Warsaw University of Technology

Agent System and Application

Crowd Evacuation System Final Report

Authors:
Abba Umar, Andra Umoru, Swetha Kumar

June 26, 2023

Contents

List of Figures

1 Introduction

A crowd evacuation system is a collection of policies, procedures, and equipment created to make it easier for many people to leave a particular area safely and quickly in the case of an emergency. A crowd evacuation system's main objective is to reduce the possibility of accidents or fatalities by managing and directing the flow of people while also giving people clear and useful instructions. Numerous elements, such as emergency exits, emergency lighting, signage, alarms, and communication systems, are frequently included in crowd evacuation systems. To increase the effectiveness and efficiency of the evacuation process, they might also combine technologies like video surveillance, real-time monitoring, and simulation models.

The demands and features of the crowd, including their age, physical condition, and knowledge of the location, are considered while designing a crowd evacuation system. Additionally, it considers elements like the emergency's nature, the area's size and layout, and the personnel and resources that are on hand. The success of a crowd evacuation system depends on effective planning, training, and testing. While ongoing maintenance and upgrades can help ensure that the system remains efficient and up to date, regular drills and exercises can help pinpoint areas for improvement.

1.1 Crowd Evacuation Simulation

A computer-generated model that models the movement of people during an emergency evacuation is known as a crowd evacuation system simulation. By forecasting how people will act and how quickly they can leave the area, this simulation can assess the success of the crowd evacuation system. The simulation often makes use of information about the crowd's age, physical capabilities, and knowledge of the place, as well as information on the layout, size, and location of the area's exits and other resources. The type and intensity of the disaster, such as a fire, earthquake, or terrorist strike, are also taken into consideration.

1.2 Crowd Behaviour

Examining crowd behavior during emergencies is challenging as it typically necessitates subjecting real individuals to genuinely perilous surroundings. Although fire drills offer one method, they fall short of recreating the authentic conditions of panic, often resulting in people not taking them seriously. To overcome these limitations, a promising alternative lies in the development of sophisticated computational tools that account for the complexities of human and social behavior within a crowd. Such tools could serve as a practical substitute for studying crowd behavior in emergency situations.

1.3 Some Research Question

The research questions below were answered by our implementation.

- How do various evacuation tactics (such as phased evacuation or total evacuation) affect the length of time it takes to get a crowd out of the way and the number of casualties?
- What impact do environmental elements like exit placement and the existence of impediments have on how people behave during an evacuation?
- How do an individual's attributes, such as Health, mobility, and prior building knowledge, affect their evacuation behavior?

2 Design of CES system using GAIA

GAIA is a methodology for developing multi-agent systems using a role-centered approach. The GAIA output consists of a design still too abstract for implementation. To turn this around a more concrete design is produced using AUML. AUML takes the GAIA design as input and breaks it down into a series of diagrams that facilitates its implementation. In this section, we present the design of the CES system using the GAIA methodology. The system automates the task of rescuing and periodically notifying the system.

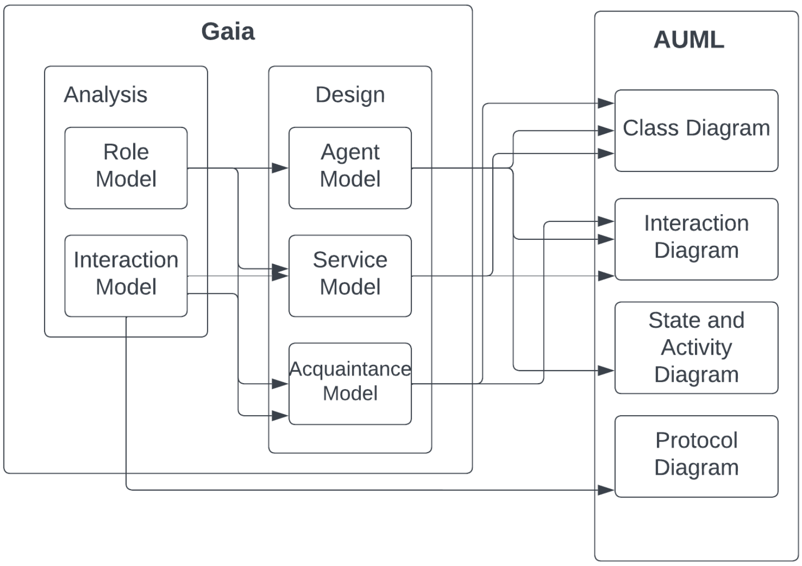

Figure 1: GAIA-AUML integration

2.1 Analysis Phase

The objective of the GAIA analysis phase is to develop an understanding of the system and its structure, without considering any implementation detail. The analysis consists of a role model, which identifies the key roles in the system, and an interaction model, which defines the interaction among the roles.

2.1.1 Identifying the Roles

A role can be thought of as an abstract description of how an entity is supposed to behave. In most cases, a role is associated with either a person, a department within an organization, or the organization itself.

We use an organizational vision in our Crowd Evacuation system, splitting it into the following departments: Information retrieval, central coordinator, and Human Management. Note that Database management is an external department of the system. We identify the members of each department, each member corresponding to a role as in the figure below.

