A Multi-Agent Framework for Cloud Based Management of Collaborative Robots

By
Tooba Samad
NUST201362739MSEECS60013F

Supervisor
Dr. Sohail Iqbal
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

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

(May 2016)

Approval

It is certified that the contents and form of the thesis entitled "**A Multi-Agent Framework for Cloud Based Management of Collaborative Robots**" submitted by **Tooba Samad** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Sohail Iqbal**

Signature: ____________________

Date: ____________________

Committee Member 1: **Dr. Asad Waqar**

Signature: ____________________

Date: ____________________

Committee Member 2: **Dr. Omar Arif**

Signature: ____________________

Date: ____________________

Committee Member 3: **Mr. Shamyl Bin Mansoor**

Signature: ____________________

Date: ____________________

Abstract

The field of multi-agents presents highly efficient solutions to problems that may be too complex for single artificial agents to address effectively. By simply breaking down a complicated agent function into simpler subfunctions that can be coordinated either centrally or in a distributive environment, multi-agent systems can be applied to a wide array of problem domains. One such domain is robotics where autonomous decision-making is imperative and simultaneous learning functions are significantly gaining importance. Therefore, the combination of the aforementioned disciplines of agents and robotics provides unique opportunities in terms of achieving advanced functionalities for wide variety of applications.

Conversely, this arrangement results in a significant number of challenges as well. For instance, robotics generally involves hardware limitations that can prohibit the design of the multi-agent system. A similar case is explored here where a multi-agent system is developed for highly maneuverable quadcopters intended for the possible surveillance of a particular area for emergency evacuation purposes in case of disaster. However, the limited flight times and load bearing capacity of the quad copter present major hurdles towards the architectural design.

The basic idea remains bridging the cyber world to physical world in the most efficient way. Therefore the designed multi-agent system manages group of robots in an unstructured environment resulting from catastrophic happenings such as earthquakes. Hence, the robots are responsible for surveying a disaster affected area and locating the maximum number of trapped and/or injured people as part of the immediate response. The major feature of proposed architecture is that it effectively manages group of quad copters within a cloud framework for increased reliability and ease of accessibility.

The proposed multi-agent system for cloud base management of collaborative robots has been tested using simulation, Several experiments with varying earthquake magnitude and corresponding scale of devastation have been carried out; whereby the results show that proposed architecture effectively manages group of robots given an unstructured environment.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Tooba Samad**

Signature: ________________

Acknowledgment

Beginning with Allah Almighty, I would like to express my sincere gratitude to my supervisor Dr. Sohail Iqbal for his constant encouragement and guidance in this research endeavor; and to Dr. Peter Bloodsworth for suggesting the initial direction for research and precious comments throughout this time period.

I also would like to thank all my committee members Dr. Omar Arif, Dr.Asad Waqar and Mr. Shamyl bin Mansoor for their kind support and suggestions whenever required. I am also grateful to my colleagues Ghalib Ahmad Tahir, Mansoor Rehman, Mehreen Mushtaq and Naveed Iqbal who have always been a source of great help and encouragement for completion of this thesis.

In the end, I would like to thank my parents for giving me this opportunity and for supporting me from miles away. I would like to especially praise my father who has always shown interest in my work, discussing my thesis progress regularly. Thank you for being there giving light and happiness to my life.

Dedication

With immense gratitude, I would like to dedicate this thesis to my parents and my teachers Dr.Sohail Iqbal and Dr.Peter Bloodsworth, for their guidance,endless support and motivation because this was impossible without them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter gives the basic introduction of the concepts involved in this research. It also presents the background and motivation for this study. Finally, it gives a brief overview of the working methodology for proposed system, and also includes the structure of the remaining thesis document.

1.1 Introduction

Over last few decades, technology has evolved incredibly leading to the development of smarter systems. Still today, there exist several situations where human beings are unable to effectively accomplish complicated tasks even with the aid of technology. For instance, building evacuation in case of an emergency, aerial monitoring of farms to reduce crop damage, locating habitats of dengue mosquito for their elimination and much more. The development of smarter systems to address these problems is not possible with the use of one type of technology alone, rather it is achievable with the combination of several technologies like internet, bluetooth, robotics and cloud computing.

In the above mentioned circumstances, multi-robot systems are considered an advantageous approach in contrast to single robot systems because collaboration between multiple robots helps in the efficient accomplishment of pre-defined goals. Thus far, a huge amount of research towards providing intelligence to robots deals with domains which are purely synthetic such as the chess board game or with toy worlds. Therefore, they tend to avoid real world problems which require multi robot systems to perceive the world around them along with associated intelligent actions. For this purpose multi-agent systems may prove to be increasingly beneficial, as they have the capability to manifest self-organization and self-steering and other control paradigms.

These are achieved through sharing of knowledge via common language platform within the constraints of specified system [1]. 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 need for detailed re-writing [2]. Hence, these systems are self-recovering and failure proof, as the components are quite redundant and possess self-managing features.

The combination of the paradigms of agents and robotics provides unique opportunities to achieve advanced functionalities for applications mentioned above. However, such systems require huge amount of processing power and storage costs. Therefore, the incorporation of cloud computing technology into the existing scenario holds the key, and the integration of multi-agent systems and cloud computing can result in high performance of systems as well as development of complex intelligent applications [3]. These applications can sense things from real world and also alter them remotely. In this research endeavor, a multi-agent system has been designed and developed which bridges cyber world to physical world – specifically in terms of remotely controlling a small group of robots (quad-copters) in physical environment by utilizing cloud computing services owing to greater flexibility provided by the distributed infrastructure.

1.2 Motivation

Despite continuous efforts being made in field of robotics to make such systems more sophisticated in order to match human capabilities, they often face shortcomings when it comes to intelligent control and collaboration. This is because their performance is bound to detailed set of instructions. In order to embed intelligence in multi robot system and make them capable enough to carry out assigned tasks they must:

- Perform goal oriented instructions.
- Understand environment they operate in.
- Develop a plan of actions they intend to perform.

Moreover multi-robot systems with limited resources require considerably proficient computing resources which increase the cost of underlying infrastructure. This restricts the installation of multi-agent system on top of a robot control mechanism as it would only result in additional power requirements for a resource limited robot (quad-copter). Therefore, such systems mostly end up with their software residing in a centralized machine or server which makes them prone to a lack of robustness. In order to work address these concerns, cloud services can be utilized to bear intensive computing workloads and the designed multi-agent system can be effectively deployed over them.

1.3 Problem definition

The aim is to design and develop a multi-agent architecture which bridges cyber world to physical world specifically in terms of controlling a small group of robots (quad-copters) in physical environment. Moreover, implementing this multi-agent system over a cloud framework which offers greater flexibility; with both soft real-time and physical constraints effectively managed within the distributed infrastructure.

1.4 Challenges

Currently, the greatest challenge is to integrate multi-agents, robotics, and cloud computing together, and making them work for a single system. . This makes the design and implementation of such systems a difficult task. The next major obstacle is addressing the limitations of physical robots in terms of storage, processing power and battery life. Also, coordinating a group of robots with limited resources and dealing with complex dynamic environments in soft real time further adds to the complexity of the overall system; as every robot with conflicting goals may interfere with the actions of others [4].

1.5 Research hypothesis and questions

Research hypothesis and questions were developed on the basis of problem definition and challenges mentioned earlier. In this section research hypothesis and questions are presented.

1.5.1 Research hypothesis

A multi-agent system deployed over the cloud is able to effectively manage collaborative robots in a physical environment.

1.5.2 Research questions

- How are multi-robot systems being controlled currently?
- How to build multi-agent system which can integrate robotics and cloud technology together?
- How successfully the proposed system can manage a multi-robot system?

1.6 Methodology

The purpose of this research exercise is to develop a multi-agent architecture for collaborative flight drones to be used as immediate response damage assessment tools in the event of an earthquake based on the following presumptions:

- Non-functioning of mobile networks – mostly due to traffic channel blocking when the number of subscriber requests exceeds network capacity.
- Lack of internet connectivity due to absence of internet infrastructure/mobile internet.
- Need for alternate means of communication using existing mobile devices which consume less energy.
- Using a basic mobile application, an SOS beacon can be continually transmitted via bluetooth low energy from most smart phone devices available today.

Therefore, collaborative flight drones can be programmed to detect such beacons through sensors, and rescue operations can be planned based on incoming patterns of information. The developed multi-agent architecture will be deployed over cloud framework because limited processing power of drones restricts the deployment of multi-agent system on top of primary flight control mechanism; as this would not only trim the battery life of drone but also effect the overall flight time.

1.7 Objectives

The objectives for a multi-agent architecture deployed over cloud framework designed to coordinate multi-robots system operating in physical environment in soft real time can be summarized as:

- Dealing with complex situation that may arise in the aftermath of catastrophic events such as earthquakes which require coordinated problem solving.
- Control and management of multi-robot systems.
- Addressing constraints of the above mentioned multi-robot systems such as limited battery life, inability to handle intensive computing workloads, and insufficient processing power to name a few.

1.8 Thesis outline

The remaining content of this thesis dissertation is outlined as follows:

- Chapter 2: Background and Related work. This chapter discusses existing multi-agent architectures for robots in terms of their salient features. All systems studied for this research exercise have been analyzed and compared on basis of identified features.
- Chapter 3:Proposed Multi-Agent Architecture.This chapter describes multi-agent architecture developed for cloud framework to manage collaborative robots.
- Chapter 4: Experimental Setup. This chapter highlights implementation details of proposed multi-agent architecture.
- Chapter 5: Experimental Results. This chapter presents results based on the comparison drawn with existing multi-agent architecture using simulation. Furthermore, the management of collaborative robots is tested in disaster scenarios of varying intensity.
- Chapter 6: Conclusion and Future Work. This chapter concludes the thesis by summarizing the research and proposes directions for future work.

Chapter 2

Background and Related Work

This chapter gives brief overview of major types of multi-agent architectures to control robots and in the context of disaster management systems. Moreover, it also discusses various simulation platforms used by different disaster management applications.Finally, a comparative analysis of the systems studied for this research exercise has been carried out based on the salient features of such systems.

2.1 Types of multi-agent architectures

Essentially, there are five main approaches to designing multi-agent architectures for control software [4]. These can be briefly stated as follows:

i Reactive vs. deliberative
Architectures which fall in reactive category have modules connected to several sensors and actuators, and implement a behavior to provide immediate response to a stimulus. As opposed to this, deliberative behaviors are not instant by design; whereby the response to each stimulus is rationalized keeping in view a set of pre-defined conditions. Therefore, a choice among the two designs is heavily dependent on the intended application.

ii Hierarchal vs. non-hierarchal
Hierarchal designs present a layered architecture where upper layers are dedicated for decision making, intermediate layers deal with supervision and control, and repetitive methods like actuator control are handled by lower layers. Like each layered architecture, hierarchal designs suffer from similar drawbacks; whereby each upper layer is dependent on the accurate functioning of the lower layer in case of one directional flow of

information. For bi-directional flows, both upper and lower layers are equally dependent on one another. Alternately, non-hierarchal architectures face shortcomings in terms of a lack of modularity and decentralized control.

iii Hybrid Architectures
Hybrid architectures are a combination of above mentioned architectures that are basically composed of:

- Deliberative layer for task planning
- Control layer
- Functional layer for immediate response to stimulus.

iv Centralized System
Such systems include a supervisor agent that takes over the leadership role and is responsible for organizing and planning the work for whole team. Moreover, it is also in charge of making decisions for whole team, whereas all other members will only act as per instructions relayed to them. This approach however, suffers from a singular point of failure.

v Distributed System
In contrast to a centralized approach, a distributed system is a collection of autonomous agents which collectively take part in the decision making process.

2.2 Multi-agent systems to control robots

There are several different multi-agent systems to control multiple robots, and follow different multi-agent architectures based on their application needs. As some applications need to work in hard real time in non-deterministic and dynamic environments, they take benefit of multi-agent architecture like ARTIS [5] which allows modelling of agents within hard real-time constraints and lets them control environment through set of sensors and effectors in order to generate a response within hard temporal restrictions. Essentially, ARTIS is an extension of the blackboard model [6] modified to meet hard real-time constraints, and it is comprised of two levels of agent architectures, i.e. ARTIS agent (reactive) and in-agent (deliberative) thereby making it a hybrid architecture. It is however centralized, due to which it is unable to support distributed robot applications.

Similarly, the IDEA architecture [7] is 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. Another architecture proposed by WuXing et al[8] is based on Prometheus methodology which supports 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 Prometheus methodology and is used to analyze system functionality, build operation scenarios and design the MAS coordination architecture after defining the types of agents through role-agent mapping.

