FORMAL ANALYSIS OF SEARCH-AND-RESCUE SCENARIOS

Formal Methods for Concurrent and Real-Time systems

Authors:
Giorgio Algisi 10768558
Simone Bevilacqua 10720775
Giovanni Visi 11011386

Instructors:
Prof. Pierluigi San Pietro
Dr. Livia Lestingi

July 22, 2024

POLITECNICO
MILANO 1863

Abstract

Description of a UPPAAL model for an automated rescue system in case of fire. The rescue is managed by a set of drones that interact with the people in the endangered area to instruct them to save others or to call specialized people to rescue them.

1 Model description

The model is composed of 4 components: $FirstResponder$, $Drone$, $Civilian$, $Semaphore$.

FirstResponder
First responders are well-trained agents that intervene in the area to save people in danger. They move in the area until a drone contacts them to give instructions about a civilian in danger.

Civilian
A civilian is a person who finds himself in the endangered area and who tries to safely escape from it. Civilians keep moving until they reach the exit or when they get close to the fire: at that point, they're in danger and they need to be saved within a fixed amount of time otherwise they die. They can also be instructed by a nearby drone to rescue other people.

Drone
The job of the drones is to monitor the area looking for people in danger. During this activity, each drone can decide to either instruct a civilian to save another one in danger or to make a civilian go and tell a first responder to save a person in danger.

Semaphore
The role of the semaphore is to coordinate the movements of the agents involved in the model. Since civilians and first responders can't occupy the same cell (unit of area) at the same time, their movements need to be managed by a centralized unit that is, indeed, the semaphore.

2 Components description

We designed three models with different characteristics from a basic one ($model1$) to an intermediate one ($model2$) to the final and more complete

model ($model3$). Based on the model, the four components have different behaviors and interactions.

First responder

First responders represent specialized people who rescue civilians in danger in the area. In the first two models, first responders keep moving, randomly choosing the next position, while in $model3$ they detect the fire position and start moving around it since it's the place where it's most likely to find new victims. This process can only be interrupted by a drone that sends a message through the *busy* channel to the first responder. $Model2$ and $model3$ also introduce the concept of queue for each first responder. This is done to let drones assign a rescuing task to a first responder who's already busy so that the rescue will be performed right after the current one.

Civilian

The $Civilian$ template represents the people in the endangered area. Each civilian keeps moving either randomly choosing the new position ($model1$ and $model2$) or by going towards the closest exit ($model3$). When they walk close to the fire they are in a danger status and they stop moving. When in danger, if nobody comes to help before a fixed amount of time, the civilian dies. If, instead, a civilian is close both to a person in danger and to a drone, he could be instructed by the latter to rescue the civilian in danger: in this case, they act as zero responders. At this point, after contacting a first responder or saving the person in danger the civilian is safe. The $Moving$ state is the central node of the automaton where many things can happen. If not safe, not in danger and not required to rescue anyone, a civilian can move. If in danger the automaton moves to the $Danger$ state, if safe it moves to the $Safe$ state, otherwise, it can move to the $Rescue$ state where it waits for the rescue time before being safe.

Drone

This template contains the key logic of the whole model. Drones move around the area to monitor the situation: in $model1$ and $model2$ this is done by following a predefined path based on the room characteristics, while in $model3$ drones follow a dynamically calculated path that goes around the fire keeping a distance from it equal to their visibility. In this way, we can increase the effectiveness of the monitoring activity since drones will never find anything interesting (civilians or first responders) inside the fire. After each movement, drones scan their field of vision and look for a person in danger. In case they find one, they check whether another civilian and/or a first responder are nearby: if they find both of them in $model1$ they non-deterministically de-

cide what to do, while in *model*2 and *model*3 they select the most convenient choice based on their distances. Otherwise, they just instruct the civilian to act as a zero responder. If they decide to call a first responder, this interaction is performed through the *busy* channel and then waiting for a reply message on the *contact* channel. If the drone chooses instead to instruct the civilian, it uses the *help* channel to send a message.

Semaphore
This template is a finite state machine that synchronizes the moves of the people in the area. The main goal is to make each movement atomic so that we can't get to a situation with two civilians in the same position. To achieve this, when a person wants to move to a new cell a message is sent to the $Semaphore$ through a dedicated channel $tryingToMove$. The Semaphore checks whether the new proposed position (globally stored as $proposedPos$) is empty and, in that case, it allows the movement by replying with a message through the *move* channel. When the move is not allowed the proposed new position is reset to the current one. This template is also the one that takes care of the initialization of the grid and other global variables and then lets all the other templates start by sending a message on the *sys_init* channel.