DPT	Roles	Informal Description
Information Retrieval	People Detection	Detect and count the number of people in the environment
	Danger Alert	Sending alerting about hazard in an environment
	Exit Detection	Detecting of exit in simulation environment
Central Co-ordinator	Task Distribution	Coordinate and distribute task to other agents
Human Management	Rescuing	Rescuing people that are in danger in the simulation environment
	First Aid Treatment	Check health status and provide assistance

Figure 2: Roles in the System.

2.1.2 Interaction Model

The second step in the analysis phase corresponds to the identification and documentation of the protocols associated with each role. In order to facilitate the understanding of the protocol documentation we focus our attention on all the interactions of the Task Distributor, expressed as the three protocols: Danger, Detection, and Rescuing as shown below.

The danger protocol describes the interaction among the Task Distributor, Danger Alerter, and Environment in order to know what is the current status of the environment. Firstly, the Task Distributor(Iniator) interacts with Danger Alerter(receiver) in order to obtain the latest status of the environment. The Task distributor initiates this interaction with a waiting time as an input, which represents the maximum time allowed before beginning the next request to the Danger Alerter. Then the Danger Alerter requests a connection with the environment. Finally, the Danger Alerter provides the Task distributor with the latest information.

The Detection protocol describes how the interaction between the Task distributor and the People Detector allows for the detection and counting of people in the environment. Firstly, the Task Distributor makes a request to the People Detector. The people Detector detects and counts the people. Finally, the People Detector sends detected information to the Task Distributor.

The Rescuing Protocol shows the interaction between Task Distributor and Rescuer.

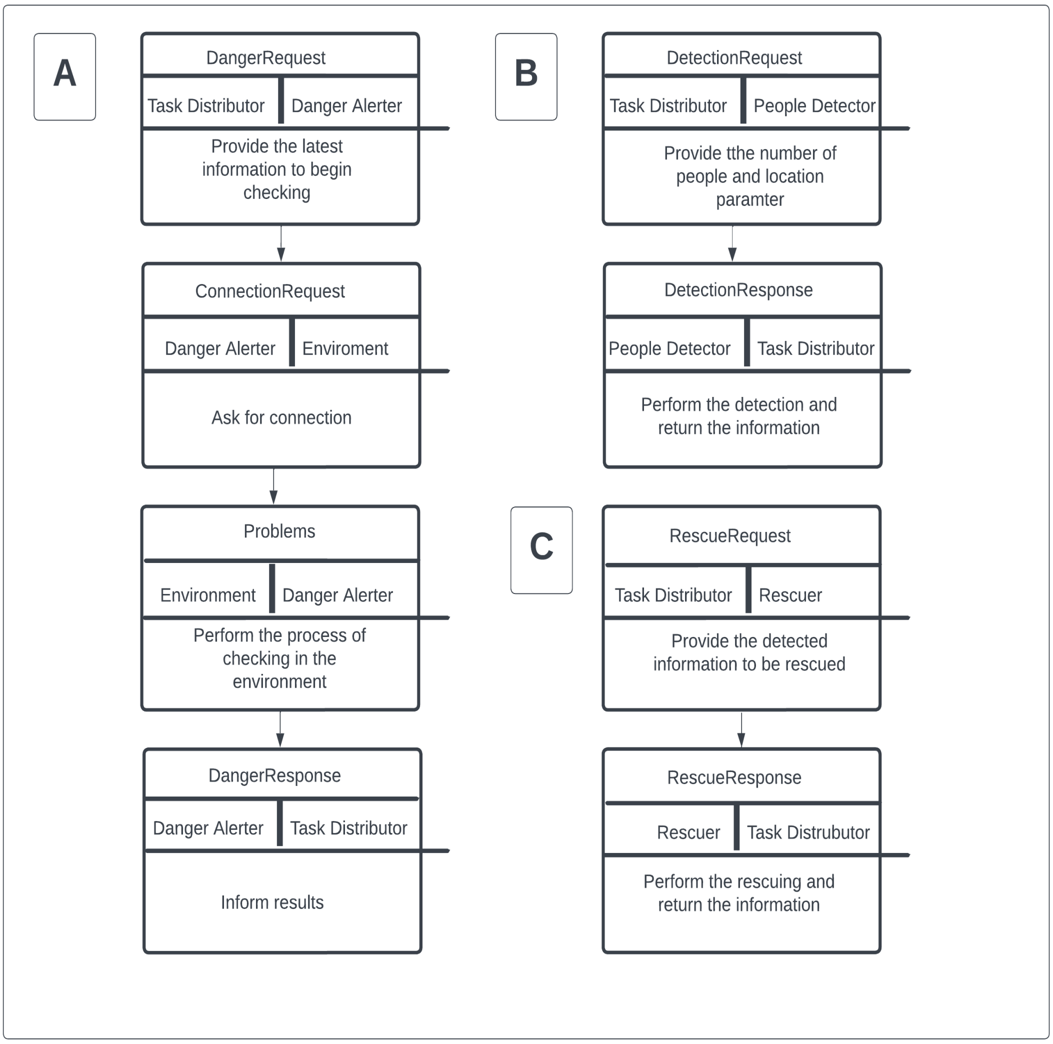

Figure 3: Protocols associated with Task Distributor.

