A DECISION MODEL FOR EMERGENCY RESPONSE

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By
Ashok Kumar Vellaswamy Chelaiah Rothimaswa

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2012

Fargo, North Dakota
North Dakota State University
Graduate School

Title

A Decision Model for Emergency Response

By

Ashok Kumar Vellaswamy Chelaiah Rothimasw

The Supervisory Committee certifies that this disquisition complies with North Dakota State University’s regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kendall Nygard
Chair

Dr. Changhui Yan

Dr. Chao You

Approved:

03/20/2013

Date

Dr. Brian Slator

Department Chair
ABSTRACT

In the real world, emergencies occur at different places. The emergencies fall into three categories: a fire emergency, a medical emergency, and a police emergency. Different emergencies occur at different places at different times. An important problem in the real world is how emergency vehicles are efficiently dispatched.

This paper concerns the design and development of a software tool for decision making in an emergency response. With this Emergency Response System (ERS) tool, we will treat all emergency vehicles as agents. For a fire emergency, we have a fire truck (one agent), for a medical emergency, we have an ambulance agent, and for a police emergency, we have a police-car agent. The emergencies are created at random locations. The ERS tool uses Fuzzy logic [1] to dispatch the vehicles efficiently. We assign the weights dynamically to all the response units. Based on those weight values, the tool assigns and dispatches the vehicles.
ACKNOWLEDGMENTS

I would like to express my thanks to my adviser, Dr. Kendall Nygard. His research topic, advice, guidance, and support greatly helped in making this work possible. Also, I would like to sincerely thank Dr. Kendall Nygard for his suggestions and improvements which helped to improve the paper’s quality.

During my studies and during my paper, I would like to thank my mom, my dad, and my brothers (Muthu and Ganesh) who gave me a lot of support. I also want to give special thanks to my friends, Vijay Kumar, Naresh Pillarikuppam, and Krishna Chinthakayala, who were well-wishers for me.

Finally, I want to thank my family members who helped me to achieve this dream. Without them, I could not have achieved my dream.
# TABLE OF CONTENTS

ABSTRACT ......................................................................................................................... iii

ACKNOWLEDGMENTS .......................................................................................................... iv

LIST OF TABLES ................................................................................................................... ix

LIST OF FIGURES ................................................................................................................ x

CHAPTER 1. INTRODUCTION ................................................................................................. 1

1.1. What Needs to be Coordinated in an Emergency Response System? ................. 2

1.2. Best Mechanism that Fits the Decision-Making Requirement ......................... 3

1.2.1. Why Fuzzy Logic? ........................................................................................................ 3

1.2.2. Fuzzy Logic in an Emergency Response System .................................................. 4

1.2.3. Advantages of Fuzzy Logic ....................................................................................... 4

1.2.4. Disadvantages of Fuzzy Logic ................................................................................... 4

1.3. Which Technology is Better to Fit the Coordination Mechanisms? ............... 4

1.3.1. Java Agents .................................................................................................................. 4

CHAPTER 2. LITERATURE REVIEW ....................................................................................... 5

CHAPTER 3. SOFTWARE REQUIREMENTS ........................................................................... 12

3.1. NetBeans IDE ............................................................................................................... 12

3.2. NetBeans Installation ................................................................................................. 13

3.2.1. Steps to Install NetBeans ......................................................................................... 13

3.3. Technical Requirements ............................................................................................. 14

3.3.1. JADE .......................................................................................................................... 14

3.3.2. Concepts of JADE ..................................................................................................... 14

3.3.2.1. JADE Containers ................................................................................................. 14
5.5.2. Class Diagram

5.6. Fuzzy Rules

5.6.1. One Event Occurred

5.6.2. Two Different Events Within a 30-Mile Range

5.6.3. Two Different Events with More than a 30-Mile Range

5.6.4. Two Different Events Within a 30-Mile Range with Different Priority Levels

5.6.5. Two Different Events with More than a 30-Mile range with Different Priority Levels

5.6.6. Two Same Events Within a 30-Mile Range

5.6.7. Two Same Events with More than a 30-Mile Range

5.6.8. Two Same Events Within a 30-Mile Range with Different Priority Levels

5.6.9. Two Same Events with More than a 30-Miles Range with Different Priority Levels

5.6.10. Three Different Events with Three Different Priorities

CHAPTER 6. IMPLEMENTATION

6.1. Software Agent

6.2. ERS Visualization

6.3. Emergency Response System

6.3.1. Selection Panel

6.3.2. Input Panel

6.3.3. Display Dispatched Units

CHAPTER 7. EVALUATION

7.1. Reliability
CHAPTER 8. FUTURE WORK AND CONCLUSION .................................................. 59

8.1. Future Work ............................................................................................. 59

8.2. Conclusion ............................................................................................... 60

REFERENCES .................................................................................................. 61
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Fuzzy Rules</td>
<td>22</td>
</tr>
<tr>
<td>2. Fuzzy Rules Analysis for ERS</td>
<td>31</td>
</tr>
<tr>
<td>3. Fuzzy Rules Consistency</td>
<td>49</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Emergency Response System Architecture</td>
<td>26</td>
</tr>
<tr>
<td>2. Flow Diagram</td>
<td>28</td>
</tr>
<tr>
<td>3. Class Diagram</td>
<td>30</td>
</tr>
<tr>
<td>4. Jade Remote Management GUI</td>
<td>42</td>
</tr>
<tr>
<td>5. Emergency Response System Tool interface</td>
<td>43</td>
</tr>
<tr>
<td>6. Emergency Response System Tool Interface with One Event</td>
<td>45</td>
</tr>
<tr>
<td>7. Emergency Response System - Dispatched Units for Event</td>
<td>46</td>
</tr>
<tr>
<td>8. Emergency Response System - Dispatched Units for Two Events</td>
<td>47</td>
</tr>
</tbody>
</table>
CHAPTER 1. INTRODUCTION

The Emergency Response System is comprised of the first responders who need to attend immediately after the incident occurs [2]. An emergency could be natural or man-caused. Calloway and Keen (1996) define the emergency response system as a “combination of Information Technology and the Agents which are responsive” [2]. With that said, we can imagine how complicated it would be to coordinate actions effectively between various organizations, both public and private.

Bringing together information technologies and the emergency response system [2] identifies seven threats as the basis for managing emergency responses in addition to the forces of nature. The identified seven threats are (1) human, (2) biological, (3) nuclear/radiological, (4) incendiary, (5) chemical, (6) explosive, and (7) cyber-attacks [2] against information and data systems from Illinois Homeland Security Summit, 2002. These seven threats form a categorization for incidents that emergency response should address. Any emergency may be considered to a percent wise of fire emergency, medical emergency and police emergency. All these emergency responding systems should be coordinated to be effective during an incident.

In this research, we mainly spotlight the three public organizations that would need to react to any of the above-mentioned incidents: (1) fire fighter, (2) medical emergency, and (3) the police department. We need to consider six vehicles for these three emergency types. We should support at least two emergency events simultaneously. The response units should be dispatched to the next emergency event once the response units are done with one event. Response units should know the location of the emergency event and should be able to serve the nearest emergency event if the second emergency is far away from the response-unit location. When an emergency occurs, the critical questions that need to be answered would be actions and
organizations that need to be coordinated, how the coordination is achieved, and what software
would fit to implement the support[2].

1.1. What Needs to be Coordinated in an Emergency Response System?

When an incident occurs, the organizations that need to be involved, the actions that need
to performed, and the activities that need to be coordinated depend on the type of incident,
magnitude of the incident, damage caused by the incident, location of the incident (urban or less-
populated area), and jurisdiction where the incident occurred. With a higher magnitude for the
incident and more damage, the number of organizations, both public and private, involved
increases, thus the decision-making process becomes more complicated for decisions about
which organization needs to do what tasks, which tasks should be coordinated among these
organizations, which available resources need to serve the incident, and how many resources are
needed.

Within an emergency-response organization, the employees have their own policies,
reporting structures, roles, and procedures [3]. Also, there may be temporary volunteers who
might be pulled in based on the requirements. Employees from various organizations may not
have coordinated before, and this fact makes the coordination more complicated.