Some of the existing architectures used in different applications are modification of previously proposed designs such as architecture for controlling robotic wheel chair proposed by GalindoC. et al [9], which is both a modification and extension of previous existing ACHRIN [10] (hybrid) architecture The components in ACHRIN are designed in client-server fashion, which makes modification and expansion a difficult task. However, this was accomplished through incorporation of Multi-agent system and implementing each component of ACHRIN as separate autonomous agent. The designed agents have their own beliefs and intentions, as well as the ability to learn different methods and appropriate skills autonomously to perform their actions.

Several systems require multi-robot coordination and cooperation to achieve a particular goal in games like soccer[11]. For this purpose, layered architectures have been proposed to facilitate coordination and cooperation of a team of robots where each layer provides services to the upper layer. In this scenario, the controlling multi-agent system resides on a local server which gets information about field through visual sensors, makes decisions according to the information received and then 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 this case, a hybrid architecture which takes into account the advantages of hierarchal and reactive architectures to minimize the response time and carry out planning tasks in order to achieve coordination through distributed and centralized aspects is highly preferred [4]. Such architecture essentially deals with coordination of multiple robots in an uncertain/unstructured environment and is comprised of physical, control and coordination layers. It uses multi-agent system models to handle cooperation/coordination issues within a team of robots to enhance the total utility of system, and the approach used for coordination of robots is centralized.

Another framework which deals with multi-robot coordination proposes a paradigm for managing cooperative behavior of finite state robots [12]. It focuses on methodologies regarding ease of control for application specific robotic systems. Unfortunately, the proposed paradigm works indoor under structured environment and fails to address the tasks involving communication and shared information. Coordination strategies presented in [13] show experiments to explore various collaboration strategies used by group of mobile robots to map and explore uncertain/unstructured environment. The approach is based on shared resources which are distributed among 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.

Complex applications often require distributed problem solving. Therefore, architecture for a mobile robot is presented in [14] to facilitate an autonomous robot with its complex behavior. However, this architecture is for a single robot and does not address the coordination issues which arise when working with multi-robot applications.

2.3 Disaster management systems

As goal of this research exercise is to manage multiple robots (quadcopter) in the wake of catastrophic events such as earth quakes, the developed system is required to be reactive and highly responsive. Also, it should have support for efficient decision making, action coordination and enhanced data collection. In this context, we have divided existing disaster management systems in following two broadly categories:

- Agent based Disaster Management Systems.
- Conventional Disaster Management Systems.

2.3.1 Agent based systems

- Humanitarian relief through crowdsourcing
  The system explored first presents a prototype to illustrate a post disaster scenario for better planning and coordination of humanitarian relief efforts [15]. It relies on agent based modeling and geographic information obtained from crowd movements, along with other publically available data sources. The developed prototype simulates a scenario where victims of an earthquake search for food. The study has been primarily based on a magnitude 7.0 earthquake that hit Haiti back in 2010, with the purpose to determine how humanitarian relief efforts can timely reach victims especially at times of great uncertainty. The key challenge under such unprecedented circumstances lies in predicting a

crowd's movements beforehand given sudden changes in infrastructural landscape – for instance the destruction of existing land routes.

Major novelty of the system is that it integrates raster collected from several sources and vector data structures in the same simulation. Moreover this model assumes that agents in most affected areas have greatest needs. Based on this, the goal of agents is to maximize their energy and agents near relief distribution points have detailed knowledge of their location which is then shared with other agents via diffusion method. The relevant research work successfully utilizes agent based modelling and geographical information to facilitate rescue efforts, however the crowdsourced component of information used by this system lacks structure and proper format since reliability on people in times of panic may lead to inaccurate or partially accurate information. Instead of comprehensiveness and accuracy, the system tends to focus on characteristics like timeliness and accessibility of information; because during chaotic situations when food is set as primary goal, being the basic human instinct analyzing the rapid spread of information helps to ascertain the most effective plans for distribution of relief goods. Furthermore, modelling people's behavior as an agent based system results in more realistic conclusions because adaptability is an inherent quality of intelligent agents.

- Dynamic adaptive disaster simulation
  Predicting movements of a large population in times of disaster is a complex task especially when making real-time decisions to carry out rescue efforts could prevent excessive human and economic loss. One such system is also proposed by Francis Chen et al [16] to predict movement of people using cellular and GIS data, which continuously updates simulation with real-time data by tracing movements of cell phone users. Inferences have been integrated into agent based simulation to determine future movements of people based on principles of fluid dynamics. Due to unavailability of real cell phone data owing to privacy concerns, synthesized data having similar properties has been used.

  Once the data set has been generated, system infers those areas which people prefer or avoid, for predicting movements. The entire affected region is modelled as a synthetic elevation field with differences in gradient representing areas that are comparatively more or less affected, Disaster sites or high-risk locations are termed as "high elevation" whereas safer localities are represented as defined as "low elevations". Therefore, in case of a disaster, the movement of people would invariably be towards areas of lower elevation. The work presented in this paper tries to address two main challenges i) Developing components of simulation system ii) generating data for simulations. Next, the validation of the results has been done using Manhattan distance technique to show generated predictions are valid, both internally and predictively. Moreover, the predictive model has been shown to conform to synthetic model in terms of evaluating areas of high and low elevations, which makes a strong case for the system's accuracy. 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 modelling for assessing resilience of a community in seismic events
  Similar to the previous system, agent based modeling techniques can also be used to assess resilience of community in the event of a major disaster [17]. The system explored here is based on a virtual city which includes major elements such as different types of buildings, road networks, population and other infrastructure. The scenario of a seismic event is used as input to show damage sustained and interactions between people predisposed by shock and panic.

  In addition to this, disaster alleviation plans and policies can be integrated in this virtual environment to evaluate their effectiveness in terms of preventing human and economic loss. The virtual city uses data presented in ATC 58 and requires user preferences (no. of children, residents, employed and unemployed) for simulation. Upon initialization, a selected percentage of people are distributed throughout the city to various locations depending on their occupation type, time of day and day of the week. Next, to simulate movements of people and vehicular traffic, several navigation algorithms had to be developed. The resulting system then has two modes; "default mode" where people follow routine behaviors – and "emergency mode" for when a disaster occurs. Once the system switches to emergency mode and people are unable to carry on with their routine activities, the emulated change in behavior can be monitored for valuable insights. Furthermore, while the usefulness of emergency drills for training people in terms of how to react in case of a disaster cannot be argued; the work presented by Boston et al. can be used to run multiple simulations. Thereby, a significant amount of time and human effort is saved while useful observations are made.

- Agent Evacuation Simulation Using Hybrid and Free Space Models
While modeling human behavior during a typical situations remains a difficult task, efficiently responding to disasters under limited time and resource constraints is crucial to prevent the loss of lives, property and livelihoods. In this respect, agent based systems provide a paradigm to simulate human behavior in environments created by real world or synthetic data because it is important to study human and traffic movements within affected area [18]. For this purpose, Masaru Okaya et al [19] have proposed a hybrid model to simulate population's movement through large number of agents in affected areas without utilizing intensive computational resources. The model is capable of simulating behavior of agents in wide areas through network, free- space and integrated models. For simulation purposes, they have considered a scenario where a disaster occurs when students are attending their lectures within inside a college campus. Based on this, the movements of rescue teams and students present inside, outside, and around the building entry and egress points have been emulated.

  According to simulation results, the Hybrid model is significantly cost effective in terms of computational resources because it interleaves between the computationally intensive free-space model and the less complex network model to illustrate maps. Furthermore, the evaluation of results carried out by comparing simulation times and wall clock time taken by agents to reach destinations, average times required by student and rescue team agents to move respectively and arrival times of 300 and 900 student agents in all three models (free space, network and combined one) supplement the viability of overall design. The data being used for simulation purposes can be considered both comprehensive and accurate as it combines structured data from two computational models and corroborate with actual simulation. Moreover, the efficiency of the hybrid simulator makes the information readily accessible as less time is required to determine agent behavior and results can be shared easily.

- Rescue Management System
Another proposal for rescue management by Kei Sato et al [20] presents a system composed of grass root components for various levels of disaster management which were developed individually for different purposes. First, an ERM system identifies certain areas to simulate the effects of disaster. Next, disaster situations are constructed in selected

areas using agent based simulation techniques, and the results of simulation are visualized on web using 2D or 3D models. Finally, emergency responders identify victims in worst affected buildings by switching to 'House size' simulations and plan out rescue efforts. The simulations used in this exercise are based on GIS data from governmental institutions and 3D data created using Sketch Up. This information is then superimposed over Google Earth and integrated using different synchronization techniques. Finally data from real world is linked with locations on a map. Thus the developed system is able to provide latest update on the status of a disaster, and also allows people to upload pictures and reports in real-time.

The system can also prove to be advantageous in terms of disaster management as data used by ERM is comprehensive. However, since there is allowance for data to be uploaded from diverse sources, therefore chances are that the data present in various formats might contain irrelevant and inaccurate information which also gets dumped into the database. This can significantly inflate the cost of storage required. Alternatively, the proposed system does not compromise on the timeliness of response because of the inherent capability of switching to the detailed 'house size' simulation which allows emergency responders to exactly locate the victims trapped inside damaged buildings. Furthermore, it is also ensured that all information and simulations remain accessible to the relevant authorities over the web interface.

- RESCUE
  Ideally, emergency response requires pre-planning 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, an integrated framework [21] has been developed with decision support and agent based simulation to help practitioners related to disaster management, plan effectively for emergency response.

  The proposed system helps construct real-time plans based on the characteristics of disaster, and also ensures timeliness and appropriateness of such plans through flexible search algorithms based on opportunistic methods. This reduces the complexity by analyzing effective units and corresponding emergency response. In addition to decision support, the agent based simulation under development has been designed with the objective to show how a dynamically constructed rescue team coordinates emergency response activities. The approach illustrated here has been proven to be quite successful in terms of formulating a timely response because of significant computational efficiency; therefore resulting operational models for rescue efforts can be shared readily. Moreover, 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.

2.3.2 Conventional systems

- SAHANA

  SAHANA [22] is an example of more conventional open source software developed for disaster management and relief efforts. As major disasters like tsunamis and earthquakes require accurate and timely information for effective decision making and carrying out rescue efforts to curb the scale of destruction, the faster the humanitarian community is able to gather, analyze and distribute reliable information, the more effectual the response would be. Essentially, the developed system is composed of different components designed to address specific coordination problems through communication via shared databases that provide access to web based information.

  Major features of this system are that it improves consequential gaps in information required for efficient response and can be scaled up or down depending upon the requirements. Furthermore, it facilitates information portability to locations where there is no internet connectivity and has support for user roles and authentication to ensure privacy of data and protect from malicious users. Moreover, the developed system allows trusted organizations – public, private, or military – to build their own networks while ensuring data consistency and avoiding repetition. The system also successfully embraces characteristics like ease of adaptability, economics in deployment, comprehensiveness and efficient information gathering.

  The data being used is recorded in a structured format by designated officials who are accountable for its accuracy. Another highly useful feature of the system is that the information remains accessible even in those areas where there is no connectivity because of distributed repositories at authorized locations. However, since data is being recorded manually before translation into an electronic format, the possible delays incurred signal towards a compromise in the timeliness of response for the system overall.

- Simulation and System Dynamic Models for Transportation of,Patients Following a Disaster
  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 at smarter planning of rescue efforts. In this regard, Ying Su et al [23] have presented a simulation model for planning the transport of patients to nearest medical centers following the occurrence of a disaster. Their major focus has been on discrete event model simulation to aid the process of planning the rescue strategies – specifically transportation of injured people for medical assistance under unpredictable circumstances. Moreover, determining the time taken by given number of ambulances to reach patients, the longest waiting time for an injured person, the number of ambulances required to transport patients, and finally standard operating procedures to be followed for transferring patients to nearby hospitals define scope of their research.

  For this purpose,a simulation model has been designed which uses Chengdu city in Sichuan province as disaster site and its data was collected from EMS (Emergency Medical Services), which includes information about number of hospitals, time required to transport a patient from different locations to hospitals, and time to load and unload patients. Based on this information, the sample results at each hour were analyzed separately and duly validated by individuals from emergency medical services and management. These domain experts were consulted to inspect whether the response of the developed system conforms to actual ambulance services. Their work is beneficial for victims of disaster as it considers a post disaster scenario, where data is 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 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 case of the absence of infrastructure in the aftermath of disasters such as earthquakes.