Firstly, Task Distributor will assign the same number of Rescuers as the number of people counted by the People detector in a way that is useful for the application of the rescuing technique. Finally, the Rescuer notified the Task Distributor about the rescue operation and the person's status.

2.1.3 Role Model

The purpose of this stage is to generate an elaborated roles model, representing each role by a textual template that documents its permissions, responsibilities, protocols, and activities.

Task Distributor

Description :- This agent is responsible for assigning tasks to other agents. The Task Distributor constantly examines the current condition of the system and utilizes this information to decide how to assign tasks based on priority and available resources.

Responsibilities :-

Liveness : (CalculateState.UpdateDecisionAlgorithm)

Safety : The Task Distributor agent verifies that agents are capable of performing assigned tasks based on their priority and urgency.

Permissions :-

Read : The Task Distributor agent can read the priority and urgency of tasks and the capabilities of other agents

Change : The Task Distributor agent can assign tasks to other agents.

Activities :- CalculateState, QueryDF, AssignTasks

Protocols :- DangerRequest, DetectionRequest, RescueRequest, .

People Detector

Description :- This agent is responsible for detecting the presence and counting the number of people in the rescue simulation environment.

Responsibilities :-

Liveness : (ScanEnvironment.UpdateState)

Safety : Maintain uninterrupted interaction with the task distributor.

Permissions :-

Read : The state of the environment, including counting people.

Change : Its own state as it detects people.

Activities :- ScanEnvironment, NotifyTD, RegisterDF,

Protocols :- DetectionResponse, DetectionTime.

Rescuer Agent

Description :- This agent is responsible for rescuing people that are trapped within the emergency environment.

Responsibilities :-

Liveness : (RescuePeople.ProvideMedicalAssistance)

Safety : Ensure constant communication with other agents involved in the rescue operation.

Permissions :-

Read : The health status of individuals, Exit.

Change : Victim location data, Treatment assignments.

Activities :- RescuePeople,ProvideMedicalAssistance.

Protocols :- RescueResponse, RescueTime, PatientStatusResponse.

Danger Alerter

Description :- This agent is responsible for monitoring and detecting potential hazards in a rescue simulation.

Responsibilities :-

Liveness : (MonitorEnvironment.AlertAgent)

Safety : The Danger Alerter detects and alerts potential dangers to ensure safety.

Permissions :-

Read : Environmental[U+202F]conditions.

Change : None.

Activities :- MonitorEnvironment,AlertAgent, QueryDF.

Protocols :- DangerResponse, DangerTime.

2.2 Design Phase

The aim of the GAIA design phase is to transfigure the analysis models into a sufficiently low level of abstraction that traditional design techniques may be applied in order to implement the agents. The design process involves three models. The agent model identifies the agent types that will make up the system and the agent instances that will be instantiated from these types. The service model identifies the principal services that are required to accomplish the roles of the agent. Finally, the acquaintance model documents the communication links between the different agents.

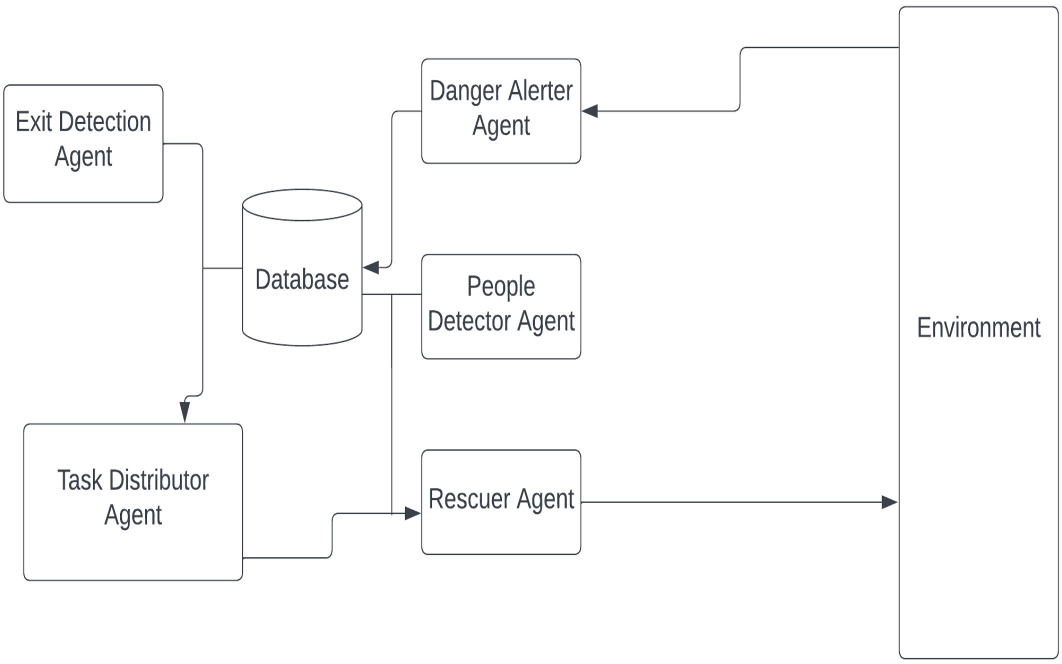

Figure 4: Agent Model.