Decision making becomes more complicated when there are multiple incidents at the
same time in an area or when an incident happens and does not pertain to a single legal
jurisdiction. When there are multiple incidents, decisions such as the resources needed at each
incident and which jurisdiction should take what resources become more ambiguous and
difficult.

In an emergency, the damage can be minimized only if all the above decisions are made
quickly and with efficiency. If the emergency-response organizations are not flexible enough to
move their units based on the incident requirements, the damage could be done before the organizations respond. Hence, decisions should be quick and effective to handle an incident.

1.2. Best Mechanism that Fits the Decision-Making Requirement

In this paper, fuzzy logic is used as the decision-making mechanism. Decision making in an emergency depends on various factors such as the intensity of the incident, the nature of the incident, the location of the incident, the resources available to serve the incident, dependencies, the effect on the surroundings, and many other factors. To decide based on n number of factors becomes more complex while, in many cases, fuzzy logic proves to be efficient.

1.2.1. Why Fuzzy Logic?

Fuzzy logic was first used by Dr. Lofti Zadeh from the University of California at Berkely during the 1960s. Fuzzy logic [1] is a problem-solving algorithm for a wide range of systems. The system might be small or large. All these systems use fuzzy logic. There is lot of hardware and software that use fuzzy logic. The systems reach a final decision based on the inputs that are provided to the fuzzy logic systems. Fuzzy logic uses values ranging between 0 and 1. Fuzzy logic works similar to the human brain. As humans take data from different resources, we come to a final conclusion with information from all of these sources. Fuzzy logic [4] works with objects. All the objects are considered as one set. Let us consider this set of objects as K. If k1 is subset of set k, then we will have a function which maps the element k1 by using numbers. These numbers range from 0 to 1. Some characteristics of fuzzy logic are as follows:

- The conclusion is seen as a result of adaptable limitations [5].
- Any system which works logically can use fuzzy logic.
1.2.2. Fuzzy Logic in an Emergency Response System

Fuzzy logic [6] is used specifically to deal with the public problems. Fuzzy logic helps us to deal with problems where the given situation has uncertain information. In these types of condition, the Emergency Response System tool, with the help of fuzzy logic, is used for decision making.

1.2.3. Advantages of Fuzzy Logic

- The main advantage of fuzzy logic is that it is user friendly. Even though the user is not an expert, it can interpret with the user [5].
- Fuzzy logic expresses the information which is uncertain in a regular way [5].

1.2.4. Disadvantages of Fuzzy Logic

- A system which uses fuzzy logic needs more testing than a system which does not use fuzzy logic [8].
- Fuzzy systems are more expensive to develop because they take a lot of resources from development to testing.

1.3. Which Technology is Better to Fit the Coordination Mechanisms?

1.3.1. Java Agents

With any given incident, multiple organizations need to respond. For operations to be efficient most of the time, these organizations need to coordinate their activities and should be cooperative. For some situations, organizations may need to work collaboratively based on the intensity and the nature of the incident. In this paper, each organization which needs to serve the purpose is considered a single agent.
CHAPTER 2. LITERATURE REVIEW

Research done by Stella Ying Shen and Michael J. Shaw, is one of the complex studies that delves into all the complications that exist during an emergency response. Identify the tasks that need to be coordinated as well as the coordination mechanisms and information technology that need to be used for coordination. They consider the Incident Command System (ICS) employed by the National Fire Academy and many other public-safety professionals. This paper considers eight agents, with each agent representing a unit (firefighter, policemen, emergency medical services, the mayor’s office, the Environmental Protection Agency, public health, the Red Cross, and federal agencies) that needs to respond, and matches these eight agencies with thirteen tasks (dispatching upon a 911 call, firefighting, controlling traffic, investigating hazardous material, caring for victims, preparing media statements, confirming resource access, analyzing air samples, notifying hospitals, preparing shelter and food, requesting additional fire engines, requesting federal aid, and investigating weapons). The Emergency Response System decides and sends out the units that need to respond, unlike the above research the first responding units provides information to the central controlling center to communicate with the responding units to send them.

A study done by NaimKapucu [9] analyzes the needs of an emergency response system based on the September 11, 2001, attack. This research considers a broader aspect of the information needed for an emergency response system instead of triggering the response units as proposed in this paper. Research by NaimKapucu [9] considers the coordinated interorganizational structures and trust between the public and the private sectors. It considers the manner in which emergency agencies respond the response activities they take. This research considers the organizations that were involved and that need to respond as nodes of interaction.
This theory considers the general properties (robustness, fullness, redundancy, and rapidity) identified by the Multidisciplinary Center for Earthquake Engineering Research (MCEER). This research relies on social-network means sharing of data among organizations to build trust and interoperability, unlike the idea proposal for using a centralized unit-triggering system.

A model,” which was proposed by Timothy Schoenharl, Greg Madey, Gabor Szabo, and Albert–Laszlo Barabasi and which was given the name wiper system [10], is intended to provide information to the first responders and emergency planners, but does not decide which units need to respond to the emergency. The proposal consists of multiple agents. First, the data source measurement layer (also called the Real Time Data Source [RTDS]) collects data from one or more cell-phone providers and converts the data into a summarized format before it passes to the next agent. Second, the Detection Simulation and Prediction (DSP) agent determines whether the summarized information that is received is an emergency event and proposes what actions need to be taken, passing on this information to the Decision Support System (DSS) agent. Finally, the DSS is a front end for the wiper system which the first responders, public-safety personal and emergency planners can access by using a web browser. This system leaves the final decision about which units should respond for the first responder’s team. This research is one of the closest to the idea proposed, except that the wiper system can be considered as an enhancement as a data source for the system in this paper which decides the units that need to respond.

Research” [11] by Lili Yang, Raj Prasanna, and Malcolm King is about the information that needs to be available for the responders but lacks the triggering mechanism in the current research. Research by Lili Yang, Raj Prasanna, and Malcolm King [11] mainly talks about the amount and accuracy of the information that needs to be available for emergency responders. Also, the research is done about how and what information needs to be shared among
organizations as well as the On Site Security Control emergency information system. The study also discusses the Radio Frequency Identification (RFID) technologies for communication. The research [11] does not discuss which agents to send to which emergency.

Rui Chen, Raj Sharman, Nirupama Chakravarti, H. Raghav Rao, and Shambhu J. Upadhyaya studied the challenges for interoperability among the first emergency response team regarding information sharing in their research [12] which mainly tackles the communication challenges but is not involved with the decision process like the Emergency Response System does. They considered the difficulties faced through hardware, middleware, the application layer compatibilities, and other budget restrictions for all the response systems to adapt to a single, centralized system. They came up with a data model, a standard guideline that needs to be followed by all emergency response systems. They developed an xml data model for better and easy interoperability between various systems. This study [12] is very close to this research because the research [12] considers intelligent response agents. These response agents in the research [12] consider incident type, scene security, location details, incident level (severity), and with who (authorized personal) the information should be shared. It does not apply this information to any decision process, whereas the current research triggers units to respond to the emergency.

Paper [13], proposed and developed by Raymond J. Menard and Curtis E. Quady, is a web-based application that is accessible through any standard internet browser. The application provides information about the emergency situation but does not send out the units as the Emergency Response System does. A system based on browser means that it is compatible and accessible from any system. Based on an auto trigger (an emergency trigger system like an alarm) at a site (physical building) or a call, the central database is updated with the data that is
directly accessible by various emergency responders. The system developed by Raymond J. Menard and Curtis E. Quady excludes any intermediate public-safety answering points and, thus, shares the information with the authorized responders quickly. The system developed by Raymond J. Menard and Curtis E. Quady does not have the information about the units that need to respond. This application can be attached as a predecessor with the Emergency Response System which decides what units need to respond based on the emergency.

