats, and other useful applications.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Yan Z, Jouandeau N, and Cherif AA. A survey and analysis of multi-robot coordination. *Int J Adv Robot Syst* 2013; 10(12): 399.
2. Ferrer EC. The blockchain: a new framework for robotic swarm systems. DOI: abs/1608.00695
3. Gzel MS, Kara M, and Beyazkl MS. An adaptive framework for mobile robot navigation. *Adapt Behav* 2017; 25(1): 30–39.
4. Nikolaidis S, Hsu D, and Srinivasa S. Human-robot mutual adaptation in collaborative tasks: models and experiments. *Int J Robot Res* 2017; 36(5–7): 618–634.
5. del Pobil AP, Kassawat M, Duran AJ, et al. UJI RobInLab's approach to the Amazon Robotics Challenge 2017. In: *2017 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)*, Daegu, South Korea, 16–18 November 2017, pp. 318–323. Daegu, South Korea: IEEE.
6. Serugendo GDM, Gleizes MP, and Karageorgos A. Self-organization in multi-agent systems. *Knowl Eng Rev* 2005; 20(2): 165–189.
7. Michael S. *Objective coordination in multi-agent system engineering: design and implementation*. Berlin, Heidelberg: Springer-Verlag, 2001. ISBN: 3-540-41982-9.
8. Talia D. Clouds meet agents: toward intelligent cloud services. *IEEE Int Comput* 2012; 16(2): 78–81.
9. Mouad M, Lounis A, Pierre S, et al. Control architecture for cooperative mobile robots using multi-agent based coordination approach. In: *6th national conference on control architectures of robots*, Grenoble, France, May 2011, pp. 15. INRIA Grenoble Rhône-Alpes.
10. Ren W, Beard RW, and Atkins EM. A survey of consensus problems in multi-agent coordination. In: *Proceedings of the 2005, American control conference*, Vol. 3, Portland, OR, USA, 8–10 June 2005, pp. 1859–1864. IEEE.
11. Olsson PM, Kvarnström J, Doherty P, et al. Generating UAV communication networks for monitoring and surveillance. In: *2010 11th international conference on control automation robotics vision*, Singapore, Singapore, 7–10 December 2010, pp. 1070–1077. IEEE.
12. Chen D, Zhixin L, Lizhe W, et al. Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems. *Mobile Netw Appl* 2013; 18(5): 651–663.
13. Soler J, Julián V, Carrascosa C, et al. Applying the ARTIS agent architecture to mobile robot control. In: Carolina MM and Simão SJ (eds) *Advances in artificial intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 359–368.
14. Hayes-Roth B. A blackboard architecture for control. *Artif Int* 1985; 26(3): 251–321.
15. Mosterman PJ, Sanabria DE, Bilgin E, et al. A heterogeneous fleet of vehicles for automated humanitarian missions. *Comput Sci Eng* 2014; 16(3): 90–95.
16. Ezequiel CAF, Cua M, Libatique NC, et al. UAV aerial imaging applications for post-disaster assessment, environmental management and infrastructure development. In: *2014 International conference on unmanned aircraft systems (ICUAS)*, Orlando, FL, USA, 27–30 May 2014, pp. 274–283. IEEE.
17. Galindo C, Cruz-Martin A, Blanco JL, et al. A multi-agent control architecture for a robotic wheelchair. *Appl Bion Biomech* 2006; 3(3): 179–189.
18. Galindo C, Gonzalez J, and Fernandez-Madrigras JA. An architecture for cognitive human-robot integration. Application to rehabilitation robotics. In: *IEEE international conference mechatronics and automation*, Vol. 1, Niagara Falls, Ontario, Canada, 29 July–1 August 2005, pp. 329–334. IEEE.

19. Wong CC, Cheng CT, and Chan HM. Design and implementation of an autonomous robot soccer system. *Int J Adv Robot Syst* 2013; 10(5): 228.
20. Patrick T, Duc-An V, Edouard A, et al. Gama: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In: Nirmitt D, Alan L and Michael W (eds) *Principles and practice of multi-agent systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 242–258.
21. Fierro R, Das A, Spletzer J, et al. A framework and architecture for multi-robot coordination. *Int J Robot Res* 2002; 21(10–11): 977–995.