2.2.1 Agent Model

The agent model aims to document the different types of agents that will be used in the system and the instances that will realize these types of agents at run time. It is defined by using a simple structure in the figure above.

2.2.2 Service Model

The service model indicates the services associated with the agent role and specifies the main properties of these services. The figure above illustrates the services associated with the Task Distributor.

The Start Check service defined from the Danger Protocol, is invoked when the wait Time has been completed and returned the latest information.

The Start Detection service, defined by Detection Protocol, requests a Detection service to the People Detector, obtaining the relevant information.

The Perform Rescue service defined by the Rescue Protocol, takes the people's profile in order to rescue and provide First AID.

Services	Inputs	Outputs	PreCondition	PostCondition
Start check	Latest Area	Latest Area,Status	waitTime=24hours,Area!=null	Area!=null&&Status=Updates
Start Detection	Detection Requirement ,Status	Detected Information	(Status!=null&& DRequirement!=null)&& compa(status,Drequire)=True	Detected Information!=incomplete
Perform Rescue	Detected Information	Rescue Information	Detected Information=incomplete	Rescue Information!=null

Figure 5: Agent Model.

2.2.3 Acquaintance Model

The final model of a GAIA design is the acquaintance model. This model simply defines the communication links that exist between the agent types. The figure below shows the interaction between different types of agents that make up the CES prototype.

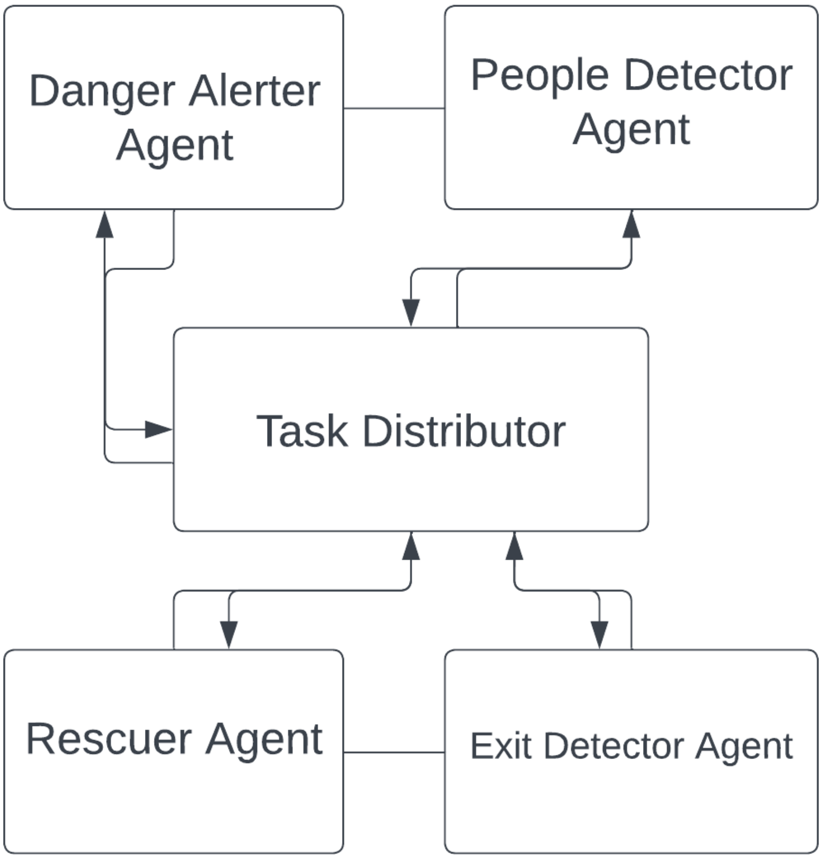

Figure 6: Acquaintance Model.

3 Towards a more concrete model with AUML

The GAIA outcome is an abstract design to be refined by traditional design techniques. We have found a way to further refine GAIA with AUML because it allows one to apply typical object-oriented properties of UML and new features for specifying other aspects of the agent interaction that is not covered by GAIA Cernuzzi (2004).