Paper [14] proposed By Felipe Meneguzzi, Jeanoh, Nilanjan Chakraborty, Katia Sycara, Siddarth Mehrotra, and Micheal Lewis is based on XML which provides all the required data except the triggering mechanism as the Emergency Response System does. Anytime Cognition: An Information Agent for Emergency Response (ANTICO) domain is developed to have all activities that need to be considered during an emergency response. For each activity, ANTICO provides observable features and the information needed for that activity (for example, pulling the phone numbers of all people who need to be contacted). The agent recognizes a user’s current activities. At the same time, the agent should be proactive based on the data updates received from the incident site. The agent is designed in six segments: Observer, Intent predictor, Cognitive Workload estimator, Information gatherer, Information adapter, and Information Presenter. The agent is designed with a graphical user interface (GUI) for presentation to the final user which contains a map and all other required information. This proposal, which almost decides everything from who to call and how many units are required based on the incident intensity, is the closest to the current research except that it does not trigger the response units by itself.

“A Decision Support System for Disaster Management in Buildings” [7] uses the Jade framework for decision process designed by Avgoustinos Filippoupolitis and Erol Gelenbe. This
system also considers a multi-agent model for decision making with the information available to various agents. This design assists people in evacuating the building quickly by suggesting the closest possible exit in case of an incident by monitoring the incident from inside. This system does not send any information to first responders, whereas the Emergency Response System triggers the first-responding units.

“Multi-Agent Crisis Response Systems” [16] assists the decision maker with planning and determining resources to use in an emergency but does not send out the units by itself as the Emergency Response System does. The paper [16] is about the DEFACTO FIRE GRID response system. This research considers the filtering and data-fusion methods of the received data, decision making and the adaptive nature of the system, and interaction and information sharing among multiple responders. DEFATCO is a user-centric system which gains and draws valuable lessons that will be applied in the real world. DEFACTO gets its initial information from in-building sensors that are installed at the incident site. This system is not a real-time adaptive one because it does not consider the units that are already in use. Whereas the proposed Emergency Response System has a record of all available and unavailable units.

Increasing demand for the emergency response systems and decreasing capacity are due to technical aspects, but are also due to the ambiguous relationship between various government and private organizations that does not involve in the decision-making process as the Emergency response system. This aspect of the emergency response is explained in the research done by Louise K. Comfort, Kilkon Ko, and Adam Zagorecki [17]. Research by Louise K. Comfort, Kilkon Ko, and Adam Zagorecki [17] considers the magnitude of the disaster, the number of jurisdictions involved with the response decision making, and simple cooperation factors that influence the response efficiency. Network theory is used to identify the core information that
needs to be exchanged between organizations. Also, this paper [17] demonstrates the effect of one organization’s failure on another organization’s efficiency. This research states that the type and quality of the initial response as well as the degree of coordination among the involved response units would affect the response system’s efficiency.

This theory [17] finally states that there is an inverse relationship to the initial disaster magnitude and a positive relationship to the initial amount of resources available with the efficiency of the emergency response system. This research is a good study to consider as a future aspect of my proposed emergency response system.

The research by Dr. Sc. Ksenija Culo and Dr. Sc. Vladimir Skendrovic [18] clearly discusses the advantages and efficiency of fuzzy logic in decision making. This paper [18] clearly states how the decision-making parameters are considered in the fuzzy logic effectiveness. Generally, when a decision is based on a single parameter, it is pretty straightforward. Decision making gets more complex when multiple parameters need to be considered and when some parameters are contradictory. Fuzzy logic handles these multi-parameters by giving weight to each parameter based on its importance for the decision being made. Thus, fuzzy numbers are achieved based on the weight and rating of the considered parameters. As the decision is made for these fuzzy numbers, decision making is more appropriate and effective. This paper clearly states the effectiveness of fuzzy logic for decision making which is implemented in my proposed idea.

“Integrated Peer-to-Peer Applications for Advanced Emergency Response Systems” [19] is similar to this proposal. A graphical interface was proposed to handle issues with voice-oriented communication, situation information for awareness, and interoperability constraints. This proposal [19] talks about the need for computer-based communication rather than prevailing
voice-based communication which cannot provide various other aspects such as video feeds, conferencing with personnel at other organizations, statistical data, and others. Also, this proposal talks about the interoperability limitations among various organizations involved in a situation. This proposal [19] does not decide on the technology that needs to be used and also proposes a graphical interface for information sharing rather than decision making unlike the current model.
CHAPTER 3. SOFTWARE REQUIREMENTS

In this section, the paper discusses the software requirements and frameworks used to develop the Emergency Response System. NetBeans is the Integrated Development Environment IDE used to develop/run the application. Java Agent Development Environment (JADE) is the framework used to build the application-level components. Software agents are the building blocks for the core application functionality.

3.1. NetBeans IDE

IDE refers to integrated development environment and is the platform that facilitates application developers to write their code, compile for any errors, and run the program. NetBeans is an open-source IDE. NetBeans is an independent programming language and allows computer programmers to code in a wide variety of choices, ranging from Java, PHP, Groovy, C to several other languages [20]. The NetBeans IDE works well on all major operating systems (OS), Windows, Linux, and Solaris, that are compatible with the Java Virtual Machine (JVM) [20]. NetBeans IDE provides programmers with a wide verity of features to control and modify the application effectively.

The factors that separate NetBeans from other competitive leaders for integrated development environments are the better support available for hanging technology trends and the ability to adapt to the latest Java enhancements before other IDEs. NetBeans provides an easy way to develop the user interface in no time with many rich, high-quality features and a wide range of tools to build the interface [25].

The following features and the effective functionalities of NetBeans favor it as the framework to develop the Emergency Response System:

- Works on the fly, an open-source IDE.
• Rich, user-interface builder.
• The local history feature allows tracking the recent changes to a file.
• The NetBeans profiler helps investigate memory usage.
• Powerful debugging capabilities.
• The bookmarking feature of the code.
• Allows making a file favorite, which is a very useful feature for a big project with many classes and large code bases.
• Customizable code formatting and single file-compilation features.
• Extensible feature of NetBeans allowing developers to add their own features.
• Features helpful in optimizing the application performance and scalability.

3.2. NetBeans Installation

The latest version of NetBeans available for application developers is 7.2.1. It is an easy and simple process to download and install NetBeans. The program lets you choose between a regular features install and a customized install, allowing you to pick the features and bundles provided by NetBeans.

3.2.1. Steps to Install NetBeans

• Get the NetBeans installer from http://netbeans.org/downloads/index.html to your system.
• Run the executable .exe to choose the type of installation process and to complete the install process.
• Details about the installation - http://netbeans.org/community/releases/71/install.html
3.3. Technical Requirements

To understand the technical details of the Emergency Response System, we need to have basic knowledge/have answers for the following questions: What is a software agent? How is a software agent created and destroyed? How do software agents communicate to accomplish a certain task? In the following sections, we will discuss the need for JADE and how it is related to the software agent’s communication.

3.3.1. JADE

JADE [21] stands for Java Agent Development Framework. JADE is an agent-development framework built with the underlying principles of Java in compliance with the Foundation for Intelligent Physical Agents (FIPA) specification that facilitates multiple agents’ communication. JADE is more of a middleware to develop an agent-based communication system. JADE is one of the current, promising agent-development frameworks with an extensible agent-development model and predefined features provided to programmers in order to build a rich, agent-centric application.

In simple terms, JADE is a platform to accomplish tasks by software agents or agents communication. It can also be called asynchronous agent programming. The latest available version is JADE 4.2.0.

3.3.2. Concepts of JADE

3.3.2.1. JADE Containers

An active instance of JADE is known as a container and there could be more than one agent in each container [15]. In JADE, there should at least one container that is always active (the main container), and all other containers register with the main container the moment they start/become active. A pair of active containers would make a JADE platform for agent
communication. In order to accomplish these responsibilities, the main container holds two special agents [15].

- The Agent Management System (AMS) is responsible for the naming service, maintaining the uniqueness of software agents, and also creating and destroying agents as needed.
- The Directory Facilitator (DF) is responsible for Yellow-Pages type services and helps the agent to find another in order to achieve agent communication.
- Command to launch main container java -- –cp<classpath>jade.Boot –gui

3.3.2.2. Agent Communication

In JADE, agent communication plays a vital role in building an effective agent-centric application, and agent communication is accomplished by asynchronous message transfers between agents. These messages follow the standards specified by FIPA by implementing Agent Communication Language (ACL) [15].