22. Nieto-Granda C, Rogers JGR III, and Christensen HI. Coordination strategies for multi-robot exploration and mapping. *Int J Robot Res* 2014; 33(4): 519–533.
23. Du Y, de Silva CW, and Liu D. A multi-agent hybrid cognitive architecture with self-awareness for homecare robot. In: *2014 9th international conference on computer science education*, Vancouver, BC, Canada, 22–24 August 2014, pp. 223–228. IEEE.
24. Crooks AT and Sarah W. GIS and agent-based models for humanitarian assistance. *Comput Environ Urban Syst* 2013; 41: 100–111.
25. Chen F, Zhi Z, and Greg M. Dynamic adaptive disaster simulation: developing a predictive model of emergency behavior using cell phone and GIS data. In: *Proceedings of the 2011 workshop on agent-directed simulation*, ADS '11, Boston, USA, 3–7 April 2011, pp. 5–12. Society for Computer Simulation International.
26. Boston M, Liu Z, Jacques C, et al. Towards assessing the resilience of a community in seismic events using agent based modeling. In: *NCEE 2014—10th US national conference on earthquake engineering: frontiers of earthquake engineering*, Alaska, USA, 2014, pp. 10.
27. Frank F and Paul B. Agent-based systems for disaster management. *Commun ACM* 2007; 50(3): 41–42.
28. Okaya M, Yotsukura S, Sato K, et al. Agent evacuation simulation using a hybrid network and free space models. In: Jung-Jin Y, Makoto Y, Takayuki I, et al. (eds) *Principles of practice in multi-agent systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 563–570. ISBN: 978-3-642-11161-7.
29. Coates G, Hawe G, Wilson DW, et al. Adaptive co-ordinated emergency response to rapidly evolving large-scale unprecedented events (rescue). In: *International ISCRAM Conference*, Lisbon, Portugal, 2011, pp. 1–5.
30. Currión P, Silva C, and Van de Walle B. Open source software for disaster management. *Commun ACM* 2007; 50(3): 61–65.
31. Su Y, Yang L, and Jin Z. Simulation and system dynamics models for transportation of patients following a disaster. In: *2008 International workshop on modelling, simulation and optimization*, Hong Kong, China, 27–28 December 2008, pp. 93–96. IEEE.
32. Wada A, Yamashita T, Maruyama M, et al. A surveillance system using small unmanned aerial vehicle (UAV) related technologies. *Special Issue Solv Soc Issue Thro Bus Act* 2013; 8: 68–72.
33. Ueyama J, Freitas H, Faical BS, et al. Exploiting the use of unmanned aerial vehicles to provide resilience in wireless sensor networks. *IEEE Commun Mag* 2014; 52(12): 81–87.
34. Petrovic T, Haus T, Arbanas B, et al. Can UAV and UGV be best buddies? Towards heterogeneous aerial-ground cooperative robot system for complex aerial manipulation tasks. In: *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, Vol. 01, Colmar, France, 21–23 July 2015, pp. 238–245. IEEE.
35. Erdelj M, Natalizio E, Chowdhury KR, et al. Help from the sky: leveraging UAVs for disaster management. *IEEE Pervas Comput* 2017; 16(1): 24–32.
36. Othman SB, Zgaya H, Dotoli M, et al. An agent-based decision support system for resources' scheduling in emergency supply chains. *Control Eng Prac* 2017; 59: 27–43.
37. Tobias W, Dominik R, and Henrich D. Design and evaluation of a multi-agent software architecture for risk-minimized path planning in human-robot workcells. In: Schüppstuh T, Jörg F and Kirsten T (eds) *Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 103–112.
38. Riley J, Noss J, Dillingham W, et al. A high-availability cloud for research computing. *Computer* 2017; 50(6): 92–95.
39. Anderson JW, Meling H, Rasmussen A, et al. Local recovery for high availability in strongly consistent cloud services. *IEEE Trans Depend Sec Comput* 2017; 14(2): 172–184.
40. Hayajneh AM, Zaidi SAR, McLernon DC, et al. Drone empowered small cellular disaster recovery networks for resilient smart cities. In: *2016 IEEE international conference on sensing, communication and networking (SECON Workshops)*, London, UK, 27–27 June 2016, pp. 1–6. IEEE.