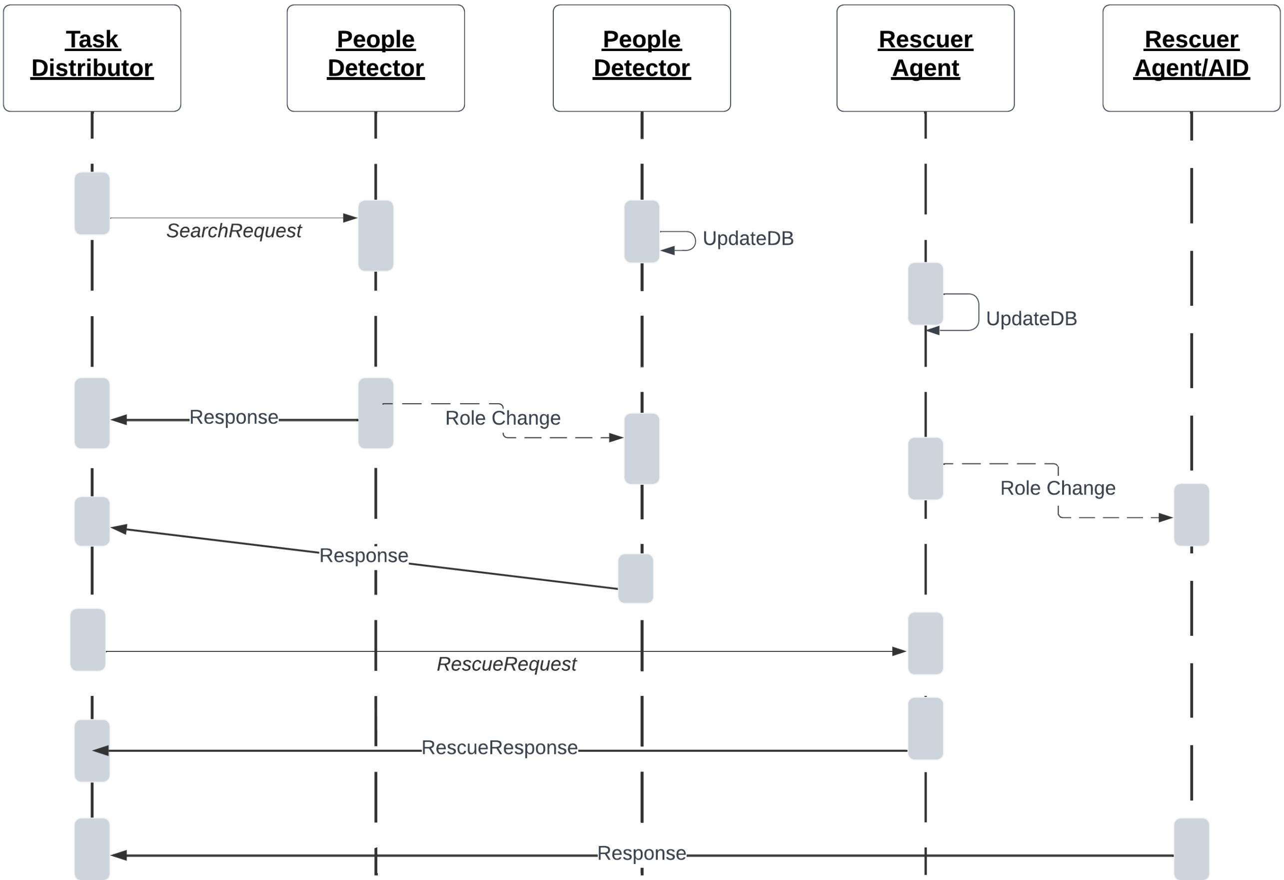

Figure 7: A sequence diagram showing interaction among agents in the system.

4 Model

The model has been created with Mesa, a Python-based ABM (agent-based modeling) framework, and it allows users to quickly create agent-based models using built-in core components (such as spatial grids and agent schedulers) or customized implementations; visualize them using a browser-based interface; and analyze their results using Python's data analysis tools. This model is an abstraction of rules occurring in the real world. The environment is a 50x50 grid of walls representing a floor plan. The agents consist of fire agents, danger alerter, rescuer, people detector, and task distributor.

4.1 Building Plan

The building plan consists of two lecture rooms, offices, and a conference room with one door each, leading to a corridor with two main exits.

Figure 8: The building plan design using Lucid.

The layout that will be used and focused on for the rest of this project, is one which attempts to replicate a single segment of the mini-building floor of the Department of Mathematics and Information Science. It contains the following interactive objects in the Mesa model.