- ACL is implemented by the jade.lang.acl.ACLMessage class
- The ACLMessage class takes care of handling the java bean methods of all the required fields.
- In JADE, all agent communications are message based.
- When a message is passed by an agent, it is placed into the receiver agent’s message queue and is left for the program to process after picking up the message.
3.3.3. **Software Agents**

Implementation of software-agent communication has grown enormously in distributed computing research and has evolved into a new paradigm in the software industry. In the application-development field, a software agent is a computer program that acts on behalf of a user in an intelligent manner and performs the tasks as programmed. Based on the way the software program or an agent has been programmed, it can be categorized as

- **Autonomous agents** – are the ones which are intelligent enough to perform a set of tasks with minimal user interaction.
- **Distributed Agents** – are the ones which will be implemented on physically distinct machines to operate,
- **Multi-agent systems** – are the ones which perform the task interacting with other agents,
- **Mobile agents** – agents which are capable of relocating themselves to achieve the objectives.

Multi-agent systems have more than one software agent carrying out a specific task. In the Emergency Response System, we implement the multi-agent system to accomplish tasks as per the message passing between agents. Implementation of software agents is well-suited for applications that involve communication between different components and distributed commuting.

3.3.3.1. **Implementing Software Agents**

In JADE, the software agent can be implemented by a class extending the jade.core.Agent class and defining the setup() method which takes care of agent initialization. Agents are identified by an instance of the jade.core.AID class which gives the identifier of a specific agent [15].

16
In order to compile an agent, the following command needs to be executed on jade runtime: `Javac –classpath sampleAgent.java`. The compiled agent can be executed with the following command: `Java –classpath .jade.BootSampleAgent`.

Once the agent accomplishes the task, it will not terminate by itself. It will still be running; in order to terminate the agent, the `doDelete()` method needs to be called. The `takedown()` method takes the responsibility of a clean-up agent operation before termination.
CHAPTER 4. CONSIDERATIONS AND TERMINOLOGY

When an emergency incident occurs, there is a need for many public and private organizations to work directly or indirectly as first responders. What organizations are involved is based on various factors, including the nature of the incident, the intensity of an incident, the location of the incident, and other factors.

4.1. Kinds of Emergencies

In the real world, we have different emergencies that occur at different locations. The emergencies are of three types.

- Fire Emergency
- Medical Emergency
- Police Emergency

4.1.1. Fire Emergency

An emergency is considered as a fire emergency when there is fire hazard at some location.

4.1.2. Medical Emergency

An emergency is considered as medical emergency when someone is suffering from an illness.

4.1.3. Police Emergency

A police emergency might be considered the emergency when there is a quarrel among a group of people.

The response units that are considered in this paper are as follows:

- Fire Truck
• Ambulance
• Police Car

In the real-world scenario, humans cannot calculate the probability for dispatching the appropriate emergency unit for the appropriate emergency. In this paper, we design an ERS tool which helps us to dispatch emergency response units very efficiently. The ERS tool keeps track of all emergency response units, and it will dispatch the emergency response units based on the criteria in the next section.

4.2. Types of Priority Levels

4.2.1. High Priority

For a fire emergency, a fire truck is considered as a high priority because a fire truck serves as the entire purpose for the fire emergency.

4.2.2. Medium Priority

For a fire emergency, a police car may not be needed. With this emergency type, a police car is considered a medium priority.

4.2.3. Low Priority

For a police emergency, we do not need a fire truck. In this scenario, a fire truck is considered a low priority.

4.2.4. Distance Factor

The factor that needs to be considered during an emergency is the distance factor. Suppose, for example, fire emergency 1 occurred around 30 miles from the response unit, and there is a similar type of fire emergency 2 around 1 mile away. The distance of the emergency response unit to serve emergency 1 might require more time, and by the time the fire truck arrives at fire emergency 1, there might be more fatalities. In these types of cases, instead of
serving fire emergency 1, the fire truck should be dispatched to fire emergency 2 which has the same priority level. Distance factor is achieved by using the coordinates’ x and y axis in the tool.
CHAPTER 5. DESIGN

This chapter briefly discusses the design that is used to build the ERS tool. We build the tool which is agent-to-agent communication. Agent-to-Agent communication is achieved by using the Java Agent Development Environment.

5.1. Inputs that Are Needed for the ERS Tool

Types of Emergency: We need to create emergencies using the ERS tool. Emergencies can be created at random places. We only consider three types of emergencies. Multiple fire emergencies can be created. Similarly, multiple medical and police emergencies can be created using the ERS tool. By creating multiple emergency events we are mimicking the real-time world where emergencies occur randomly at different places.

5.1.1. Priority of the Emergencies

Emergencies can be divided into three categories. They are as follows:

- High priority
- Medium Priority
- Low priority

The user can create a fire emergency with a priority option that is one of the three types. A fire emergency with a high priority indicates that the particular fire event has more priority than another fire emergency.

5.2. Weights Needed for the Vehicle

Weight is another factor that user can input using the ERS tool. The user can decide which type of vehicle is important for the specific emergency event.

For a fire emergency, a fire truck has more importance than a police car and an ambulance. For this situation, the user will give the weight for a fire truck as 1; for an
ambulance, the user will give the value as .9; and for a police car, the weights will be a little less than the ambulance, which will be 0.8 in this case. The user can decide on these values. Based on these weights, the ERS tool will send the response units efficiently. Fuzzy logic values will be from 0 to 1. Let us assume the weights for the response unit, based on emergency type, are considered as shown in Table 1.

**Table 1. Fuzzy Rules.**

<table>
<thead>
<tr>
<th>Type of Emergency</th>
<th>Vehicle Needed</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fire Emergency</td>
<td>Fire Truck</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Ambulance</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>Police Cars</td>
<td>0.5</td>
</tr>
<tr>
<td>Medical Emergency</td>
<td>Ambulance</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Police Cars</td>
<td>0.8</td>
</tr>
<tr>
<td>Public Safety</td>
<td>Police Cars</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Ambulance</td>
<td>0.5</td>
</tr>
</tbody>
</table>

From Table 1, it can be understood that

- IF fire emergency, THEN send fire truck, ambulance, and police cars
- IF medical emergency, THEN send ambulance and police cars
- IF public safety, THEN send police cars and ambulance
5.3. **Inputs that Are Not Needed for the ERS Tool**

Users need not give the location of the fire event which was created by the user. The tool will recognize the location of the emergency event that was created. Users need not know the location of the response unit where it was created.

5.4. **Proposed Design**

5.4.1. **Rules**

- For each emergency event, the weights for each response unit must be defined prior to dispatching the units.
- For each emergency event, the priority levels must be defined prior to dispatching the response units.

The weights must be defined for each emergency event so that the tool dispatches the response units based on the weights. The weight informs the tool about the necessity of the response unit for the event created.

The priority levels must be defined for each emergency event so that the tool understands the event severity. Based on the severity defined, the tool dispatches the response unit accordingly.

5.4.2. **Assumptions**

For the implementation of the proposed design, the following assumptions are considered:

- In the emergency events set, only three kinds of events are considered:
  - Fire Emergency
  - Public Safety
  - Medical Emergency
• In the Emergency Response Units set, two vehicles for each response unit have been considered.
  • Fire Truck
  • Ambulance
  • Police Car
• When setting the priority levels for an emergency event, three levels are considered.
  • High - for the event that needs immediate attention
  • Medium - for the event that needs moderate attention
  • Low - for the event that needs less attention
• For the weights, it is assumed that the maximum weight for a response unit is 1, and the minimum is 0. The weight cannot exceed 1.
• Once the Event Clear button is clicked, it is assumed that the emergency event has been served successfully.
• Once the Clear All button is clicked, it is assumed that all emergency events have been served successfully.