- Template-based Methodology for Disaster Management Information

Systems
Thus far, we have seen that timely decisions and effective plans based on accurate and comprehensive information are necessary elements to carry out rescue efforts in order to save lives and prevent economic loss. Moreover, lessons learnt from past events can significantly reduce the scale of damage sustained from natural calamities. In this regard, different methodologies have been also been designed to promote the role of efficient communication in disaster management, as it is vital for early warnings as well as developing and relaying emergency response plans. One such methodology has been proposed by Jintae Lee and Tung Bui [24] for disaster management which is derived from features such as severity of condition of people affected by a disaster. Hence, this approach stresses on gathering, filtering, organizing and processing the information in a case based manner prior to a disaster.

Moreover, in order to avoid panic in case of an unforeseen event, it is advisable to automate the planning and response mechanism as much as possible while leaving only implementation decisions to concerned authorities. The research presented therein provides methods to design templates for each activity in a possible disaster scenario to collect analyze and organize the relevant information required for automation. According to developed procedures, each activity in a given crisis should be represented by a template which is composed of different sections such as activity decomposition, resource utilization, reasons for success or failure, and generalizability. Once all events are emulated, then normative templates can be derived from descriptive ones. After validation, they would be used as actions in a checklist of components required to implement a system while associating their use with various levels of disaster.

While the overall approach is quite advantageous in terms of collecting desired information to build a comprehensive disaster management system, it still seems quite ambitious in terms of applicability to real-world situations – especially since it is a significant task in itself to define the scope of different types of relief activities under emergency circumstances.

2.4 Simulation platforms

Decision support and coordination have been recognized as elementary features for smarter disaster management systems which provide immediate response for emergency situations. In order to assess these key characteristics, several systems have embedded agent based modelling technique as it allows designing of pragmatic pre and post disaster situations. Furthermore, intelligent agents have the ability to support all necessary information phases, act in dynamic environments, make human like decisions by making use of optimized algorithms and human-computer interaction, and finally integrate collaboration mechanisms. For this purpose the following simulation tool kits were used by systems discussed in previous sections.

Systems proposed by Francis Chen et al and M. Boston et al as mentioned earlier in this chapter have used Netlogo [25][37] to model environment for computations and simulations. Netlogo embraces attributes like exporting and importing data to and from other applications, BehaviorSpace for parameter sweeping and 'agent sets' which makes it suitable for simulations. Moreover, Netlogo is written in Java which is a cross platform language and along with JVM (java virtual machines), this enhances performance. Apart from support for world topologies, language extensions and 3D graphics, the simulator lacks integration of data mining and analysis which makes it unsuitable for the systems which tend to find useful patterns of information and perform data analysis.

Alternately, the framework proposed by Kei Sato et al combines RCRS-Robocup Rescue Simulation [26][36] and USARSim [27] through different protocols to map integrated results from disaster simulation and agent actions in different cities. As RCRS is composed of simulator league, essentially its focus revolves around team coordination, planning and collaboration of robots to achieve specific goals. Being the key determinants of disaster management systems, comprehensiveness and timeliness are also inculcated in RCRS design which makes application to real and large scale catastrophic events possible, although lacking in accuracy. USARSim follows a practical approach for modelling robots, using actuators and sensors to accomplish complex tasks in unstructured and dynamic environments. This leads to an exchange of code between real and simulated robots, thus enhancing the accuracy of system. However, simultaneous exchange can result in long delays as errors might result when code is migrated between two platforms.

Another model presented by A. T. Crooks and S. Wise to demonstrate exploitation of data in system, was built in java using MASON [28][38] and its GIS extension [29]. MASON is java based, multi-agent simulation toolkit which successfully utilizes machine learning concepts for visualization of convoluted social environments, and is also beneficial for computationally challenged systems as it delineates between visualization and model of the particular system. Unfortunately, at present MASON is deficient in 3D visualizations and does not provide charting features which makes analysis of simulation results onerous.

Moreover,DEFACTO [39] has also been used by few disaster management applications , that currently emphasize on immediate disaster response. Designed system also support human-agent collaboration and subsumes properties like artificial intelligence, 3D visualizations and also enhance humanagent interaction in complex environment but proposed system only evaluates strategies involving human-agent interaction, agent and human teams.

2.5 Comparative analysis

Based on existing multi-agent architectures and disaster management systems, following characteristics have been identified that a multi-agent system should have in order to control and manage robots – especially in case of post-disaster scenarios.

2.5.1 Multi-agent system characteristics

- Fault Tolerance
  Indicates how well a system can handle failures and indicates the incorporation of mechanisms such as failure detection and recovery.

- Scalability
  Scalability is an important characteristic which ensures efficiency and productivity of system at large scale, while considering the dynamics of chaotic situation after disaster. It also refers to how well system can perform when new software or hardware modules are added to the exiting design.

- Adaptability
  Any practical system should have the capability to adapt according to the situation in order to provide better results. In the case of natural disasters, the outcomes are mostly unknown. Hence, an effective system must be able to adapt and react according to the actual situation.

- Coordination
  It refers to the type of coordination mechanism a system utilizes. It can be either centralized or distributed.

Table 2.1 summarizes the comparative analysis of existing multi-agent systems to control robots based on above mentioned characteristics.

Name of Architecture	**Coordination**	**Scalability**	**Fault Tolerance**	**Adaptability**	**Environment**	**Approach used**	**Purpose of Design**
Architecture for cloud based management of collaborative robots.	Distributed	Yes	Yes	Selective	Simulated	Reactive and Proactive	Data gathering during catastrophic events.
Designing an Agent System for Controlling a Robotic Soccer Team	Centralized	No	No	Yes	Real robots	Reactive	Controlling group of robots participating in E- league competition.
Control Architecture Mobile Robots using Multi-Agent based for Cooperative Coordination Approach	Centralized	No	Yes	Yes	Simulated	Reactive and Proactive	Coordinate movements of robots and to assist humanitarian tasks.
A Framework and Architecture for Multi-Robot Coordination	Distributed	Yes	Yes	Yes	Real robots	Reactive and Proactive	Real-time Coordination of multi-robot system.
A Multi-AgentArchitecture with Distributed Coordination for an Autonomous Robot	Distributed	No	Yes	Selective	Simulated and Real Robots	Reactive and Proactive	Robot should perform complex trajectory planning.
Coordination or multi-robot coordination and strategies f mapping	Centralized	Yes	Yes	No	Real robots	Reactive	Explore various collaboration strategies used by group of mobile robots to map uncertain/unstructured environment.
ARTIS	Centralized	Yes	Yes	-	Real robots		Transport objects from one office to another on same floor.
IDEA	Not Centralized	-	Yes	-	-	-	To control multi-robot systems.

Table 2.1: Comparative Analysis of Existing Multi-Agent Architectures

2.5.2 Disaster management system characteristics

Similarly, the common features against which the effectiveness of disaster management systems may be evaluated are listed as:

- Reliability
  Socio economic loss incurred due to natural disaster can be reduced if emergency response system is reliable. Therefore, presence of this feature holds vital importance, as it refers to how well a system is able to respond at or above its capabilities to respond to unfortunate events. According to Evaluating the Reliability of Emergency Response Systems for Large-Scale Incident Operations [30] reliability of such systems depends on several factors, in particular scale, scope and complexity of incident should be taken into account when measuring reliability of system. Similarly techniques like fault tree analysis, effects, failure mode and critical analysis can be used to scrutinize system's response reliability. Additionally they have mentioned four steps as shown in diagram below to inspect emergency response systems for large scale disasters.

- Flexibility
  Disaster management systems should be designed in a way that embraces flexibility. When it comes to responding to large scale disaster, flexibility plays an essential role to achieve resilience at various levels including individual, societal and organizational. Thus, flexibility encourages ad hoc use of the system which is often required for emergency management systems, as described in "Improving Disaster Management: The Role of IT in Mitigation, Preparedness, Response, and Recovery" [31]. Moreover, flexibility is one of the common attribute of agent based models, distinguishes them from conventional systems, as design of ABM system can easily accommodate more agents and it provides natural framework to adjust complexity of agents' w.r.t behavior, ability to learn, degree of rationality and interaction [32]. There are still many challenges identified by Gary R. Webb [33] which need to be addressed in order to incorporate flexibility in disaster response systems.

- Scalability
  Scalability is another important characteristic which ensures efficiency and productivity of disaster management system at large scale, while considering the dynamics of chaotic situation after disaster. Agent based models are tend to be more scalable then conventional systems as they have ability to handle the growing amount of work.

- Performance
  Effective and productive disaster management system requires quick and reliable decision making especially in situations of panic. Besides other characteristics, the response time of system should be high because a reliable, flexible or scalable system will be impractical if its performance is compromised. Apart from application, performance of disaster management system depends upon effective data management by making efficient use of communication channels because according to Andreas Meissner et al [34] transmission of large amount of information like convoluted maps with in short time is a complex task. Therefore overall system's capacity at local level also effects performance of the system, so it is preferred to enhance the capacity of local components in order to improve overall performance of the system [35].

- Conformance to real world scenarios
  Successful and practical disaster management system should conform to reality because natural description of the system is necessary to formulate plans in order to carry out effective decision making. Systems which tend to be more close to reality are more likely to generate accurate results. Moreover, ABM systems have potential to provide natural description of the system which conforms to the reality with greater degree.

Based on above mentioned features Table 2.2 shows comparisons of existing disaster management systems.

Name	**System Category**	**Comprehensive**	**Accuracy**	**Timeliness**	**Accessibility**
Humanitarian relief through crowdsourcing,volunteered geographical information and agent based modeling.	Agent-based	No	Partial	Yes	Yes
Dynamic adaptive disaster simulation.	Agent-based	Yes	Yes	No	Yes
SAHANA	Conventional	Yes	Yes	No	Yes
RESCUE	Agent-based	No	Yes	Yes	No
Towards Assessing,Resilience,of a Community in Seismic Events,Using Agent Based Modeling.	Agent-based	No	Partial	Yes	No
Agent Evacuation Simulation Using Hybrid and Free Space Models	Agent-based	Yes	Partial	Yes	No
Simulation and System Dynamic Models for Transportation of,Patients Following a Disaster.	Conventional	Yes	Yes	Partial	No
Proposal of Building Rescue Management System.	Agent-based	No	Yes	Yes	Yes
A Template-based Methodology for Disaster Management Information Systems	Conventional	Partial	Partial	No	No

Table 2.2: Comparative analysis of Disaster Management Systems

Chapter 3

Proposed Multi-agent Architecture

This chapter gives a comprehensive overview about proposed multi-agent system architecture for cloud framework to manage collaborative robots in physical environment. Different components of the system have been illustrated, along with detailed design of agents that make up the architecture. Furthermore, the interaction between different components of a system has also been described.

3.1 Proposed framework

The proposed multi-agent architecture deployed over a cloud framework to control multiple robots (quadcopters) in physical environment can best be described in terms of main components taken from blackboard architecture. Here, a common or shared knowledge base blackboard is updated by a diverse set of knowledge sources, where each knowledge source contributes to the solution using its partial knowledge. Simply put, blackboard model is designed to handle convoluted problems where solution is typically sum of its parts.Our proposed framework is somewhat similar to distributed multi-agent system presented by Bianca Mariela [14].

Since the proposed framework 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.Victim's information is conveyed to quadcopters through an android based disaster application which uses Bluetooth low energy beacon to communicate to another beacon mounted at the top of quadcopter as shown in Fig 3.1.Using collective sensory knowledge of different quadcopters deployed for surveying disaster affected areas; multi-agent system fabricates a mental picture of 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.