3 Channels

The channels used for synchronization between components are the following:

- *sys_init*: one broadcast channel used by the semaphore to communicate when the system initialization process has ended.
- $saved[numMan]$[1]: one channel between the drone and each civilian to communicate when the person in danger is finally saved.
- $tryingToMove[numMan+numResp]$[2]: one channel between the semaphore and each civilian to have the new position approved.
- $move[numMan + numResp]$: one channel between the semaphore and each civilian to approve new positions.
- $busy[numResp]$: one channel between drones and each first responder to communicate new civilians to save.

[1] $numMan$ is the number of humans
[2] $numResp$ is the number of first responders

- $help[numMan]$: one channel between drones and each civilian to communicate new civilians to save.
- $contact[numMan]$: one channel between first responders and each civilian to communicate with the zero responders.
- $serving[numMan]$: one channel between first responders and each civilian to communicate when the first responder is ready to rescue the person in danger associated with the zero responder.
- $jobRefused[numResp]$: one channel between drones and each first responder to communicate whether they can accept or not a new rescue.
- $accepted[numMan]$: one channel between first responders and civilians to communicate whether the civilian has accepted or not the assigned task (for SMC).

4 Design choices

1. The random choice of the next position where to move is modeled at the automaton level rather than at the software level. This is done by letting the automaton non-deterministically choose between the 3 possible moves on the x-axis and the 3 on the y-axis.
2. Also, the random choice between instructing the civilian to rescue the victim (act as zero responder) and instructing the civilian to call a first responder to save the person in danger is modeled at the automaton level rather than at the software level. The drone has, in fact, a non-deterministic choice to make from the $TakeDecision$ state.
3. We added a queue to the first responders since it might happen that if the closest first responder is busy the drone won't have again the chance to assign him the rescue for a civilian in danger since it keeps moving. In this way, we can assign to faster and closer first responders many tasks (rescues) speeding up the whole rescue process.
4. The area is modeled as a grid containing the IDs of the civilians or some default values to represent exits and fires. This choice comes from the need to quickly scan the adjacent cells when deciding the state of a civilian (safe or in danger) and for the semaphore to be able to authorize the movements checking the presence of other civilians.

5. We didn't make the drones make any decision based on parameters that are not known to them in real life like the Nv parameter of the civilians, even though this could potentially improve the models' efficiency.

6. Drones and first responders dynamically choose their path around the fire to have an easily configurable system. Drones choose their path based on their visibility Nv in order to maximize their monitoring activity wasting as few cells as possible. At the same time, first responders keep a fixed distance of 2 from the fire to avoid the civilians in danger who are stuck in the neighboring cells (distance of 1 from the fire). The search for this path is done by going through all the fire cells in the area and selecting all the positions that have the required distance from them. After that, the centroid is calculated so that all the fire boundary cells are sorted clockwise based on the angle between them and the centroid. This path is computed just once at the start of the execution so that the movement phase becomes straightforward since they just need to iterate through the path.

7. In $model2$ and $model3$, where drones make decisions more smartly, we decided to make them consider the distances between first responders and civilians who can act as zero responders. Considering that in a realistic scenario, a rescuing activity done by first responders is way more effective than the one performed by common civilians, we added a weight when comparing the distances in favor of the first responders.

8. All the models have been designed to be fully parametric. Any aspect (from the grid's and fire's size to the agents' features) can be modified depending on the requirements. This type of design allows us to work with scalable and configurable models.

5 Experimental settings

The system can work with rectangular-sized areas. Their dimensions are defined by *height* and *width* parameters. The other parametric characteristics of the models are:

- tZR: time taken by the zero responder to rescue a victim.
- tFR: time taken by the first responder to rescue a victim.
- Tv: time before death for a civilian in danger.
- initial position of civilians and first responders.

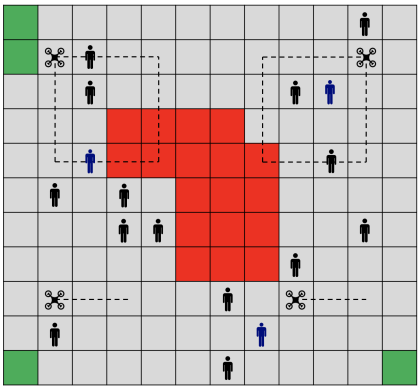

Figure 1: Grid

- initial position of the drones. In *model*1 also the predefined path to follow.
- exits position.
- fires position.

The configuration we chose for the results is represented in Figure 1 and uses a 12 x 11 grid with 14 civilians (stick men in black) and 3 first responders (stick men in blue) all randomly placed inside the area. The drones involved are 4 and, in *model*1 their visibility is chosen according to their predefined path to cover the whole area.