5.5. Requirement Gathering of the Tool

In this section, the design of the Emergency Response System is explained. The tool design starts with gathering the requirements that are needed to develop the tool. The following requirements must be met by the tool in order to represent the system.
• The purpose of the tool is to recognize the kind of event that was created by the user.
• The tool must be able to accept any of the three events that are defined.
• The tool must be able to accept the weights that are set for an event by the user.
• The tool must be able to accept each unit’s priority levels that are set for the respective event by the user.
• The tool shall dispatch the units with an Attend call from the user.
• The tool shall display the event that is being attended by the units.
• The tool must display the details of the units attending the event name.
• The tool must be able to clear the event when the user selects the Clear Button call.
• The tool must be able to clear all the events when the user clicks the Clear All button.
• The tool must be capable of dispatching the vehicles to the specific event required by the user.
• The system must be able to recognize the event location.
• The system must be able to calculate the distance between the event and the origin.

A ERS tool which was designed using the fuzzy logic technique attends the desired or all the events and dispatches the response units accordingly. The architecture of the Emergency Response System approach is shown in Figure 1.
First, JADE should be running prior to the Emergency Response Tool. Once we run JADE, then we need to load a couple of configurations. From Figure 1, it is clear that one configuration will be the Remote Agent management, which is the main container, and the other is Agent 2, the Emergency Response Unit’s container. Remote Agent Management is the default configuration.

Agent 2, the Emergency Response unit’s container, has two vehicles for each of the three response types, and these three vehicles are agents. These three agents are response units which are named the fire truck, police car, and ambulance. Once we run the tool with the Emergency Response Unit configuration, the Emergency Response System will generate a GUI for the user.
Then, the user can start creating events with the GUI’s help. Once the GUI is loaded, there are three inputs that the user needs to be provide to the GUI.

The first step will be the type of event that the user wants to create. Here, we are considering three event types. From the available emergency events, the user can pick one emergency, and he can click the graph. Once the user clicks the graph, the emergency event is created at that location. In the Emergency Response System, each emergency event is considered an agent.

The weights set by the user to each emergency vehicle for each emergency event is another input. The Emergency Response system is built in such a way that the user can decide which emergency vehicle has more priority than the other response units for each emergency type. If the user gives the fire truck a weight of 1 for a fire emergency and if the user gives the fire truck a weight of 0.8 for a medical emergency, then the importance of the fire truck is more for a fire emergency compared to a medical emergency. Once the user creates multiple events, he can select the required emergency event from the list of available events, and he can set the weights for the fire truck, police car, and ambulance.

Setting the priority levels for emergency events is another input the user should provide to the Emergency Response System. When the user picks a fire emergency, the ERS tool has an option to add the priority level for the emergency event. The user can create multiple fire emergency events that have different priority levels. The user can also create multiple emergency events with the same priority level. The Emergency Response System is more flexible. The Emergency Response system has three priority levels. One will be high; another will be medium; and the next will be low.
Once the user inputs all the values, the last option is the Attend All button. Once the user clicks Attend All, all the response units will be dispatched to the emergency events based on the priority, distance, and the weights provided by the user. If the user wants to create more events than he should, again, follow the steps from step one, and he can click Attend All for the response unit to serve that emergency request.

5.5.1. Flow Diagram

Below figure 2 explains the flow of the Emergency Response Tool.

![Flow Diagram]

**Figure 2. Flow Diagram.**
Figure 2. Flow Diagram (continued).
5.5.2. Class Diagram

Below Figure 3 shows all the classes and methods and their relationships that are used for the Emergency Response Tool implemented in this paper.

Figure 3. Class Diagram.
5.6. Fuzzy Rules

From the Table 2 we have defined the fuzzy rules. Based on these fuzzy rules, the ERS will dispatch the emergency vehicles.

Table 2. Fuzzy Rules Analysis for ERS.

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weightage</th>
<th>Decision</th>
<th>Response Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FE</td>
<td>1</td>
<td>120</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td>2 Different events within a 30-mile range</td>
<td>FE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>100</td>
<td>FT</td>
<td>0.8</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.8</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td>2 Different events with more than a 30-mile range</td>
<td>ME</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>ND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>1</td>
<td>170</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>ND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td>2 Different events within a 30-mile range with different priority levels</td>
<td>FE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>0.5</td>
<td>100</td>
<td>FT</td>
<td>0.8</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.8</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td>2 Different events with more than a 30-mile range with different priority levels</td>
<td>FE</td>
<td>0.5</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>170</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
</tr>
</tbody>
</table>
Table 2. Fuzzy Rules Analysis for ERS (continued).

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weightage</th>
<th>Decision</th>
<th>Response Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Same events within a 30-mile range</td>
<td>PE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>1</td>
<td>100</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.5</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.8</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td>2 Same events with more than a 30-mile range</td>
<td>ME</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.7</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>150</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td>2 Same events within a 30-mile range with different priority levels</td>
<td>FE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>0.5</td>
<td>100</td>
<td>FT</td>
<td>0.9</td>
<td>D</td>
<td>FT2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.7</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.6</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td>2 Same events with more than a 30-mile range with different priority levels</td>
<td>FE</td>
<td>0.5</td>
<td>110</td>
<td>FT</td>
<td>0.9</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>1</td>
<td>170</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.9</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.7</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td>3 Different events with 3 different priorities</td>
<td>FE</td>
<td>1</td>
<td>130</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>0.8</td>
<td>140</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.4</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>0.5</td>
<td>150</td>
<td>FT</td>
<td>0.3</td>
<td>D</td>
<td>FT2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.9</td>
<td>D</td>
<td>AM2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV1</td>
</tr>
</tbody>
</table>
In Table 2, AM1 stands for Ambulance1; AM2 stands for Ambulance2; CV1 stands for Police Vehicle1; CV2 stands for Police Vehicle2; FT1 stands for Fire Truck1; FT2 stands for Fire Truck2; FE stands for Fire Emergency; ME stands for Medical Emergency; PE stands for Police Emergency; FT stands for Fire Truck; AM stands for Ambulance; CV stands for Police Vehicle; 1 stands for High Priority; 0.8 stands for Medium Priority; 0.5 stands for low priority; D stands for Dispatched; and ND stands for Not Dispatched.

5.6.1. One Event Occurred

From the figure 2, let us consider a scenario with one event. In this scenario, we create one fire emergency event, and we assign the priority level. Let us assign the priority level as 1, a high-priority event, and after that, we will assign the weights to the emergency response events. Let us assign 1 for a fire truck, 0.8 for an ambulance, and 0.5 for a police vehicle. Once we assign the weights and click Attend All, Fire Truck1, Ambulance1, and Police Vehicle1 are dispatched to the fire emergency event.

5.6.2. Two Different Events Within a 30-Mile Range

In this scenario, create two different events, one being a medical emergency event and the other being a fire emergency. If you refer to Figure 1, we create a fire emergency with a high priority at location 110. For a fire emergency, we now assign the weights as 1 for a fire truck, 0.8 for an ambulance, and 0.5 for a police vehicle. Next, a medical emergency is created at location 100 with the priority level as 1. For this medical emergency, we assign the weight as 0.8 for a fire truck, and 1 for an ambulance, and 0.8 for a police vehicle. Once we assign the weights and click Attend All, Fire Truck1, Ambulance1, and Police Vehicle2 are dispatched to the fire emergency, and Fire Truck2, Ambulance1, and Police Vehicle1 are dispatched to the medical emergency. Because the ambulance and police vehicle have high weights, the first available
units, FT1 and CV1, are sent to the medical emergency. Because the fire truck weight is high for the fire emergency, the first available resource, which is FT1, is sent to the fire emergency. Because the weight for the ambulance and police vehicle is less compared to the medical emergency, AM2 and CV2 are dispatched to the fire emergency.

5.6.3. Two Different Events with More than a 30-Mile Range

In this scenario, we create one medical-emergency event and one public-safety event. We assign the priority level as 1 for the medical emergency and 1 for the public-safety event. We create the medical emergency at location 110 and we create the police emergency at location 170. The distance between these two emergency events is 60. Once we create the events, we need to assign weights to the vehicles for these two events. For the medical emergency, we assign the weight as 0 for the fire truck, 1 for the ambulance, and 0.5 for the police vehicle. For the police-emergency event, we assign the weight as 0 for the fire truck, 1 for the ambulance, and 0.5 for the police vehicle. Once we give the weights, we click Attend All. Then, Ambulance1 and Police Vehicle1 are dispatched to the medical emergency, and Ambulance2 and Police Vehicle2 are dispatched to the police emergency. Because the weight for the police vehicle is the same for both a police emergency and a medical emergency and because the police emergency is far away, serving the medical emergency is more effective in this situation, so the first available resources, AM1 and CV1, are dispatched to the medical emergency; the next available resources, AM2 and CV2, are dispatched to the police emergency.