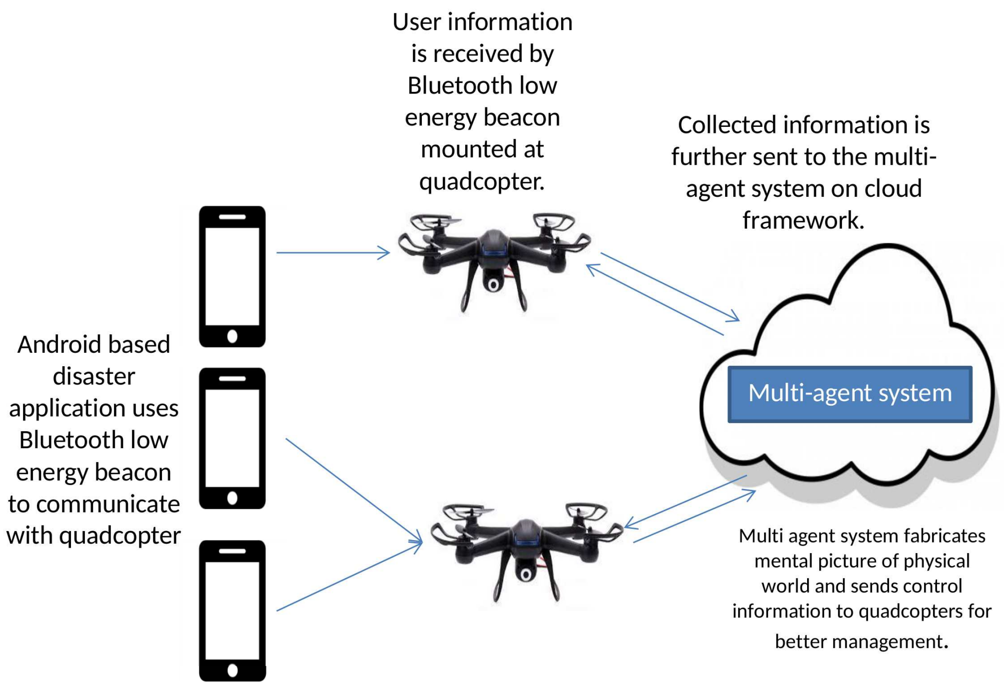

Figure 3.1: Allocation of new Quadcopter

The proposed multi-agent framework embodies following components:

1. Environment Monitoring
2. Quadcopter Management
3. Immediate Response Planning
4. Blackboard Consolidation

3.2 Environment monitoring

During catastrophic events,timely response can save many lives and minimize post disaster damage. Therefore, monitoring component of proposed system repeatedly monitors the environment for disaster situations and makes decisions accordingly. Fig 3.2 describes monitoring component in detail and outlines its behavior while observing environment.

3.3 Quadcopter management

This component makes best possible use of available resources for capturing maximum data about victims of an earthquake. It achieves this by remotely surveying an 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 assigns survey zone to its associated quadcopter and controls movement of quadcopter between adjacent plots in a specific zone.

3.3.1 Quadcopter initialization

Whenever a disaster occurs, the environment monitoring agent initiates quadcopter management agent which needs to initialize quadcopters in a given area.

The quadcopter management agent determines number of quadcopter to be sent in an affected area based on the following factors:

1. Using past data
   Following a disaster, quadcopter management agent consults earthquake history of similar magnitude and determines number of quadcopters to be sent. If past data includes multiple entries of earthquakes of same magnitude then decision about number of quadcopter will be taken using following formula:

   $$P_(qc+1) = \frac{1}{no.of enteries found}\{[P_qc1 + (r \times \frac{Q_n1}{\Delta t_1 \times P_qc1})] + [P_qc2 + (r \times \frac{Q_n2}{\Delta t_2 \times P_qc2})]...[P_qcn + (r \times \frac{Q_nn}{\Delta t_n \times P_qcn})]\}$$

   Where
   $P_(qc+1)$ = proportion of required quadcopter to be sent.

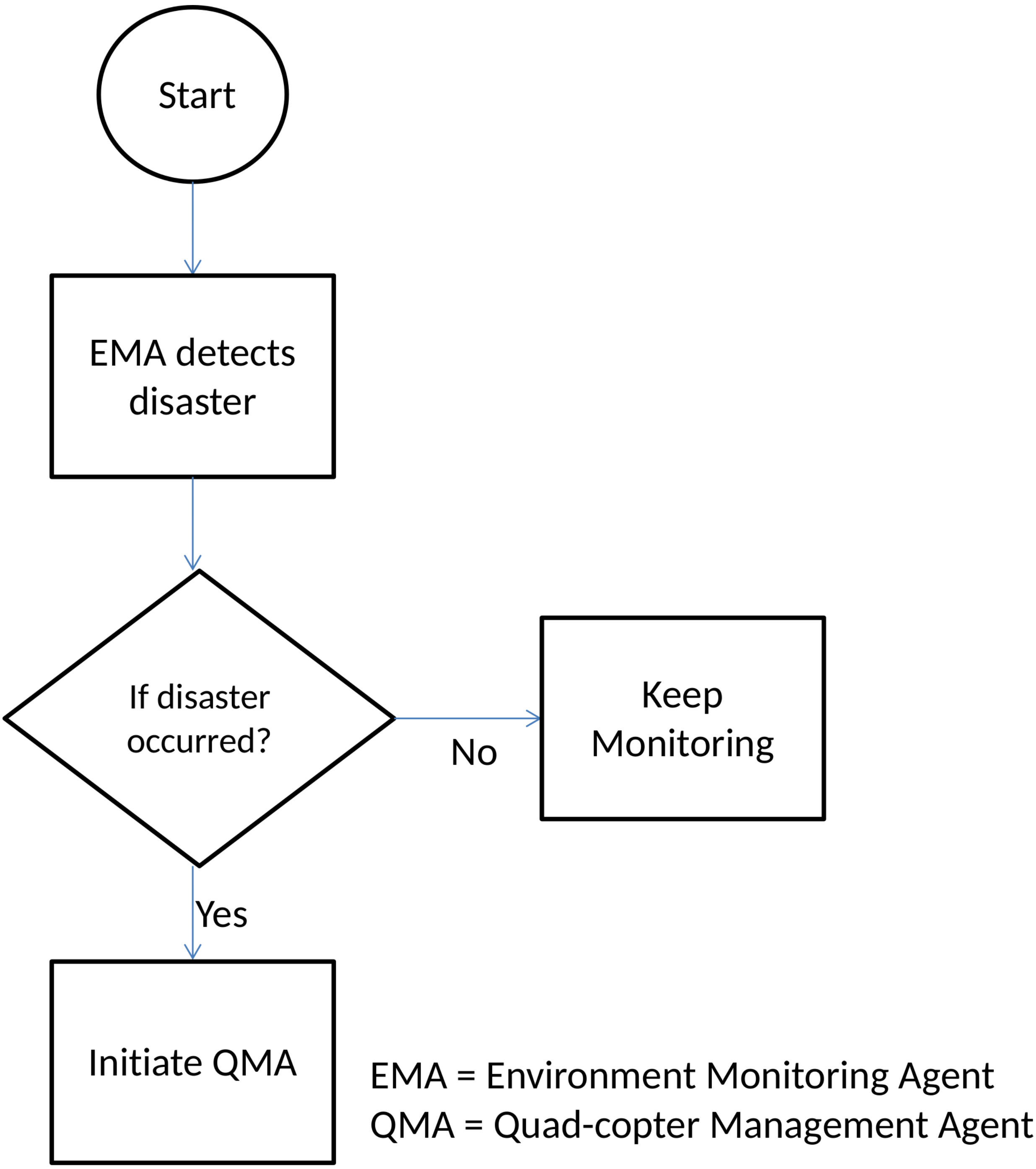

Figure 3.2: Environment Monitoring Component

P_qc = proportion of sent quadcopters.
r = rate of population growth.
Q_n = no. of Quadcopters sent.
Δt = difference between the previous and current years.
Alternately, in case of single entry in past data, number of quadcopters to be sent will be decided by following formula:

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

Above mentioned formulas are based on logistic model [48]

2. Using assumptions of Earthquake Magnitude Given the uncertainty regarding the magnitude of earthquakes, there is often very little information about past earthquakes occurred in certain area, so in that case our system will utilize 50% of available resources(quadcopter).

Once a decision about quadcopter initialization has been made, quadcopter management agent will create and add associated quadcopter agents in multi-agent architecture to represent and control individual quadcopters.

3.3.2 Quadcopter survey

After creation of associated quadcopter agents, individual quadcopters are assigned to their respective survey zone which consists of multiple adjacent plots. During survey, local agent of quadcopter periodically updates its associated agent with current location as described by Fig 3.3. Moreover, it checks for messages and also sends acknowledgement to the sender.

Based on location updates from local agent of particular quadcopter, associated quadcopter also manages the movements between different plots in assigned zone and marks particular zone as ‘OK’ if all plots have been surveyed. Fig 3.4 below further explains the quadcopter movement scenario.

3.3.3 Allocating more quadcopters in area with greater extent of damage

Apart from quadcopters assignment and management of quadcopter’s movement, this component also allocates more robots to area with greater extent of damage based on the count of distress beacons broadcasted by the victims from a particular zone.

The maximum number of message violations in a particular zone has been set to 3; and as soon as the threshold is crossed the quadcopter management

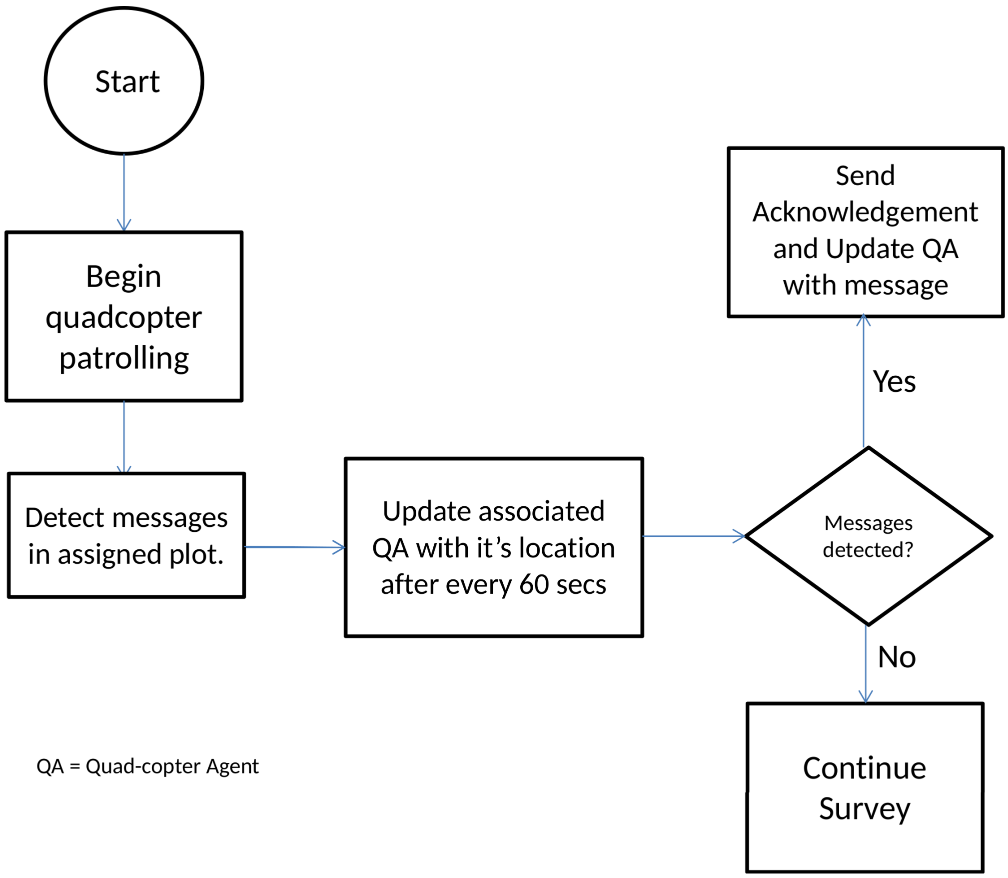

Figure 3.3: Quadcopter Survey

agent will allocate new quadcopter in that particular zone and assign a new set of plots to it which have not been surveyed yet.

3.4 Immediate response planning

Another important aspect of this system is pertaining to the support it offers in the planning of immediate response efforts in case of disaster. This component constructs clusters based on sensory information received from victims' using Euclidean distance and hierarchical clustering technique. After clustering the data, this component assigns priority value 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). Therefore, clusters with higher priority more likely require immediate help. Furthermore, this component uses Dijkstra algorithm to calculate shortest distances to the centroid of high priority clusters, which helps rescue teams to plan their movements accordingly.

3.5 Blackboard consolidation

This component plays significant role in this system whereby a central database is continually updated by a group of quadcopters based on the respective data received by them. This information is accessible to all other agents in 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 a disaster scenario.

3.6 System architecture

System architecture is composed of individual interrelated components that make up whole system architecture. Block diagram in Fig 3.5 explains how different components interact together to provide whole system functionality.

