

Formal Methods for Concurrent and Real-Time Systems

**FORMAL ANALYSIS OF SEARCH-AND-RESCUE SCENARIOS**

Homework Project

Presented to: Prof. Pierluigi San Pietro & Dr. Livia Lestingi

By: Manuela Marenghi, Simone Lucca, Louis Dutheil 11028762, Matteo Croce

Spring 2024

**Abstract**:

This report presents a UPPAAL model for a scenario of search and rescue. The global idea is to represent a scenario of search and rescue with civilians that need to be saved from a dangerous environment with fire, first responder trying to save the civilians and drones dispatching instructions to the civilians.

Contents

[1. Design 3](#_Toc172041130)

[1.1 Area layout 3](#_Toc172041131)

[1.2 Stochastic features 3](#_Toc172041132)

[1.3 Design assumptions 4](#_Toc172041133)

[2. Model description 4](#_Toc172041134)

[2.1 High Level Description 4](#_Toc172041135)

[Civilian 4](#_Toc172041136)

[Drone 4](#_Toc172041137)

[First Responder 4](#_Toc172041138)

[2.2 Component description 4](#_Toc172041140)

[2.2.1 Civilian 4](#_Toc172041141)

[2.2.2 Drone 5](#_Toc172041142)

[2.2.3 First Responder 5](#_Toc172041143)

[3. System configuration 6](#_Toc172041144)

[3.1 System parameters 6](#_Toc172041145)

[3.2 Channels 7](#_Toc172041146)

[3.3 System set up 7](#_Toc172041154)

[4. Verification results 7](#_Toc172041155)

# 1. Design

## 1.1 Area layout

The grid is designed as a rectangular layout with the specific constraint that the number of drones must not exceed the number of columns. In the model, it is crucial that no two drones occupy the same cell simultaneously. Therefore, if the number of columns exceeds the number of drones, it ensures that each drone has a distinct group of columns to operate in, thus preventing any overlap.

**Drone positioning** is determined by their IDs, with the drone having the lowest ID positioned on the far left of the grid, and subsequent drones placed in progressively rightward columns. If a drone is not correctly positioned, it will be reset to the top-left cell of its designated column.

**Exits** are situated at the edges of the grid and serve as escape points for civilians. The model reflects the civilians' instinct to reach safety by moving towards the nearest grid edge, where exits are most likely to be found. Additionally, civilians are modeled to avoid fire, ensuring they do not move towards cells adjacent to a fire.

Immagine che contiene quadrato, Rettangolo, linea, Simmetria

Descrizione generata automaticamente

Immagine che contiene modello, Simmetria, quadrato, Rettangolo

Descrizione generata automaticamenteFigure 1 Figure 2

Figure 1 illustrates the movement of two drones within a 6x6 grid. The drones follow their designated paths, ensuring no two drones occupy the same cell simultaneously.

Figure 2 depicts the movement gradient of civilians in the same grid, highlighting their instinctual behavior to move towards the nearest edge to reach safety. The exits, where civilians can escape to safety, are represented in green.

## 1.2 Stochastic features

The following stochastic features are modeled in Homework\_stocv.xml, where an uncontrollable edge with a probabilistic weight is added:

* The drones’ sensor failure rates, represented by the variable Pf (probability of failure), which is a characteristic specific to each drone.
* The humans’ instruction acknowledgment rates, represented by the variable Pr (probability of accepting to help a person in need), which is a characteristic unique to each instantiated Civilian. This probability applies each time a Civilian receives information from a drone, whether they are instructed directly to help or instructed to call a First Responder.

## 1.3 Design assumptions

1. No two entities can be found on the same cell of the grid.
2. No entities can be found on cells with fire or exits since fire will immediately result in their demise, and any person located on an exit will have already left and be considered saved a priori.
3. Distances on the grid are always measured using the Manhattan distance. This distance metric calculates the total number of horizontal and vertical steps required to move between two points, providing a straightforward and practical measure for movement within the grid.
4. Survivors will feel morally obligated to assist people in need and will always follow the instructions provided by the drone.

# 2. Model description

## 2.1 High Level Description

Our model is composed by three entities:

* Civilian
* Drone
* First Responder

### Civilian

The civilians must be saved, but they are not passive, they save themselves by leaving using an exit, or they can even save / help other civilians when instructed by a drone.

### Drone

The drone’s mission is to patrol on the grid and when detecting a civilian in need of assistance it must communicate its presence to another civilian around so that this one can help.

### First Responder

## First Responders navigate the grid with the primary objective of assisting individuals in need. They will provide help if they are near a person requiring assistance or if they are directed by a civilian who has been instructed by a drone to seek their aid.

## 2.2 Component description

### 2.2.1 Civilian

After initiation, the current grid positions of the civilians are evaluated for safety. If a civilian is in a safe position (with no fire within one unit of distance), they are considered a SURVIVOR. If not, the civilian is classified as InNeed and will require assistance. If they do not receive assistance after a defined lapse of time they die and are removed from the grid. If a civilian is located near an exit, they are considered saved and are removed from the grid.

**In Need:**

* Once a civilian enters the "InNeed" state, they will remain in that condition. If they are not assisted within Tv time units, they will perish, emitting a signal on the "dead" channel to indicate their demise.