- Human
- Furniture
- Door
- Exit
- Fire
- Smoke
- Lecture room

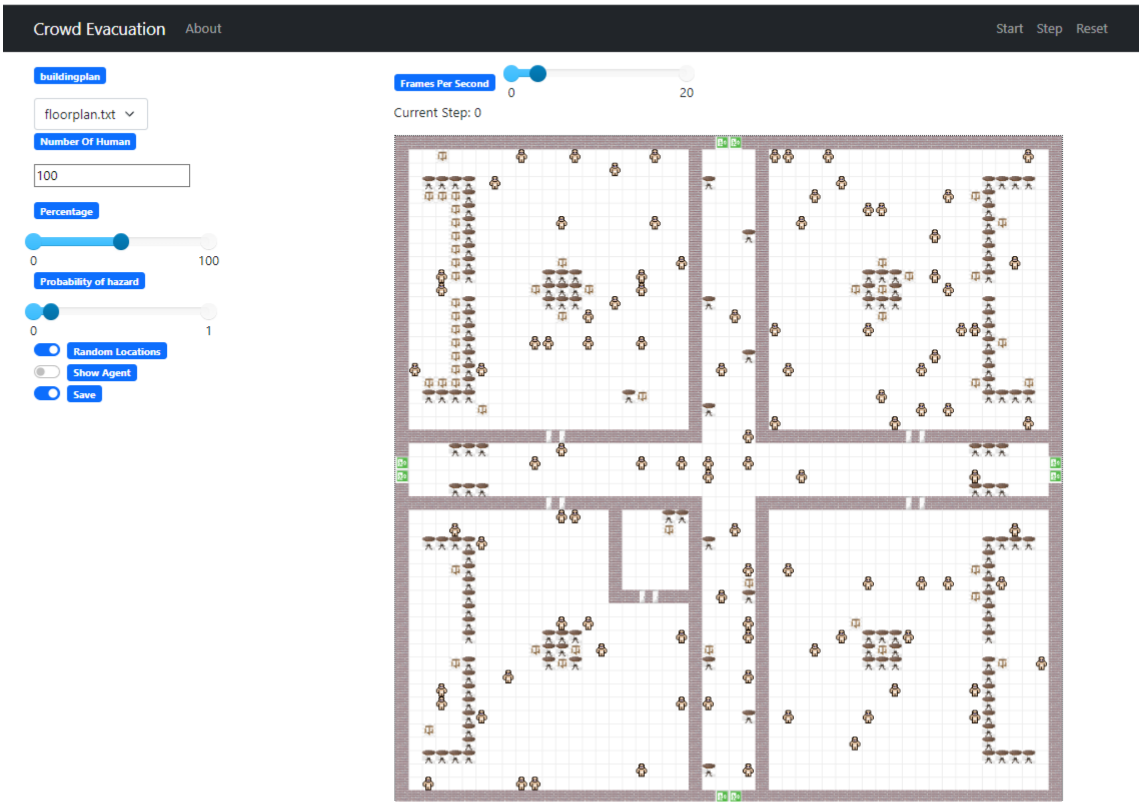

Figure 9: The building plan design using Lucid.

4.2 Building Object Attribute

Building object attributes refers to identifying and defining the characteristics or properties of an object. These attributes provide a description or set of features that help to distinguish and understand the object better. Object attributes can vary depending on the context and the specific domain of interest. All the agents and the building object use the following attributes.

Attribute	Data Type	Description
Flammable	Boolean	Indicates whether fire can spread to the given building object. If false, fire will not replicate this wall. Only Human and furniture are flammable.
Traversable	Boolean	If an object can be moved over, this value is true otherwise false.
Visibility	Integer	Represents how easily seen an object is, through the smoke of a fire.
Position	coordinates	Set of values that represent the position or location of a point in a given system or reference frame.
Smoke	Boolean	If an object can emit smoke in the environment, this value is true otherwise false.

4.3 Human

The attributes below take influence from those proposed by Fangqin (2008).

Attribute	Data Type	Description
Position	Coordinate (X, Y)	The coordinates of the human's current location.
Health	Decimal (0 to 1)	Represents the health of the human. Zero being dead, one being perfectly healthy and any decimal values in-between representing the severity of the health condition.
Mobility	Integer (0 to 2)	value to represent the current mobility state of a human. Zero indicates that they can not move, 1 indicates normal movement and 2 indicates panic movement.
Speed	Integer (0 to 2)	The number of tiles the human can move per step.
Shock	Decimal (0 to 1)	If the agent sees something which shocks them, this value increments accordingly and is used when deducing if the agent will enter panic mobility.
Nervousness	Integer (0 to 10)	Represents the agent's mental stability. This value influences how they behave when confronted with various scenarios.

4.4 Other Agents Attributes

These attributes collectively define the characteristics and capabilities of individual agents within a system, enabling them to act autonomously, make decisions, and interact with other agents or the environment based on their internal state and the external stimuli they receive.

4.4.1 Danger Alerter

Attribute	Data Type	Description
Believes Alarm	Boolean	Indicates whether or not, the agent will believe the fire is real when it first starts. This will change to true if they receive an alarm from danger alerter, or they see the fire/smoke.
Position	coordinates	Set of values that represent the position or location of a point in a given system or reference frame.

4.4.2 Rescuer Agent

Attribute	Data Type	Description
Rescue	Boolean	A value denoting whether or not an the agent is able to rescue others.
Rescue Count	Integer(1 to n)	A value keeping count of how many successful rescue the agent has made. As this value rises, the probability that the agent will help again decreases exponentially.

4.4.3 People Detector Agent

Proposed in JIN (1997)

Attribute	Data Type	Description
Visibility	Integer(1 to n)	The visibility of the agent when determining if the agent is visible from a given location, possibly through smoke .
Vision	Integer(1 to plan)	The range value of an extended More neighborhood, where the agent is the center point. Smoke has an impact on vision, according to previous research on visibility in fire smoke.

4.4.4 Fire Agent

Proposed by Pablo Cristian Tissera (2014) in the table below:

Attribute	Data Type	Description
Smoke Radius	Integer(1 to n)	The range value of an extended Moore neighborhood, where the fire is the center point. .
Smoke Spread Rate	Integer(1 to n)	The number of model steps it takes for the smoke to spread into its Moore neighborhood.