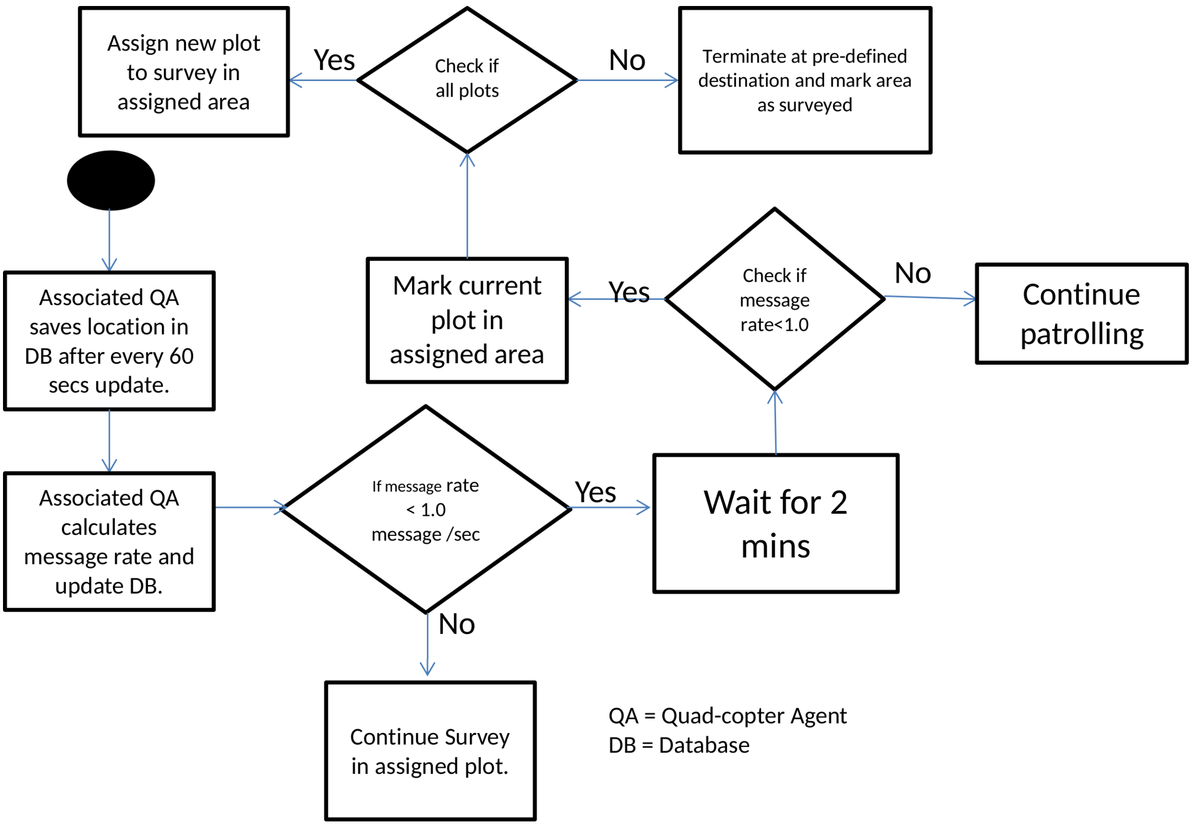

Figure 3.4: Quadcopter Movement

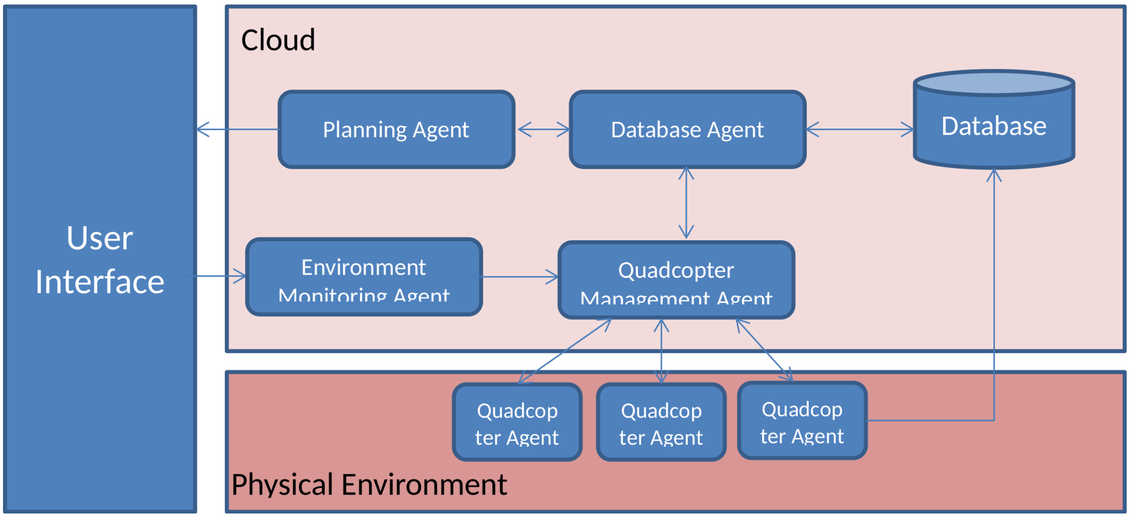

Figure 3.5: System architecture

3.7 Summary

Overall designed system will process information receive by quadcopters at particular area followed by disaster specifically earthquake. For data collection purpose multi-agent system must detect disaster and for this purpose environment monitoring module is included in the system. Once, disaster at particular area has been detected quadcopter management module will take the responsibility of quadcopter initiation using past data (if available) or assumptions, creates quadcopter agents which are associated with particular quadcopter and sends control information to them, manages quadcopters during survey and allocates more quadcopters in an area with greater extent of damage. Next, based on data received by quadcopters immediate response planning is accomplished with help of hierarchical clustering. Also, shortest paths to the centroid of high priority cluster are planned using Dijkstra algorithm. Finally, interaction between above mentioned modules to achieve overall system functionality has been explained.

Chapter 4

Experimental Setup

This chapter discusses implementation details of multi-agent framework proposed in previous chapter. It provides a brief overview of multi-agent development methodology and platform used for experimental setup. Next, simulation system that has been developed for data collection is presented. At the end, description of cloud framework where multi-agent system resides is outlined.

4.1 Introduction

Implementation and testing of robotic frameworks that require conformance to the real world scenarios is a convoluted task; as it involves extensive series of steps from real time planning to movement of robots in physical world. In our case, diverse technologies such as robotics, multi-agents and cloud computing are used to achieve a particular goal. Therefore, development process needs vigilant groundwork to assure usefulness of developed framework and repeatability of results.

Development of agents demands suitable programming language, which is associated with intended application of multi-agent framework. Since our framework is intended to manage quadcopters, which are used for data collection at disaster site following an earthquake. Therefore, programming language should have support for real time coding and is platform independent, so Java programming language was selected for implementation purpose.

Finally, in order to test adaptability of proposed multi-agent framework for cloud, it is useful to have simulation software to simulate quadcopters at disaster site following earthquakes with varying magnitude, which further communicates with multi-agent system over a cloud framework for planning purposes.

4.2 Proposed framework methodology

Initially several multi-agent architectures and development methodologies were surveyed and analyzed and the methodology we followed includes different phases presented in figure 4.1 below

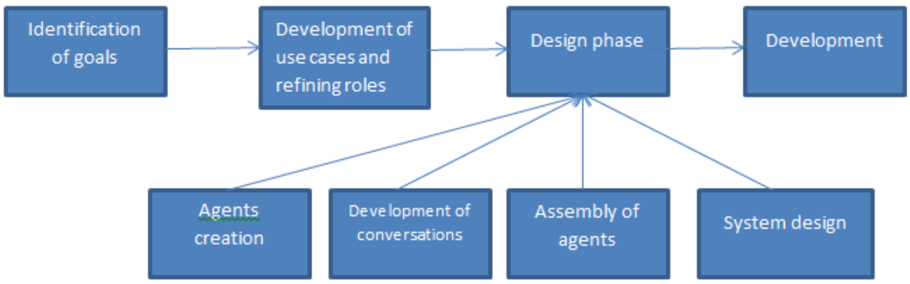

Figure 4.1: Framework development methodology

First, goals of proposed framework were identified and according to them various use cases were developed which were translated to different roles that needs to be performed in order to achieve set of goals identified. Second phase mainly includes two sub phases: i) construction of use cases to describe system behavior. ii) second one is sequence diagram which shows sequence of events to achieve desired system behavior.Next, design phase includes creation of agents that are capable of performing roles identified in second phase, development of conversations for different agents that needs to communicate with one another succeeded by steps of assembling agents and construction of system design. Finally, development of system defines the implementation configuration of a current system.

4.3 Multi-agent platform

Originally, several existing platforms for development of multi-agent frameworks were scrutinized and JADE [40][44] was selected among them as it facilitates the implementation of multi-agent applications in acquiescence with FIPA specification.

4.3.1 JADE

JADE platform keeps the high performance of developed agent system with Java – programming language selected for this thesis. In addition to this JADE offer flexible and efficient communication channel among agents and encourages peer to peer communication, this makes it even more suitable for rapid decision making in case of disaster. Moreover, it incorporates features like interoperability, uniformity and portability, easy to use and pay as you go philosophy.

Apart from above mentioned features, JADE's modular architecture enhance its adaptability to resource constrained environments and from functional perspective it provides the basic services for distributed multi-agent applications by allowing each agent to discover other agents dynamically[41]. Whereas from application perspective agents are identified with help of unique name or id and provides a group of services which can also be modified later.

Following features of this platform made it suitable for our work:

- It is able to execute long and intricate task which are triggered at any time.
- It can handle applications which integrate several interactions between internal components.
- Moreover, those applications that require appearance and disappearance of components in a dynamic way or finding each other at runtime; can be effectively executed using JADE platform.

4.4 Simulation system

Another major part of this thesis is simulation of disaster site and movement of quadcopters within area. For this purpose various multi-agent simulation platforms were surveyed and GAMA [42][46] was chosen among them. As it is capable of using geographical information system (GIS) for environments of agents, handling large number of heterogeneous agents, providing platform for automated and controlled experiments and also allow designing of a model in such a way that allows non computer experts to interact with them.

In GAMA everything from roads to buildings is an agent and these agents can interact with each other to provide system functionality[47]. Major objective of our work is to collect data of victims following an earthquake disaster in such a way that captures maximum number of victims and GAMA is an appropriate platform to achieve this objective. Therefore, disaster site shown in fig 4.2 has been modeled and following key assumptions were made for purpose of simulation.

- Total area simulated city = $240000m^2$.

- No. of quadcopters required for total survey = 4.
- Total flight time of quadcopter = 20 mins.
- Speed of quadcopter = 10 m/s
- In 20 mins total distance travelled by a single quadcopter = 12000 m.
- Scanning width 5m
- Total area surveyed by quadcopter = $12000x5 = 60000m^2$.
- In 1 min battery life of quadcopter is decreased by 5%.

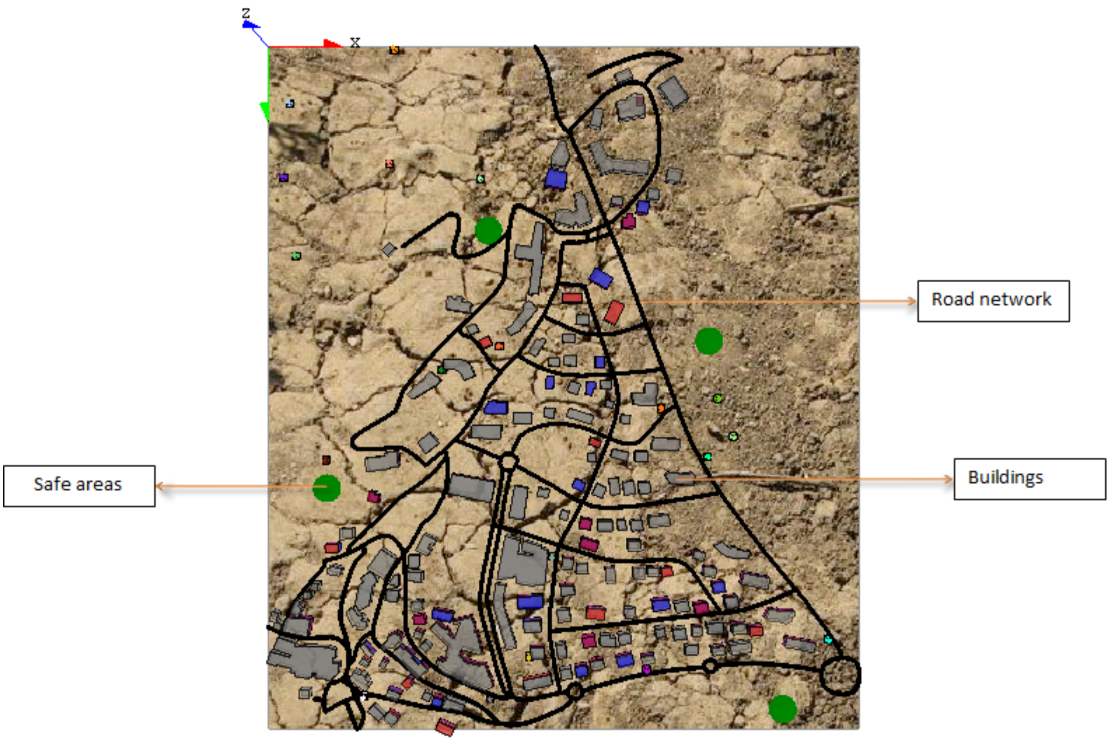

Figure 4.2: Disaster site modelled in GAMA

Along with this simulation also shows degree of damage at different earthquake magnitudes which is presented in Fig 4.3.