6 Statistical Model Checking

We implemented the required SMC properties making civilians follow the drones' instructions only with probability P_{listen} and drones fail to detect civilians with probability P_{fail}, where $P_{listen}, P_{fail} \in [0, 1]$. To implement the SMC version of the models we had to adapt the previous ones modifying some synchronization mechanisms between the components.

The part related to sensors' failure rates impacts the *Drone* automaton by adding an intermediate node which allows it to express the probability of scanning the map and making a decision rather than fail to do so and keep moving as shown in Figure 2.

The part related to humans' instructions acknowledgment rates is implemented by adding the new channel $accepted[idZR]$ to make the civilian communicate to the first responder whether he will contact him or not placing the decision in the global array $decision[]$. The final message exchange is the following:

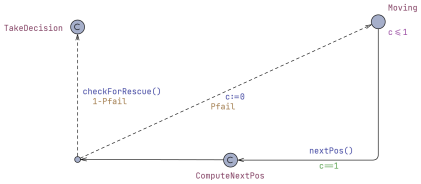

Figure 2: Drone with SMC

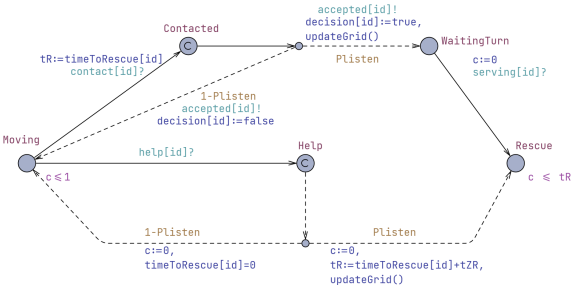

Figure 3: Civilian with SMC

- *Drone* sends a message on the $busy[idFR]$ channel to communicate to a civilian and to a first responder that they have been selected to save a human in danger.
- $FirstResponder$ replies with a message on the $contact[idZR]$ channel to communicate to the drone and the civilian that the task has been taken care of.
- *Civilian* replies with a message on the $accepted[idZR]$ channel to communicate to the first responder that the task has been accepted (due to the P_{listen} probability as shown in Figure 3).
- $FirstResponder$ sends a message on the $serving[idZR]$ channel when ready to serve the task.

7 Properties

7.1 Mandatory properties

We verified the two mandatory properties.

Property 1 It is possible for a percentage $N\%$ of all civilians to reach a safe state within time T_{scs}.

$$\exists \Diamond^{\leq T_{scs}} (numSaved \geq N\%)$$

The best values that we obtained for each model are:

- Model 1: 64% of civilians are safe within 12 time units.
- Model 1 [SMC]: 57% of civilians are safe within 10 time units.

- Model 2: 71% of civilians are safe within 14 time units.
- Model 2 [SMC]: 57% of civilians are safe within 13 time units.
- Model 3: 79% of civilians are safe within 7 time units.
- Model 3 [SMC]: 79% of civilians are safe within 7 time units.

As shown, performances increase with the evolution of the models due to more intelligent behaviors of the drones, civilians and first responders.
SMC versions show worse results than the related models since the probability of the drones being ignored by the civilians or failing has been considered.
The verification aimed to evaluate each version of the model individually in order to find the best performances. For this reason, different time bounds have been used.

Property 2 A percentage $N\%$ of all civilians is always guaranteed to reach a safe state within time T_{scs}.

$$\forall \Box^{\leq T_{scs}} (numSaved \geq N\%)$$

The best values that we obtained for all models are that 7% of civilians are safe within 1 time units. These results were obtained considering verification times under 2 hours. From Figure 4 we can see that in *model*3 we could achieve up to 6 civilians saved in any possible trace, but the verification time required to prove this property would be much higher than 2 hours.

7.2 Mandatory properties SMC

Property 1 We verified the percentage related to the values for $N\%$ and T_{scs} found before. We have found the following values:

- Model 1 [SMC]: 0.7548% of the time, 64% of civilians are safe within 12 time units.
- Model 2 [SMC]: 10.7515% of the time, 57% of civilians are safe within 10 time units.
- Model 3 [SMC]: 23.1178% of the time, 71% of civilians are safe within 14 time units.