5.6.4. Two Different Events Within a 30-Mile Range with Different Priority Levels

In this scenario, we create two different events with varying priority levels, one being a fire emergency with a priority of 1 and the other being a medical emergency with a priority of 0.5. The fire emergency is created at location 110, and the medical emergency is created at
location 100. Now, we assign the weights. For the fire emergency, we assign 1 to the fire truck, 0.8 to the ambulance, and 0.5 to the police vehicle. For the medical emergency, we assign 0.8 to the fire truck, 1 to the ambulance, and 0.8 to the police vehicle. Once we assign the weights, we click on Attend All. Then, FT1, AM1, and CV1 are dispatched to the fire emergency, and FT2, AM2, and CV2 are dispatched to the medical emergency. Because the fire emergency has a priority higher than the medical emergency, all the first-available resources are sent to the medical emergency, and the rest of the response units are sent to the fire emergency.

5.6.5. Two Different Events with More than a 30-Mile range with Different Priority Levels

In this scenario, we will create two emergency events, one being the fire emergency with a priority of 0.5 and the other being the medical emergency with a priority of 1. The fire emergency is created at location 110, and the medical emergency is created at location 170. Now, we assign the weights. For the fire emergency, the priority is 1 for the fire truck is 1, 0.8 for the ambulance, and 0.5 for the police vehicle. Now, the weights for the medical emergency are assigned. For the fire truck, the priority is given as 1; for the ambulance, the priority is given as 0.8; and for the police vehicle, the priority is given as 0.5. Once we assign the weights, we click on Attend All. Then, the response units are dispatched in the manner shown in Figure 1. FT1, AM1, and CV1 are dispatched to the fire emergency, and FT2, AM2, and CV2 are dispatched to the medical emergency. From the fuzzy rules table, it is clearly understood that, even though the medical emergency has a high priority, all the first-available response units are dispatched to the fire emergency which is very close to the response unit’s location. Because the
medical emergency is far away and all the response units have the same priority, serving the closest emergency event pays off better than serving an emergency event which is farther away.

5.6.6. Two Same Events Within a 30-Mile Range

In this scenario, we consider two similar events. From Figure 1, let us consider the police emergency event. The first police emergency event is created with a priority level of 1 and at location 110. Once we create the first emergency event, we assign its weights. Second, the police emergency event is created with a priority level of 1 and at location 100. Once we create the second emergency event, we assign is weights. For the first police emergency event, fire truck was assigned a priority of 0; ambulance was assigned a weight of 0.8; and police vehicle was assigned a weight of 1. For the second police emergency event, fire truck was assigned a priority of 0; ambulance was assigned a weight of 0.5; and police vehicle was assigned a weight of 0.8. Once we assign the weights, we click on Attend All. Then, all response units are dispatched as shown in Figure 1. Because fire truck has no priority, the fire truck will not be dispatched to either of these emergency events. Because ambulance and police vehicle have a higher priority for the first emergency and because the distance between both events is less than 30, the response units are dispatched to the event which has a higher weight. In this scenario, the first emergency event has a higher weight for the first police emergency event. The first available response units, AM1 and CV1, are dispatched to the first police emergency, and the next response units, AM2 and CV2, are dispatched to the second police emergency.

5.6.7. Two Same Events with More than a 30-Mile Range

In this scenario, we create two similar events. The distance between these two events is greater than 30. For this scenario, let us consider two identical medical emergencies. The first medical emergency event is created at location 110, and the priority for this medical emergency
event is assigned as 1. Now, we assign the weights for the response units. For the fire truck, let us give it a weight of 0; for the ambulance, let us assign the weight as 0.7; for the police car, we assign the weight as 1. Once we assign the weights, we click on set weights. Now, we create a second medical emergency event at location 150, and the priority for this medical emergency event is 1. Now let us assign weights for the response units for the second emergency event. For the fire truck, let us assign the weight as 0; for the ambulance, let us assign the weight as 0.8; and for the police car, let us assign the value as 1. Once we assign weights, we set weights. Once we set the weights, we click on Attend All. Once we click attend all, all the response units are dispatched in this manner. Because the police vehicle has the same priority for both emergency events and the distance between these events is greater than 30 miles, the first police vehicle, CV1, is sent to the first medical emergency. Because the priority of the ambulance for the second emergency event is greater than the first medical emergency event, the first medical response unit, AM1, is dispatched to the second medical emergency event instead of going to the first medical emergency event. The fire truck is not sent to any events because it does not have any priority.

5.6.8. Two Same Events Within a 30-Mile Range with Different Priority Levels

In this scenario, we consider two identical fire emergency events at different locations. The distance between these two locations is less than 30. Let us create the first fire emergency event at location 110, and we assign the priority as 1. After assigning the priority, we set the weights for response units. Let us assign 1 for the fire truck, and 0.8 for the ambulance, and 0.5 for the police car. Once we assign, we click on set weights. Now, the second fire emergency event is created at location 100, and we assign the priority as 0.5; once we assign the priority, we need to set weights for the response units. For the fire truck, let us assign 0.9; for the ambulance,
let us assign the weight as 0.7; and for the police car, let us assign the weight as 0.6. Once we assign the weights, we need to click on set weights. Once we set the weights for both emergency events, we need to click on attend all. Once we click attend all, the available response units are dispatched in this manner. Because the distance between the first emergency and the second emergency is not greater than 30, the tool checks for the event priority. In this scenario, the first fire emergency event has a higher priority than the second fire emergency. Therefore, all the first available response units are dispatched to the first fire emergency event; FT1, AM1, and CV1 are dispatched to the first fire emergency event. Then, the other available response units are dispatched to the second emergency event. FT2, AM2, and CV2 are dispatched to the second fire emergency event which has a priority as 0.5.

5.6.9. Two Same Events with More than a 30-Miles Range with Different Priority Levels

In this scenario, we consider two identical fire emergencies created at different locations. The distance between these two locations is greater than 30. Let us create the first fire emergency event at location 110, and we assign the priority as 0.5. After assigning the priority, we need to set the weights for response units. Let us assign 0.9 for the fire truck, 0.8 for the ambulance, and 0.5 for the police car. Once we assign weights, we click on set weights. Now, the second fire emergency event is created at location 170, and we assign the priority as 1; once we assign the priority, we set the weights for the response units. For the fire truck, let us assign a weight of 1; for the ambulance, let us assign the weight as 0.9; and for the police car, let us assign the weight as 0.7. Once we assign the weights, we click on set weights. Once we set the weights for both emergency events, we click on attend all. Once we click on attend all, the available response units are dispatched in this manner. Because the distance between the first emergency and
second emergency is greater than 30, the tool does not check for the event priority. Instead of checking the priority, the tool checks the distance, and all the first-available response units are dispatched to the first fire emergency. FT1, AM1, and CV1 are dispatched to the first fire emergency event. Because the distance between the two emergency events is greater than 30, serving the first fire emergency event, which is closer, is efficient. Once the first-available response units are dispatched to the first fire emergency, the other available response units are dispatched to the second emergency. FT2, AM2, and CV2 are dispatched to the second fire emergency event which is at location 170.

5.6.10. Three Different Events with Three Different Priorities

In this scenario, we create three different events at varying locations. The distance between these events is less than the 30-mile range. First, we create a fire emergency event at location 130 and assign the priority level as 1. Next, we create a medical emergency at location 140 and assign the priority as 0.8. Then, we create a police emergency at location 150 and assign the priority as 0.5. Once we create all three events, we set the weights of the response units for each event individually. For the fire emergency, we assign 1 to the fire truck, 0.8 to the ambulance, and 0.5 to the police car. For the medical emergency, we assign 0 to the fire truck, 1 to the ambulance, and 0.4 to the police car. For the police emergency, we assign 0.3 to the fire truck, 0.9 to the ambulance, and 1 to the police car. Once we assign weights, we click on set weights for each event separately. Once we set the weights, we click on attend all. Once we click on attend all, the response units are dispatched in this manner. Because all these events occur nearby and because all the events have different priorities, the tool considers the weights of the vehicles to dispatch efficiently. Because a fire truck has a weight of 1 for the fire emergency and 0.3 for the police emergency, the first fire response unit, FT1, is dispatched to the fire
emergency, and fire truck 2 is dispatched to the police emergency because it has a weight of 0.3.