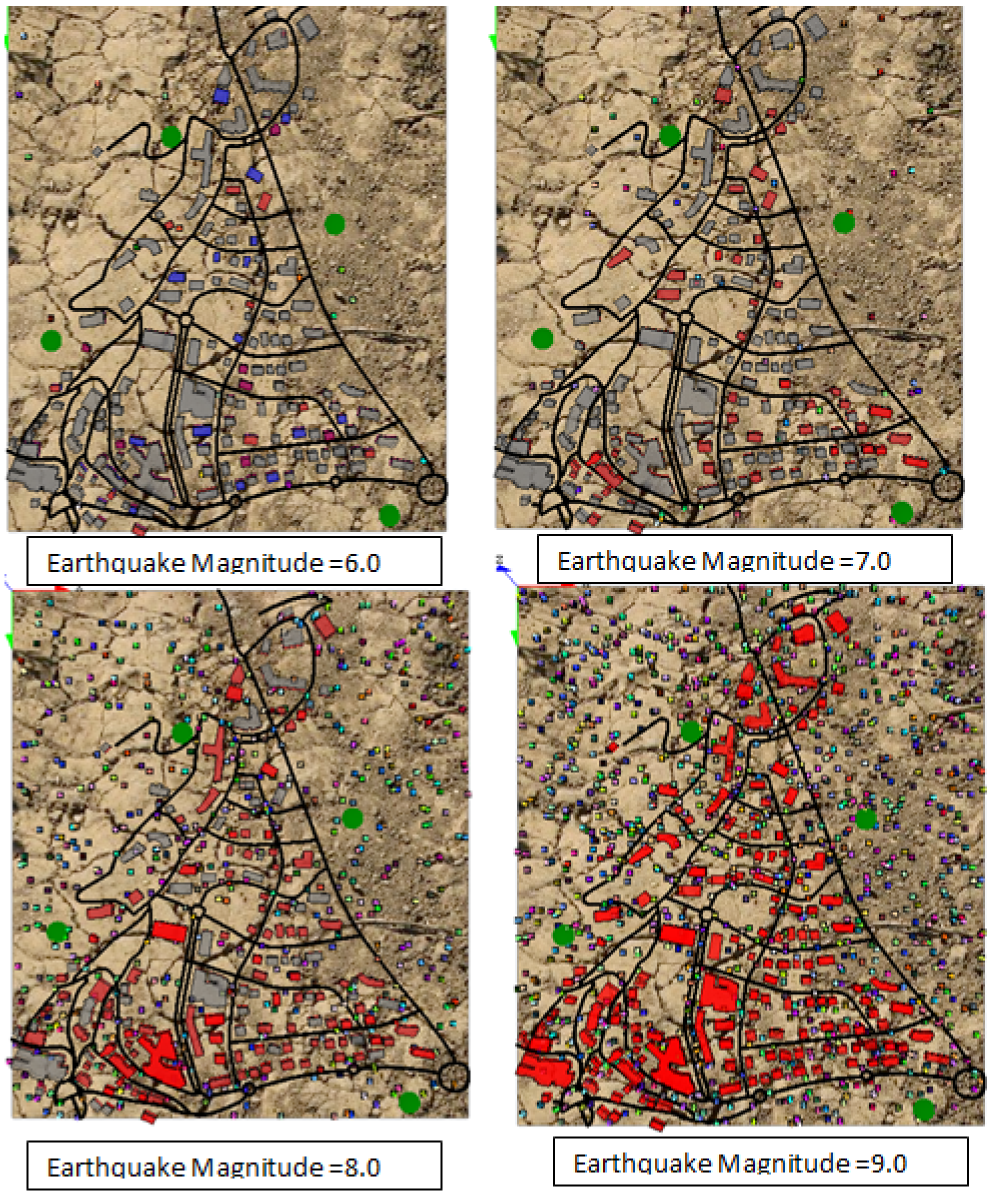

Figure 4.3: Degree of damage at different earthquake magnitudes

Furthermore when earthquake hits a particular site agents divide area into equal number of zones as shown in Fig 4.4 which composed of equal number of plots, allocates quadcopters in different zones for data collection and control movement of those quadcopters within plots. Basic idea for

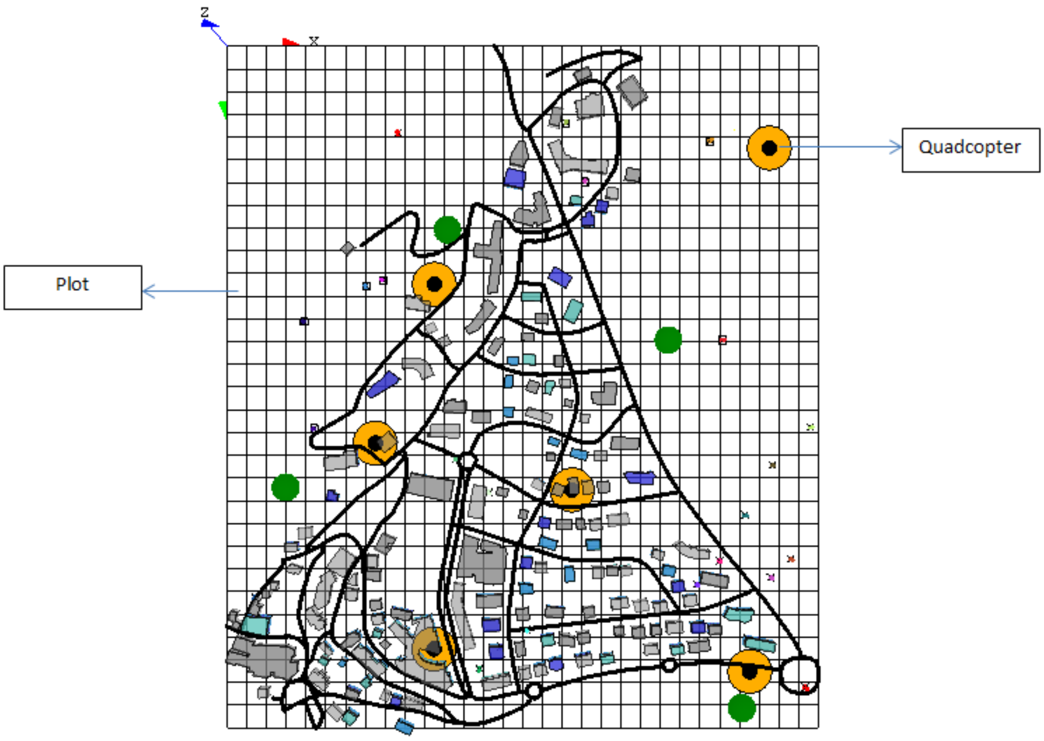

Figure 4.4: Quadcopters in different zones of disaster site

data collection is that quadcopters receive data -using Bluetooth low energy beacons placed on them-sent by victims using disaster app Fig 4.5 installed on their android phones and this application also uses Bluetooth low energy beacons for communication purposes.Data received by quadcopters include information about location of victim (longitude,latitude, altitude), status of a particular victim, date and time and magnitude of earthquake. During survey, quadcopters are managed by multi-agent system and data collected by them is finally used by planning agent to produce and prioritize clusters Fig 4.6 in order to identify people who require immediate help.

Identification of victims with immediate help can improve rescue efforts, planning and reduces the damage done by catastrophic events particularly earthquake, as timely response to those in need will increase chances of survival.

Figure 4.5: Disaster application to help users

4.5 Cloud framework

Cloud services have been utilized to enhance the battery life of quadcopter during survey, as their limited processing power restrict the installation of multi-agent system on top of primary flight control mechanism as this would require considerably proficient processors which makes it an unsuitable choice to consider because this will not only trim the battery life of drone but also effect the overall flight time. Therefore, cloud services have been used to bear intensive computing workloads [45] because designed multi-agent system will be deployed in cloud environment to effectively manage the information sent by drone.

For this purpose amazon EC2 cloud services [42] were selected because it is simple to use, provides complete control of computing resources, allows scaling of capacity up and down as computing requirements change and it is economical because it only charge for the computing capacity which is actually utilized.

4.6 Summary

In this chapter all major components of multi-agent system and simulation has been discussed. First multi-agent framework's development methodology and its phases like identification of goals, development of use cases, design and development of multi-agent architecture have been described. Next multi-agent platform i.e. JADE's features and its architecture has

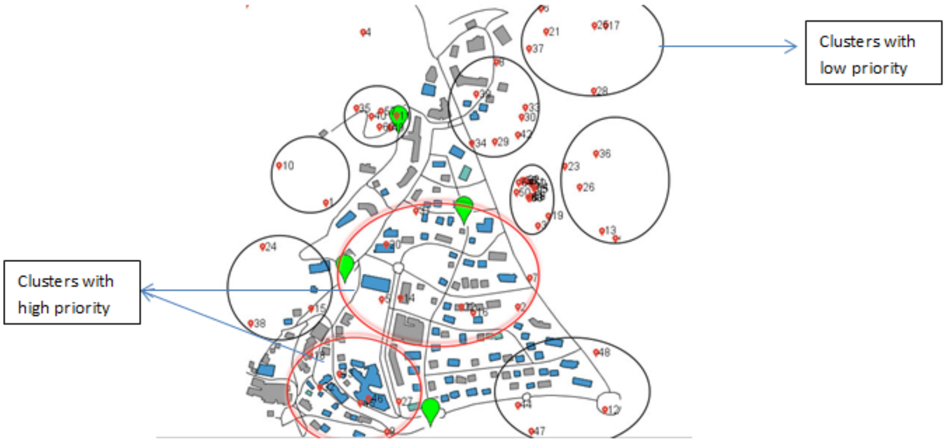

Figure 4.6: Clusters produced by planning agent

been discussed briefly. Moreover, details of simulation system which includes disaster site, effect of varying earthquake magnitudes on simulation, quadcopters movement at disaster site, user application and clustering of data. Finally cloud framework describes features of selected cloud computing service provider. In next chapter proposed framework has been tested in different scenarios and results have been provided.

Chapter 5

Experimental Results

This chapter explains results obtained with proposed multi-agent architecture using simulation and provides a brief description of data set and environment used for testing purposes. Next, evaluation basis for formulation of results have been explained. At the end, the results obtained are presented.

5.1 Dataset and test Environment

For the purpose of data collection a city affected by a disaster was simulated as referred to in chapter 4. Next, the simulated area was divided into equal number of zones to get information about population following a disaster. Once disaster site was divided into equal number of zones, quadcopters were sent to each zone to gather the information of a particular zone.

Based on collected information, the rate of distress messages sent by victims of a disaster was determined to manage quadcopters in such a way as to collect maximum possible victims' data. Since different zones have different distress message rates owing to varying proportions of disaster affected people; each zone was categorized as high, low or medium as explained further in table 5.1. Therefore, a total of three major test cases were designed and management of quad-copters by a multi-agent system was observed in all three of them. In the first case, all zones at disaster site were kept "High" priority which means that message rate of each zone was kept between 2500 and 6000. In second case, all zones were marked as "medium" priority because message rate of all zones was kept between 0 and 2500. Similarly, the third case involves all zones with "low" priority which signifies that rate of all zones observe message rate between 0 and 1000.During the testing exercise, the message rate for each zone was recorded for period of twenty minutes, and management of quadcopters was done accordingly.

Moreover, once victims' data has been collected, trends at different earthquake magnitudes were observed. In order to aid timely rescue operations, clusters were formed within the affected population and priority of each cluster was calculated based on severity of devastation. Essentially, clusters with high priority include people who require immediate help as compared to those with low priority value.