**Survivor:**

* A survivor prioritizes saving their own life by moving towards the nearest edge of the grid, anticipating that an exit will be located nearby. But when instructed by a drone, the survivor will be obliged to execute the drone’s instructions that can be of helping directly a person in need (ZeroResponder state) or calling a first responder to help the person in need (CallingFirstResponder state).

**ZeroResponder:**

* The survivor will stay in this state for as time as it is needed for it to reach the person in need, once reached it will need *Tzr* time units to save the person in need. If during this process, the person in need dies, then the zero responder will turn back to being a survivor, otherwise both the zero responder and the person in need will be saved

**CallingFirstResponder:**

* The survivor will call the first responder located in the cell indicated by the drone, using a shared variable. This process involves waiting for a duration equivalent to the distance between the survivor and the first responder. Once this time elapses, the survivor will notify the first responder about the person in need of assistance. The first responder will then proceed with the helping procedure. If the assistance is successful, both the survivor and the person in need will be considered saved. If not, the survivor will revert to the survivor state.

### 2.2.2 Drone

After being initialized the drone moves until a PNA (person needing assistance) and a survivor is detected within Dv distance units from itself.

Immagine che contiene quadrato

Descrizione generata automaticamente

Figure 3

In the scenario depicted in the image, a drone has a detection range (Dv) of 3 cells. Within this range, the drone can detect and contact a nearby civilian (zero responder) to assist a person in need who is near a fire (indicated by a red cell). The drone must decide whether to instruct the civilian to call a first responder for assistance or to directly help the person in need themselves.

**Contact first responder**

This decision is taken if a first responder is located within the field view of the drone.

Immagine che contiene quadrato, schermata

Descrizione generata automaticamente

Figure 4

**Help the person in need**

Decision taken if there are no zero responders nearby.

### 2.2.3 First Responder

Following its initialization the First Responder has three courses of action:

1. If a civilian in need of assistance is detected nearby, the first responder will rescue the civilian
2. If called by a survivor, the first responder will travel to rescue the civilian in need of assistance.
3. If none of these scenarios happen, the first responder simply moves toward the center of the grid.

**Civilian in need of assistance detected:**

The first responder is registered as busy and the coordinates of the civilian they are helping is registered using saveIdRescuing(). An intermediate check to see if the civilian in need is still alive is realized, if dead the first responder is freed using setFRFree()and goes back the initial state. Same actions are taken if the time for rescue is completed.

# 3. System configuration

## 3.1 System parameters

|  |  |
| --- | --- |
| Parameter | Description |
| GRID\_W | Grid width |
| GRID\_H | Grid height |
| N\_CIVILIANS | Total number of civilians |
| N\_DRONES | Total number of drones |
| N\_FR | Total number of first responders |
| Tv | Time for a civilian to die close to the fire |
| global\_time | Absolute time of the system |
| grid[GRID\_W][GRID\_H] | Matrix representing the scenario |
| pos[N\_CIVILIANS+N\_FR] | Array of coordinates of all Civilians and First Responders |
| posDrone[N\_DRONES] | Array of coordinates of Drones |
| helping[N\_CIVILIANS] | Array of coordinates of persons assisted by Zero Responder of id equals to the position in the array |
| calling[N\_CIVILIANS] | Array of coordinates of First Responders contacted by Zero Responder of id equals to the position in the array |
| FRHelping[N\_FR] | Array of coordinates of persons assisted by First Responder of id equals to the position in the array plus N\_CIVILIANS |
| n\_safe | Number of Civilians saved |
| Tscs | Time to evaluate properties |
| N\_perc | Percentage of Civilians safe to evaluate in the verifier part |

## 3.2 Channels

## To synchronize entities, we define the following communication channels:

## assist[N\_CIVILIANS]: Used by Drones to request Civilians to assist a person in Need.

## callFirstResponder[N\_CIVILIANS]: Used by Drones to request Civilians to call a First Responder for help.

## assistDone[N\_CIVILIANS + N\_FR]: Used by Civilians to notify when they have been saved by a Zero Responder or First Responder.

## starting: broadcast channel used by the Initializer to start the grid.

## dead[N\_CIVILIANS]: broadcast channel used by each victim to notify others in case of death.

## wrong[N\_CIVILIANS]: Used by First Responders to indicate that they have already rescued a person when they receive a request from a Zero Responder for the same person.

## 3.3 System set up

When creating a Scenario is fundamental to insert the grid configuration in the *initializeGrid()* function in the *‘Declarations’* file, following these rules :

1. Each entity is represented by an enumerated variable in the *Declarations’ file.* To set fire and exits cells it’s sufficient to indicate as follows:

*e.g. grid[y][x] = FIRE*

1. In order to place entities, there is a specific function *placeEntity id, Y, X, entityType)* as follows:

*e.g. placeEntity(0, 0, 1, CIVILIAN)*

1. At the end of the function the following initialization must occur *initCallingHelping(),* *initFRHelping()* and *n\_safe=0*.
2. Last configuration must be in the *‘System Declarations’* file where it is possible to indicate for Civilian its *Tzr,* First Responder its *Tfr* and for drones *Nv.* In the stochastic version it is also possible to indicate *Pr* (probability of giving assistance) and *Pf* (probability of failure) respectively for Civilians and Drones:

*e.g. C0:= Civilian(0, 3)*

# 4. Verification results