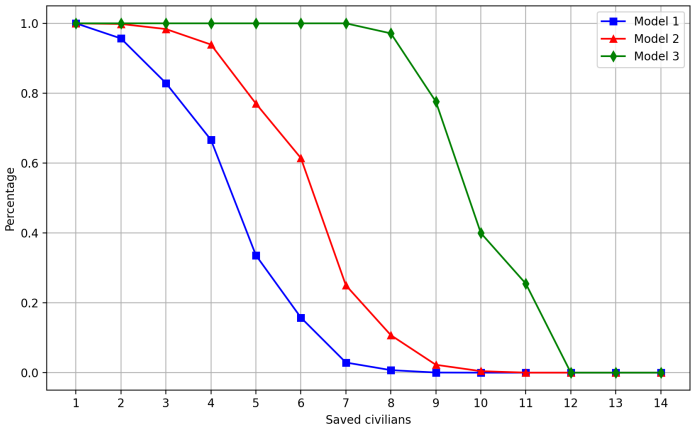

Figure 4: SMC probabilities

Property 2 We verified the percentage related to the values for $N\%$ and T_{scs} found before. We have found the following values:

- Model 1 [SMC]: 99.96% of the time, 7% of civilians are safe within 1 time units.
- Model 2 [SMC]: 99.96% of the time, 7% of civilians are safe within 1 time units.
- Model 3 [SMC]: 99.96% of the time, 7% of civilians are safe within 1 time units.

For the three models, we also verified, given the value T_{scs} found before, the probability that a generic number N of civilians is safe, with $N \in [1, 14]$. As we can see from the obtained results shown in Figure 4 the number of saved civilians grows as we adopt better models, reaching the best results with $model3$. For example, we can see that in $model3$ the probability that 7 civilians are safe is near 100%, while on $model2$ and $model1$ is way lower only reaching 25% and 3% respectively.

7.3 Additional properties

To better analyze the characteristics of our system we also decided to verify additional properties.

Property 3 It is possible to reach a percentage $R\%$ of civilians rescued by others over the total amount of saved civilians.

$$\exists \diamond (\frac{numFRSaved+numZRSaved}{numSaved} \geq R\%)$$

This property is meant to evaluate the effectiveness of the choices taken by drones when they don't make random decisions, but consider the bigger picture of the current situation in order to maximize the number of rescued humans.
The best values that we obtained for each model are:

- Model 1: 42% of civilians are rescued by others.
- Model 1 [SMC]: 45% of civilians are rescued by others.
- Model 2: 42% of civilians are rescued by others.
- Model 2 [SMC]: 50% of civilians are rescued by others.
- Model 3: 40% of civilians are rescued by others.
- Model 3 [SMC]: 50% of civilians are rescued by others.

These percentages represent the correlation between the civilians effectively saved by a rescuer (either a civilian or a first responder) and the total number of saved civilians.
With the evolution of the model, this ratio tends to decrease due to the fact that the number of civilians saved at the denominator increases faster than the numerator. The reason can be found in a greater number of civilians reaching the exits and in the 1:2 correlation for each rescued civilian.
This 1:2 correlation is also the reason for the better performances of the SMC models than the related models (typically, for each not rescued civilians the numerator decreases by one unit while the denominator of two).

Property 4 It is possible to reach a number $N_{rescueFR}$ of rescues by the first responders.

$$\exists \diamond (numFRSaved \geq N_{rescueFR})$$

This property aims to analyze the effectiveness of the joint work done by the drones and the first responders.
We verified that for all the traces in all models at least 2 first responders are

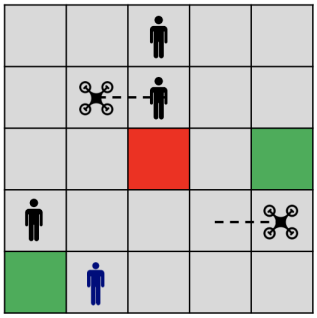

Figure 5: Grid

called to rescue the civilians in danger.

Property 5 There is no deadlock in any trace.

$$\forall\Box(!deadlock)$$

This property wants to ensure the correctness of the system ensuring that there is no deadlock, meaning that the components are correctly interacting with each other. Since this property is related to the way components in the system interact with each other we evaluated this on the three models on a different configuration shown in Figure 5. This was done to reduce the verification times since the results obtained with this small configuration can be generalized to any larger and more complex configuration.

Property 6 Analysis of the average number of saved civilians over all possible traces.

The best values that we obtained for each model are:

- Model 1: an average number of 3.9805 civilians are always saved.
- Model 2: an average number of 5.7242 civilians are always saved.
- Model 3: an average number of 9.4143 civilians are always saved.

As we can see from the results, there is a distinct improvement in the average number of saved civilians between the easiest model ($model1$) and the more complex ones ($model2$ and $model3$). This is due to a more advanced decision-making policy applied by drones that is designed to maximize the number of saved civilians in the endangered area.