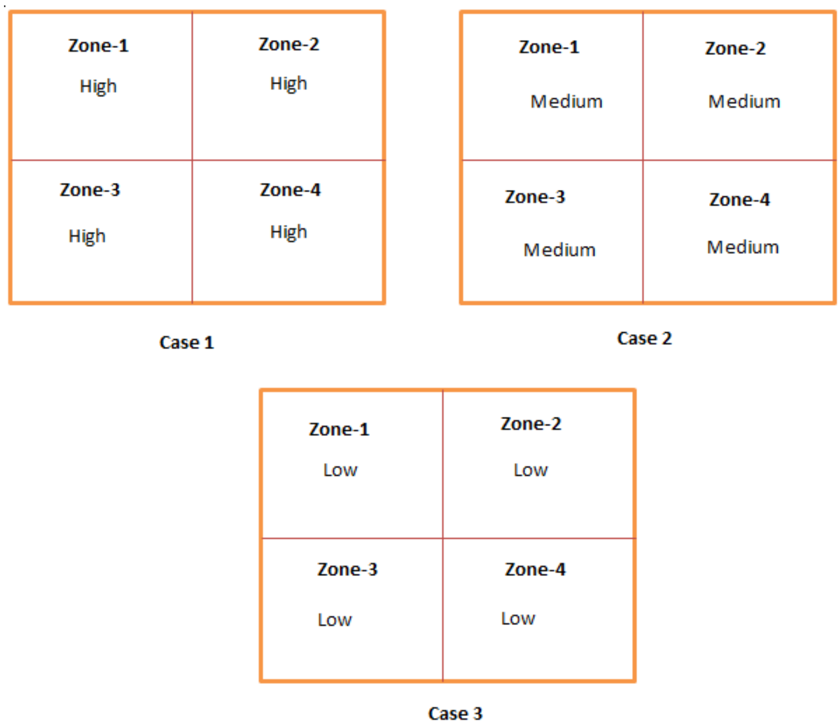

Figure 5.1: Cases used for testing and validation

5.2 Evaluation criteria

From the evaluation perspective, the following two approaches were adopted to test management of collaborative robots. Since, our basic goal is to gather maximum possible information about the affectees of an earthquake; we have compared the results using fixed assignment of quadcopters and reactive or intelligent assignment of quadcopters in all three cases mentioned in section 5.1. Furthermore, the performance of a designed multi-agent system was evaluated using local machine and cloud servers.

1. Fixed Assignment
   In this approach, the number of quadcopters assigned to each zone was kept constant to collect victim's data following a disaster (earthquake). Therefore, assuming a maximum of 8 quad-copters, 2 were assigned to each of 4 zones.

2. Reactive assignment
   Reactive assignment or intelligent assignment of quad-copters is based on an initial decision about assigning quad-copters to a particular zone using past data. Next, the number of quad-copters required for a disaster survey in a particular zone shall be increased if incoming distress message rate is greater than message handling capacity of the aerial drone.

5.3 Results

Following graph in fig 5.2 shows message rates of a particular zone over a twenty minute time period Based on message rate quadcopters were sent

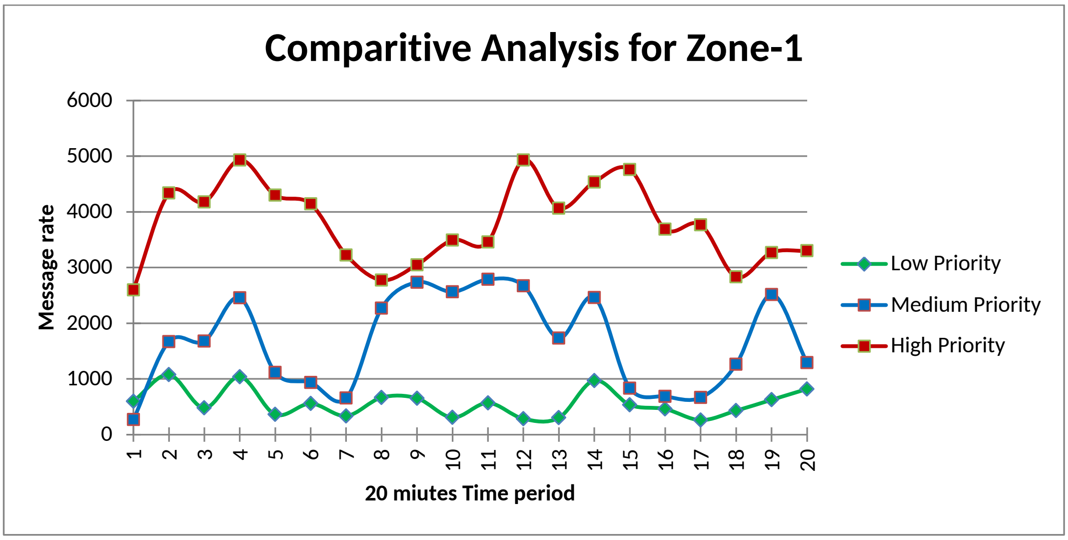

Figure 5.2: Message rates in one of the zones

using two approaches fixed assignment and reactive assignment as shown by graphs in fig 5.3, 5.4 and 5.5. According to the assumptions made earlier, we know that one quadcopter has capacity to handle 1200 messages per second. So, it can be seen in graphs that in fixed assignment-red line- we have assigned two quadcopters in each zone which denotes that 2400 messages per second are being handled throughout twenty minute time frame at particular zone. Alternatively, reactive assignment as shown on graph–green line- enhances the capacity of system to handle more messages by increasing number of quadcopters when message rate increases. From the graphs we can

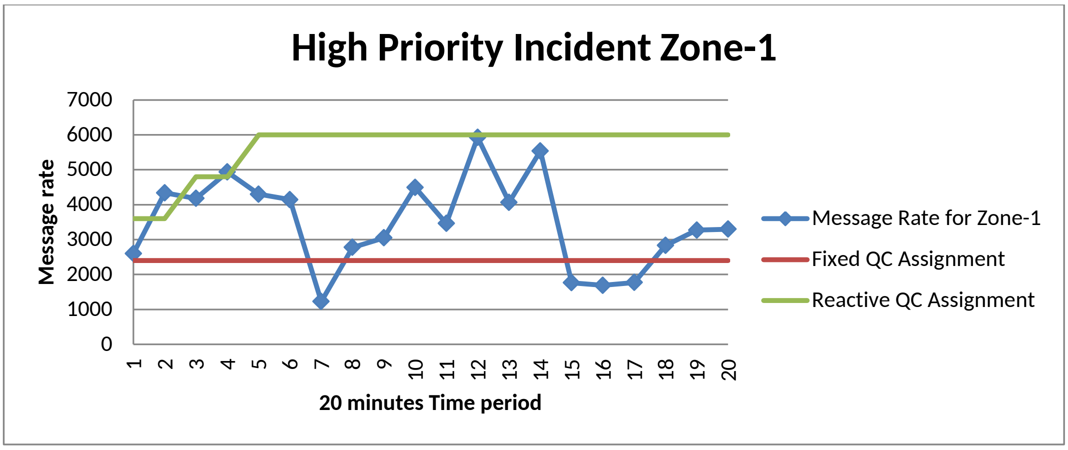

Figure 5.3: Quadcopter assignment in a zone with high message rate

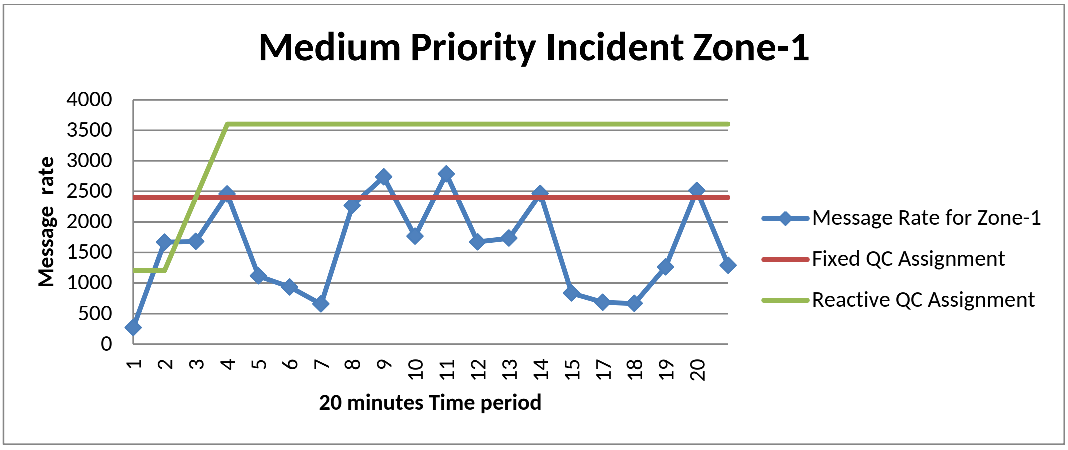

Figure 5.4: Quadcopter assignment in a zone with medium message rate

deduce that there is minimal message loss when using reactive or intelligent assignment of quadcopters as compare to the fixed assignment. Based on messages received we have also observed trends in victims at different magnitudes as shown in fig 5.6. Moreover, highest priority of cluster reached at different magnitudes was also observed as shown fig 5.7.

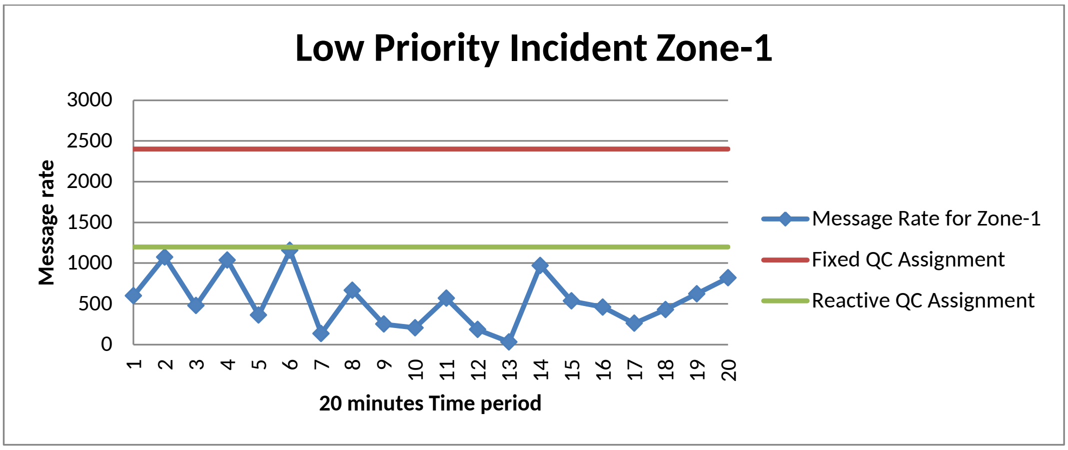

Figure 5.5: Quadcopter assignment in a zone with low message rate

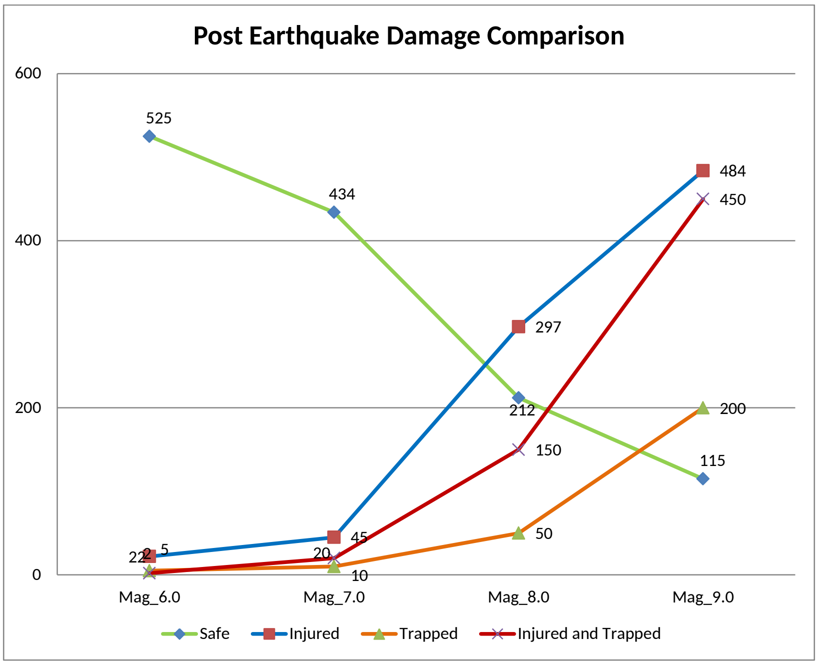

Figure 5.6: Trend in victim's data at different earthquake magnitude

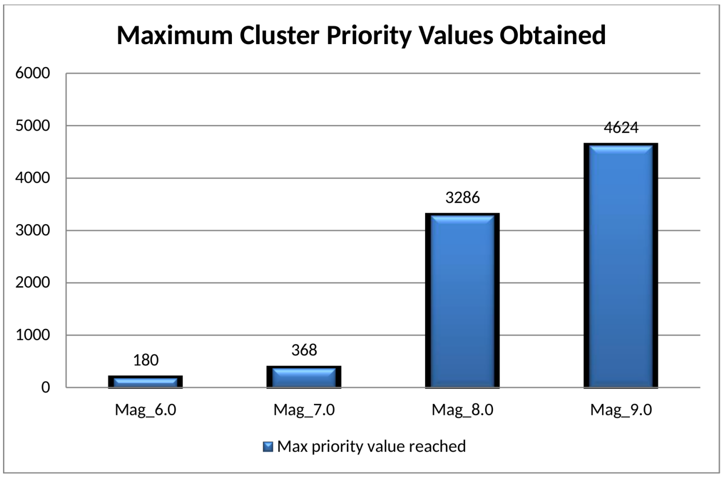

Figure 5.7: Maximum priority value of a cluster reached at different earthquake magnitudes

5.4 Summary

This chapter presents results obtained with proposed multi-agent architecture and highlights advantages and limitations of system. First zones were set to high, medium and low message rates to check how much better a system performs with different message rates. Second, two different assignment approaches-fixed and reactive assignment- were designed. These approaches further shows better results with minimal message loss when quadcopters were managed and assigned using multi-agent system rather than fixed assignment. Next, collected data enables us to observe victim's trend and maximum priority value of a cluster reached at different earthquake magnitudes. Finally, performance results of system using local machine and cloud server were presented, which shows system when placed on cloud server gives better performance.