4.5 Algorithm Explanation

During each time interval that the individual remains alive and has not yet managed to escape:

1. Check Health/Mobility for every smoke tile located within the vicinity of the human, decrease the human's health value by 0.1, and reduce their speed value by 1. When the health reaches zero we mark them as dead.
2. Refresh the agent's field of vision by incorporating the currently observable objects.
3. Check the state of panic by calculating shock and nervousness values according to occurrences of smoke, fire, panicking, or dead agents in their vision.
4. Designate the tiles currently within the agent's field of vision and not blocked by smoke or fire as "known."
5. If the agent's knowledge grid contains information about a fire escape, assign the coordinates of the exit as the target destination. In cases where multiple exits are known, select the exit with the most direct path.
6. Check rescue and count the rescued if a human is in view and their mobility is 2 or they are incapacitated we set the agent as the target.
7. Movement when the target location is insight find the shortest unblocked path toward it. when the target is not in sight we perform a depth-first search toward it.

4.6 Testing Setup

The arrangement and configuration of resources, tools, and conditions required for conducting tests or experiments. It involves preparing the environment and ensuring that all necessary components are in place to execute the testing process effectively.

4.6.1 Fixed Parameters

- Building Plan: We use the same building plan as shown in figure 9 for all of our test runs.
- Number of humans: 100
- Probability of fire: 0.5
- Random human locations: True
- Speed: Random value between 1 and 2
- Percentage of rescuing: Varied between 0% and 100% in steps of 10%.
- Frame per second: 3

4.7 Result

The purpose of the experiment was to examine how people's behavior in emergency situations varies and to compare the effectiveness of rescue efforts with and without collaboration.

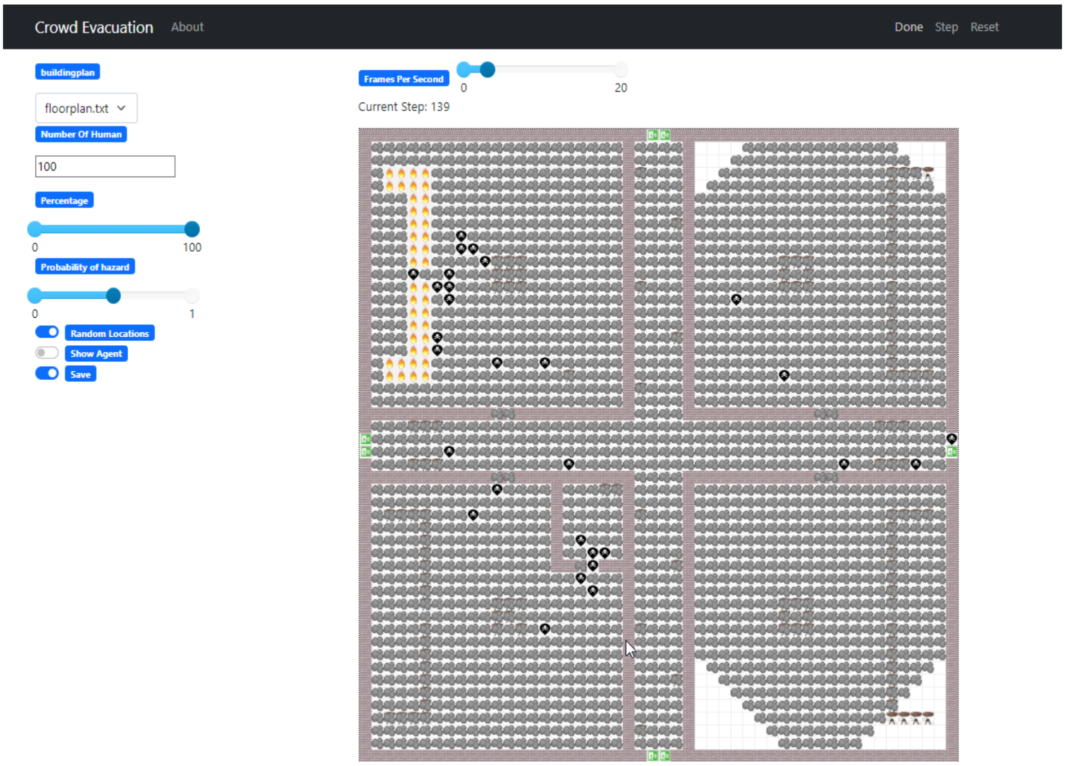

Figure 10: Simulated Result of the building after a fire outbreak.

The results indicated significant differences between 0% and 100% rescuing percentages.