Now consider the ambulance. Because the ambulance has a weight of 1 for a medical emergency and 0.9 for a police emergency, the first available ambulance, AM1, is dispatched to the medical emergency, and the second available response unit, AM2, is dispatched to the police emergency. Now consider a police car. Because a police car has a weight of 1 for a police emergency and 0.5 for a fire emergency, the first available police car, CV1, is dispatched to the police emergency, and the second available response unit, CV2, is dispatched to the fire emergency.
CHAPTER 6. IMPLEMENTATION

6.1. Software Agent

In the proposed idea, multi-agent systems (MAS) can be used, allowing collaboration with other agents to process the entire response system. Every response unit is an intelligent software agent with considerable flexibility that responds to the emergency requests by complying with the emergency type rules. A MAS performs the operation, which might be difficult to do with single agent, by collaborating with other agents. Programming agents can perform tasks according to the context’s requirements. They perform the task continuously in a flexible manner, responding to environmental changes without requiring any human intervention. The reason for preferring the software agents is that agents can work autonomously with the ability to control their own actions to be performed, collaborative with the ability to communicate with other agents effectively, responding in a timely manner acting smartly and sufficiently flexible in their actions.

The above architecture can be designed based on peer-to-peer communication on a network layer. The system can be built using Java on JADE (Java Agent Development Environment). Java is used as the core programming language for this response system because it is portable, object oriented, robust, multi-threaded, secure, and has high performance.

6.2. ERS Visualization

In this section, the graphical user interface implementation of the tool is explained. The Emergency Response System accepts the emergency events, event types, event weights, and weights of each response unit for the respective event. The output of the Emergency Response System tool is a frame that contains the dispatched units for the events that occurred.
The Emergency Response System Frame contains the type of event defined as a list on the left side; the event-creation panel is on the top right; and the weight input is on the bottom right of the panel. The Emergency Response System is developed by considering each event and response unit as an agent. To achieve the agent-based development, JADE is considered for implementation. The tool is integrated with the JADE application to achieve the agent-based development. The JADE application has to be run before loading the Emergency Response System. JADE can be loaded as “default config” by giving the main class as “jade.boot” and the Arguments as “–gui.” Once the JADE is loaded, the emergency response units can be loaded. After the default config is run, the GUI of JADE Remote Agent Management is displayed as shown in Figure 4.

Figure 4. Jade Remote Management GUI.
6.3. Emergency Response System

After the JADE system is loaded, the Emergency Response System is loaded by giving the Main Class as “jade.boot” and the arguments as “-container-agents FireTruck1:emergency.responseunits.FireAgent;Ambulance:emergency.responseunits.MedicalAgent;PoliceCar:emergency.responseunits.PoliceAgent;MainEvent:emergency.events.EventsMain. The config can be defined as “All Response Units”. When running the All Response Units Config, the system loads the Emergency Response System GUI panel. The panel for the system is shown in Figure 5.

![Emergency Response System GUI panel](image)

**Figure 5. Emergency Response System Tool Interface.**
6.3.1. **Selection Panel**

In the tool’s Selection panel, the Type of Emergency and the Level of Priority are defined. Both inputs are defined as list items. The available emergency types are fire emergency, police Safety, and medical Emergency. The available types of priority levels are high, medium, and low. The user selects the required emergency type and priority level before creating the event.

6.3.2. **Input Panel**

Once the event is created, the user sets the weight of each response unit for the event created. The tool’s Input panel is on the top, right side of the tool. Once the user creates the event, the list near Type of Event Occurred is updated with the event name. The user then selects the event name from the list. After the event is selected, the distance of the event from the origin is displayed. Now, the user enters the desired weights of each response unit for the event. Figure 6 shows the layout of the Input panel with the event created and the values set.
Figure 6. Emergency Response System Tool Interface with One Event.

In Figure 6, a Fire Emergency Event is created with high priority. Then, the user selects the event created for assigning the weights. When selecting the event from the list, the distance of the event is displayed along with the event’s X and Y coordinates. This X and Y coordinates provides the position of the event in the graph from the origin. The user enters the weight for each response unit and clicks on Set Weights.

6.3.3. Display Dispatched Units

The user has set the weights for the events, and then, the next step is to attend the event. To let the system know about the units to dispatch for the event created, the Attend/Attend All
button has to be clicked. The system looks for events based on the distance and then priority, and the response units are dispatched.

In this section, we go through one scenario. First, the user creates a fire emergency event and assigns the priority level as high, and then, the user assigns the weights for that fire emergency. The user gives the weight for the fire truck as 1; and for the ambulance, the user assigns a weight of 0.5; and for the police car, the user assigns a weight of 0.3. After assigning the weights, the user clicks on Attend All. From Figure 7, it is clear that all response units, AM1, CV1, and FT1, are dispatched to the fire emergency.

![Figure 7. Emergency Response System - Dispatched Units for Event.](image.png)
Next, the user creates a medical emergency with medium priority; then, the user assigns the weights for response vehicles. For the fire truck, the weight is 0.3; for the ambulance, the weight is 1; and for the police car, the weight is 0.6. Once the user clicks on Attend all, it is clear (from Figure 8) that, because first-response units are already dispatched to the fire emergency, the next available response units, AM2, CV2, and FT2, are dispatched to the medical emergency. In the figure 8, the red circle is the fire truck while the blue circle indicates an ambulance. A black circle indicates a police car.

Figure 8. Emergency Response System - Dispatched Units for Two Events.
CHAPTER 7. EVALUATION

In this section the Emergency Response System is evaluated. In a given area, consider the Emergency Response Units are located at location x1 and then an emergency event occurred at some location x2. The distance from location x2 to x1 is (d2-d1). Consider there might be same type of emergency event occurred at location x3 then the distance between location x3 and x2 is (d3-d2). The distance between location x3 and x1 will be (d3-d1).

Distance in Savings = ((d2-d1) + (d3-d1)) - (d3-d2)

Consider a fire emergency event occurred at location 110 which is considered as x2. The distance between the Response Units and the emergency event is d2 -d1 which is 110 - 0 = 110. Let us assume that another same type of event occurred at location 130 which is considered as x3. The distance between the First emergency event and the second emergency event is d3-d2 which is 130-110 = 20. The distance between the second emergency event and the initial location of the response units is d3 -d1 which is 130 - 0 = 130.

Total savings in distance is ((d2-d1) + (d3-d1)) - (d3-d2) i.e., (110 + 130) - 20 = 220

7.1. Reliability

The consistency of Emergency Response Tool is done with taking with all the fuzzy rules. We check whether the Emergency Response Tool’s fuzzy rules fire as expected. We have different scenarios, and we test the tool for the scenarios listed in Table 3.
Table 3. Fuzzy Rules Consistency

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FE</td>
<td>1</td>
<td>120</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
<td>CV1 should be dispatched</td>
</tr>
<tr>
<td>2 Different events within a 30-mile range</td>
<td>FE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM2</td>
<td>AM2 should be dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
<td>CV2 should be dispatched</td>
</tr>
</tbody>
</table>
Table 3. Fuzzy Rules Consistency (continued)

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Different events within a 30-mile range</td>
<td>ME</td>
<td>1</td>
<td>100</td>
<td>FT</td>
<td>0.8</td>
<td>D</td>
<td>FT2</td>
<td>FT2 should be dispatched</td>
<td>FT2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>100</td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>110</td>
<td>CV</td>
<td>0.8</td>
<td>D</td>
<td>CV1</td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Different events with more than a 30-mile range</td>
<td>ME</td>
<td>1</td>
<td>110</td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>1</td>
<td>110</td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>1</td>
<td>170</td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM2</td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
<td>CV2 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td>No of Events Occurred</td>
<td>Event Type</td>
<td>Priority Level</td>
<td>Distance from Origin</td>
<td>Vehicle Type</td>
<td>Weight</td>
<td>Decision</td>
<td>Response Unit</td>
<td>Expected Results</td>
<td>Output</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>--------------</td>
<td>--------</td>
<td>----------</td>
<td>----------------</td>
<td>------------------</td>
<td>--------</td>
</tr>
<tr>
<td>2 Different events within a 30-mile range with different priority levels</td>
<td>FE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
<td>FT1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>0.5</td>
<td>100</td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV1</td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>FT</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FT2</td>
<td>FT2 should be dispatched</td>
<td>FT2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>0.5</td>
<td>100</td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM2</td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV2</td>
<td>CV2 should be dispatched</td>
<td>CV2 is dispatched</td>
</tr>
</tbody>
</table>
Table 3. Fuzzy Rules Consistency (continued).