Chapter 6

Conclusion and Future Work

This chapter summarizes the work presented in this thesis. Content presented in different chapters is reviewed and future work is discussed in detail.

6.1 Conclusion

This dissertation deals with multi-agent framework to control robots in physical environment, in short it bridges cyber world with physical world. Basically, multi-agent system for cloud is designed to control multiple robots in the form of collaborative flight drones; which gather information from their surroundings through several sensors following a disaster specifically earthquake.

Chapter 2 presents different types of existing multi-agent architectures like reactive, deliberative, hierarchical, non-hierarchical, hybrid, centralized and distributed. Moreover, it also discusses current multi-agent systems to control robots in physical environment and existing disaster management systems with multi-agent technology and without multi-agent technology. Apart from this multi-agent system characteristics and disaster management characteristics were identified for comparative analysis.

Chapter 3 outlines details of proposed multi-agent architecture: multi-agent architecture for cloud based management of collaborative robots. It provides complete picture of proposed framework. For this purpose different major modules of system were identified and explained. Moreover it also explains complete system architecture to show interaction among different modules.

Chapter 4 describes methodology used for development of proposed framework. It also presents the multi-agent platform i.e. JADE used for implementation of multi-agent architecture. Moreover, platform i.e. GAMA used for disaster –earthquake- simulation is also discussed along with assumptions made for simulation purposes. It also discusses cloud framework used by multi-agent system to control collaborative flight drones.

Chapter 5 overviews results extracted from series of experiments carried out on setup described in chapter 4. We have compared the results gained using fixed assignment of quadcopters in effected area and intelligent assignment using our proposed framework. Finally, performance of proposed system has been analyzed using simple notebook and a cloud server.

6.2 Future work

As we know, integrating three different technologies i.e. robotics, agent technology and cloud computing to work for one single system is an intricate task but we have successfully integrated these technologies for disaster management system. Based on the conclusions drawn from this thesis, agent based systems for cloud to control robotics in physical world will inevitably make way for smarter, more flexible and highly adaptable systems. So far, for validation purposes, a series of detailed simulations has been carried out subsequently. Therefore, this research exercise will be further applied on real set of quadcopters and can be extended to various applications other than disaster management system, such as structure inspection for dilapidated homes, remote monitoring of farms to reduce crop damage and locating dengue habitats. Furthermore, proposed architecture can be improved by adding more useful agents like behavioral and deliberative, in order to meet the demands of different applications with growing complexity.

References

[1] Serugendo, Giovanna Di Marzo, Marie-Pierre Gleizes, and Anthony Karageorgos. "Self-organization in multi-agent systems." The Knowledge Engineering Review 20.02 (2005): 165-189.

[2] Schumacher, Michael. "Multi-agent systems." Objective Coordination in Multi-Agent System Engineering: Design and Implementation (2001): 9-32.

[3] Talia, Domenico. "Clouds meet agents: Toward intelligent cloud services." IEEE Internet Computing 2 (2012): 78-81.

[4] Mouad, Mehdi, et al. "Control architecture for cooperative mobile robots using multi-agent based coordination approach." 6th National Conference on Control Architectures of Robots. 2011.

[5] Jose, Soler, et al. "Applying the artis agent architecture to mobile robot control." Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2000. 359-368.

[6] Hayes-Roth, Barbara. "A blackboard architecture for control." Artificial intelligence 26.3 (1985): 251-321.

[7] Muscettola, Nicola, et al. "Idea: Planning at the core of autonomous reactive agents." (2002).

[8] Xing, Wu, et al. "Agent-oriented embedded control system design and development of a vision-based automated guided vehicle." International Journal of Advanced Robotic Systems 9 (2012).

[9] Galindo, C., et al. "A multi-agent control architecture for a robotic wheelchair." Applied Bionics and Biomechanics 3.3 (2006): 179-189.

[10] Galindo, C., J. Gonzalez, and J. A. Fernández-Madrigal. "An architecture for cognitive human-robot integration. Application to rehabilitation

robotics." Mechatronics and Automation, 2005 IEEE International Conference. Vol. 1. IEEE, 2005.

[11] García, Alejandro Javier, Gerardo I. Simari, and Telma Delladio. "Designing an agent system for controlling a robotic soccer team." X Congreso Argentino de Ciencias de la Computación. 2004.

[12] Fierro, Rafael, et al. "A framework and architecture for multi-robot coordination." The International Journal of Robotics Research 21.10-11 (2002): 977-995.

[13] Nieto-Granda, Carlos, John G. Rogers, and Henrik I. Christensen. "Coordination strategies for multi-robot exploration and mapping." The International Journal of Robotics Research (2014): 0278364913515309.

[14] Badano, BM Innocenti. A multi-agent architecture with distributed coordination for an autonomous robot. Diss. Ph. D. thesis, Universitat de Girona, 2008.

[15] Crooks, A. T., and S. Wise. "Modelling the Humanitarian Relief through Crowdsourcing, Volunteered Geographical Information and Agent-based modelling: A test Case-Haiti." Proceedings of the 11th International Conference on GeoComputation, University College London, London, UK. 2011.

[16] Chen, Francis, Zhi Zhai, and Greg Madey. "Dynamic adaptive disaster simulation: developing a predictive model of emergency behavior using cell phone and GIS data." Proceedings of the 2011 Workshop on Agent-Directed Simulation. Society for Computer Simulation International, 2011.

[17] Boston, M., et al. "TOWARDS ASSESSING THE RESILIENCE OF A COMMUNITY IN SEISMIC EVENTS USING AGENT BASED MODELING." (2014).

[18] Fiedrich, Frank, and Paul Burghardt. "Agent-based systems for disaster management." Communications of the ACM 50.3 (2007): 41-42.

[19] Okaya, Masaru, et al. "Agent evacuation simulation using a hybrid network and free space models." Principles of Practice in Multi-Agent Systems. Springer Berlin Heidelberg, 2009. 563-570.

[20] Sato, Kei, Shigeru Yotsukura, and Tomoichi Takahashi. "Proposal of Building Rescue Management System using Grass Roots Software and

Public Data."Fourth International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED 2009). 2009.

[21] Coates, Graham, et al. "Adaptive co-ordinated emergency response to rapidly evolving large-scale unprecedented events (REScUE)." 8th international ISCRAM conference. 2011.

[22] Currion, Paul, Chamindra de Silva, and Bartel Van de Walle. "Open source software for disaster management." Communications of the ACM 50.3 (2007): 61-65

[23] Su, Ying, Lei Yang, and Zhanming Jin. "Simulation and System Dynamics Models for Transportation of Patients Following a Disaster." Modelling, Simulation and Optimization, 2008. WMSO'08. International Workshop on. IEEE, 2008.

[24] Lee, Jintae, and Tung Bui. "A template-based methodology for disaster management information systems." System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000.

[25] Wilensky, U. (1999). Netlogo. http://ccl.northwestern.edu/netlogo/.Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

[26] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in largescale disasters as a domain for autonomous agents research. In IEEE International Conference on System,Man, and Cybernetics,1999.

[27] J. Wang S. Balakirsky S. Carpin, M. Lewis and C. Scrapper. Bridging the gap between simulation and reality in urban search and rescue. Robocup 2006: Robot Soccer World Cup X, LNAI Vol. 4434 Springer, page 112, 2007.

[28] Luke S, Cioffi-Revilla C, Panait L, Sullivan K and Balan G, 2005, MASON: A multi-agent simulation environment. Simulation, 81(7): 517-527.

[29] Sullivan K, Coletti M and Luke S, 2010, GeoMason: GeoSpatial support for MASON. Department of Computer Science, George Mason University, Technical Report Series, Fairfax, VA.

[30] Jackson, Brian A., Kay S. Faith, and Henry H. Willis. Evaluating the reliability of emergency response systems for large-scale incident operations. RAND CORP ARLINGTON VA HOMELAND SECURITY AND DEFENSE CENTER, 2010

[31] Schmitt, Ted, Jon Eisenberg, and Ramesh R. Rao, eds. Improving Disaster Management:: The Role of IT in Mitigation, Preparedness, Response, and Recovery. National Academies Press, 2007.

[32] Bonabeau, Eric. ”Agent-based modeling: Methods and techniques for simulating human systems.” Proceedings of the National Academy of Sciences99.suppl 3 (2002): 7280-7287.

[33] Webb, Gary R., and Francois-Regis Chevreau. ”Planning to improvise: the importance of creativity and flexibility in crisis response.” International Journal of Emergency Management 3.1 (2006): 66-72.

[34] Meissner, Andreas, et al. ”Design challenges for an integrated disaster management communication and information system.” The First IEEE Workshop on Disaster Recovery Networks (DIREN 2002). Vol. 24. 2002.

[35] Kapucu, Naim, Christopher V. Hawkins, and Fernando I. Rivera, eds. Disaster resiliency: Interdisciplinary perspectives. Routledge, 2013.

[36] RoboCup Rescue. Web. 05 Sept. 2015. ¡www.robocuprescue.org¿.

[37] ”NetLogo Home Page.” NetLogo Home Page. Web. 03 Sept. 2015. ¡http://ccl.northwestern.edu/netlogo/¿.

[38] ”MASON Multiagent Simulation Toolkit.” MASON Multiagent Simulation Toolkit. Web. 15 Sept. 2015. ¡http://cs.gmu.edu/ eclab/projects/mason/¿.

[39] Marecki, J., Schurr, N. and Tambe, M. (2005) Agent-Based Simulations for Disaster Rescue Using the DEFACTO Coordination System, In Emergent Information Technologies and Enabling Policies for Counter-Terrorism (Eds. Popp, R.L. and Yen, J.), Wiley-IEEE Press, pp. 281-297.

[40] Bellifemine, Fabio, Agostino Poggi, and Giovanni Rimassa. ”Developing multi-agent systems with JADE.” Intelligent Agents VII Agent Theories Architectures and Languages. Springer Berlin Heidelberg, 2000. 89-103.

[41] Bellifemine, Fabio, et al. ”JADE: A software framework for developing multi-agent applications. Lessons learned.” Information and Software Technology 50.1 (2008): 10-21.

[42] Amouroux, Edouard, et al. ”GAMA: an environment for implementing and running spatially explicit multi-agent simulations.” Agent computing and multi-agent systems. Springer Berlin Heidelberg, 2007. 359-371

[43] Amazon EC2 - Virtual Server Hosting [Online; accessed January , 12, 2016.].

[44] "JAVA Agent DEvelopment Framework." Jade Site. Web. 20 Nov. 2015. ¡http://jade.tilab.com/¿. .

[45] Bateman, Alex, and Matt Wood. "Cloud computing." Bioinformatics 25.12 (2009): 1475-1475.

[46] Taillandier, Patrick, et al. "GAMA: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control." Principles and Practice of Multi-Agent Systems. Springer Berlin Heidelberg, 2010. 242-258.

[47] "Gama-platform/gama." GitHub. Web. 03 Nov. 2015. ¡https://github.com/gama-platform/gama/wiki/Pedagogical-materials¿.

[48] Tsoularis, A., and J. Wallace. "Analysis of logistic growth models." Mathematical biosciences 179.1 (2002): 21-55.