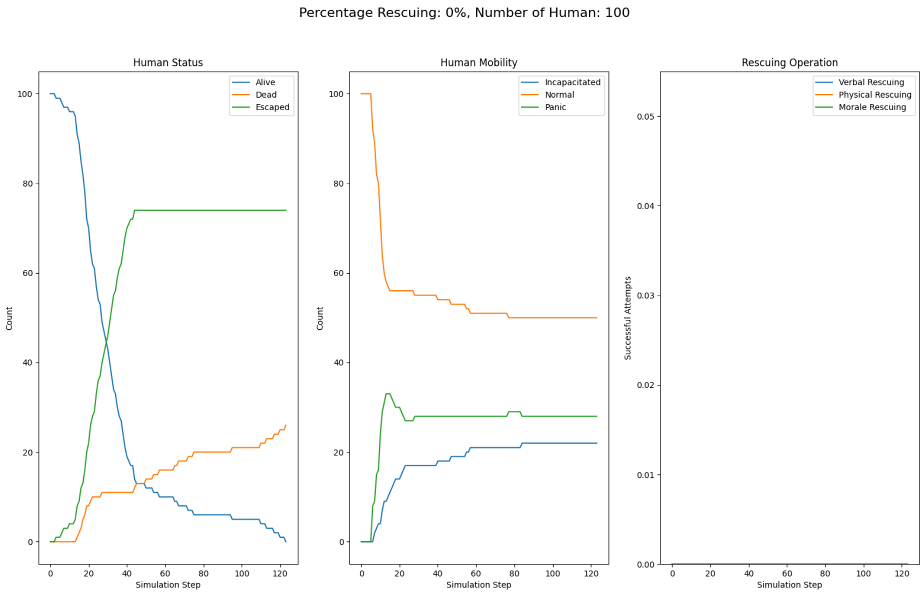

Figure 11: Graph showing the result with zero(0%) percent.

Based on the observation of Figure 11, it is evident that in scenarios without collaboration, and when compared to Figure 12 where verbal collaboration is the most prevalent type, a surprising finding emerges. Despite the similar number of panicking agents depicted in all three figures, the rate at which panicked agents are pacified differs. Another noteworthy distinction is that in the 100% simulation, the evacuation or rescue process was accomplished in fewer steps.

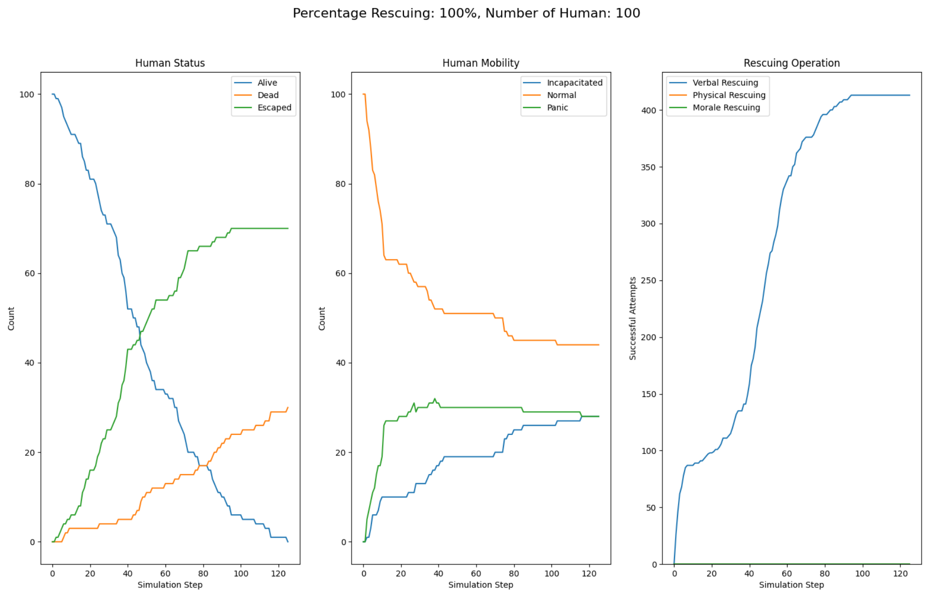

Figure 12: Graph showing the result with 100%.

5 Conclusion and future work

In this project, we have presented a simulation model to test behavior and compare rescuing with or without help in crowd evacuation.valuable insights into optimizing rescue efforts during emergency situations. By considering the complex dynamics of human movement, decision-making, and response to stress, this simulation enables researchers and emergency responders to better understand and anticipate the behavior of individuals in crowded environments. However, it is important to note that the simulation's accuracy and reliability heavily depend on the quality of the underlying behavioral models and input data. Validating the simulation against real-world data and conducting thorough sensitivity analyses are crucial steps in ensuring its applicability to real-life scenarios.

Future work includes expansion of the model's features and implementation of physical and morale rescuing in case of panic and incapacitation in the simulation environment.

References

Cernuzzi, L. Zambonelli (2004), *Experiencing AUML in the Gaia Methodology. In: Proceedings of the Sixth International Conference on Enterprise Information Systems*. Kluwer Academic Publishers.

Fangqin, TANG (2008), "Agent-based evacuation model incorporating fire scene and building geometry." URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6074190`.

JIN, TADAHISA (1997), "Studies on human behavior and tenability in fire smoke." URL `https://publications.iafss.org/publications/fss/5/3/view/fss_5-3.pdf`.

Pablo Cristian Tissera, Alicia Castro (2014), "Simulating behaviours to face up an emergency evacuation." URL `https://arxiv.org/ftp/arxiv/papers/1401/1401.5209.pdf`.