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Different events with more than a 30-mile range with different priority levels</td>
<td>FE</td>
<td>0.5</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
<td>FT1 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
</tr>
<tr>
<td>ME</td>
<td>1</td>
<td>170</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT2</td>
<td>FT2 should be dispatched</td>
<td>FT2 is dispatched</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
</tr>
<tr>
<td>2 Same events within a 30-mile range</td>
<td>PE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
</tr>
</tbody>
</table>
Table 3. Fuzzy Rules Consistency (continued).

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Same events within a 30-mile range</td>
<td>PE</td>
<td>1</td>
<td>110</td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV1</td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>1</td>
<td>100</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.5</td>
<td>D</td>
<td>AM2</td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.8</td>
<td>D</td>
<td>CV2</td>
<td>CV2 should be dispatched</td>
<td>CV2 is dispatched</td>
</tr>
<tr>
<td>2 Same events with more than a 30-mile range</td>
<td>ME</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>150</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV2</td>
<td>CV2 should be dispatched</td>
<td>CV2 is dispatched</td>
</tr>
</tbody>
</table>
Table 3. Fuzzy Rules Consistency (continued).

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Same events with more than a 30-mile range</td>
<td>PE</td>
<td>1</td>
<td>100</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>0.5</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.8</td>
<td>D</td>
</tr>
<tr>
<td>2 Same events with more than a 30-mile range</td>
<td>ME</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>0.7</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV1</td>
</tr>
<tr>
<td>ME</td>
<td>1</td>
<td>150</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV2</td>
</tr>
</tbody>
</table>
Table 3. Fuzzy Rules Consistency (continued).

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Same events within a 30-mile range with different priority levels</td>
<td>FE</td>
<td>1</td>
<td>110</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
<td>FT1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
<td></td>
<td></td>
<td></td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
<td></td>
<td></td>
<td></td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>0.5</td>
<td>100</td>
<td>FT</td>
<td>0.9</td>
<td>D</td>
<td>FT2</td>
<td>FT2 should be dispatched</td>
<td>FT2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>AM</td>
<td>0.7</td>
<td>D</td>
<td>AM2</td>
<td></td>
<td></td>
<td></td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.6</td>
<td>D</td>
<td>CV2</td>
<td></td>
<td></td>
<td></td>
<td>CV2 should be dispatched</td>
<td>CV2 is dispatched</td>
</tr>
</tbody>
</table>
Table 3. Fuzzy Rules Consistency (continued).

<table>
<thead>
<tr>
<th>No of Events Occurred</th>
<th>Event Type</th>
<th>Priority Level</th>
<th>Distance from Origin</th>
<th>Vehicle Type</th>
<th>Weight</th>
<th>Decision</th>
<th>Response Unit</th>
<th>Expected Results</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Same events with more than a 30-mile range with different priority levels</td>
<td>FE</td>
<td>0.5</td>
<td>110</td>
<td>FT</td>
<td>0.9</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
<td>FT1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>D</td>
<td>AM1</td>
<td></td>
<td></td>
<td></td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV1</td>
<td></td>
<td></td>
<td></td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>1</td>
<td>170</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT2</td>
<td>FT2 should be dispatched</td>
<td>FT2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>AM</td>
<td>0.9</td>
<td>D</td>
<td>AM2</td>
<td></td>
<td></td>
<td></td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.7</td>
<td>D</td>
<td>CV2</td>
<td></td>
<td></td>
<td></td>
<td>CV2 should be dispatched</td>
<td>CV2 is dispatched</td>
</tr>
<tr>
<td>No of Events Occurred</td>
<td>Event Type</td>
<td>Priority Level</td>
<td>Distance from Origin</td>
<td>Vehicle Type</td>
<td>Weight</td>
<td>Decision</td>
<td>Response Unit</td>
<td>Expected Results</td>
<td>Output</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>--------------</td>
<td>--------</td>
<td>----------</td>
<td>---------------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>3 Different events with 3 different priorities</td>
<td>FE</td>
<td>1</td>
<td>130</td>
<td>FT</td>
<td>1</td>
<td>D</td>
<td>FT1</td>
<td>FT1 should be dispatched</td>
<td>FT1 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.8</td>
<td>ND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.5</td>
<td>D</td>
<td>CV2</td>
<td>CV2 should be dispatched</td>
<td>CV2 is dispatched</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>0.8</td>
<td>140</td>
<td>FT</td>
<td>0</td>
<td>ND</td>
<td>AM1</td>
<td>AM1 should be dispatched</td>
<td>AM1 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>1</td>
<td>D</td>
<td>AM1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>0.4</td>
<td>ND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>0.5</td>
<td>150</td>
<td>FT</td>
<td>0.3</td>
<td>D</td>
<td>FT2</td>
<td>FT2 should be dispatched</td>
<td>FT2 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AM</td>
<td>0.9</td>
<td>D</td>
<td>AM2</td>
<td>AM2 should be dispatched</td>
<td>AM2 is dispatched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CV</td>
<td>1</td>
<td>D</td>
<td>CV1</td>
<td>CV1 should be dispatched</td>
<td>CV1 is dispatched</td>
</tr>
</tbody>
</table>
Consistency of the tool was made on the expected results versus the actual output results from the tool. From Table 3, all scenarios have been tested, and the actual output from the tool matches the expected result.
CHAPTER 8. FUTURE WORK AND CONCLUSION

8.1. Future Work

The current Emergency Response System is not capable of handling the traffic information about the route from where the response unit is dispatched to the emergency event. We should consider the traffic situation because traffic plays an important role for dispatching the response units.

The current Emergency Response System is designed for three response-unit vehicles. We consider more response units to attend more events occurring at the same time so that all events will be covered in the future.

Also, we can consider responders other than the firefighters, police department, and medical emergency based on the nature of the emergency. For example, a chemical accident would require chemical experts at the scene to handle the situation. In this way, various organizations can be considered for notification to respond automatically.

Integrate with sensor networks installed at the incident location to trigger the first responders automatically. A lot of study is being done with robots and sensor networks [7], such as a camera installed on small robots for initiating triggers for first responders and assisting them during their service at the location. Robots can be big and small so that they can get into the building before the first responders so that the responders have a good view of what they need to do.

Data mining is another important direction we can take with further research. A central database can be installed with all the data from the earlier incidents. The data contain how many resources were used for an incident and the various other resources required which can be accessed by the Emergency Response System for decision making.
“Design Principles of Coordinated Multi-Incident Emergency Response System” [7] comes up with an activity-based methodology for effective performance of first responders during an incident. This research considers various aspects, such as the sequence of the activities that need to be performed, the role of each responder, the time delay for message communication and getting ready, coordination and resource allocation, redeployment, and exhaustion. This proposal [7] considers each activity as a sequence of steps and, hence, recognizes which step is mandatory before the other activity. Integration of this activity-based model adds more accuracy and effectiveness to the current system. This change can be a good consideration for future research.

8.2. Conclusion

In this paper, we discussed some of important emergency events that occur in the real world. We were able to create six response units. We created three types of emergency events. We were able to simultaneously dispatch the response units to two emergency events. All these responses were able to serve the nearest emergency event if the second emergency event is further away from the response-unit location. We also designed the Emergency Response Tool for dispatching the Response Units efficiently using fuzzy logic.
REFERENCES


