Multi-Agent Dynamic Planning Architectures for Crisis Rescue Plans.

Cendrella CHAHINE
ULF Liban/LGP-ENIT
cendrella.chahine@enit.fr

Thierry VIDAL
LGP-ENIT
thierry.vidal@enit.fr

Mohamad EL FALOU
ULF Tripoli
mohamad.elfalou@gmail.com

François PERES
LGP-ENIT
francois.peres@enit.fr

ABSTRACT

We are interested in rescue management in crises such as in terrorist attacks. Today, there are emergency plans that take into account all the stakeholders involved in a crisis depending on the event type, magnitude and place. Unfortunately, they do not anticipate the evolution of the crisis situation such as traffic and hospital overcrowding. In addition, decisions are taken after the information has been passed from the operational level to higher levels. This work focuses on the operational level of the emergency plan. What will happen if the actors at this level, can make certain decisions without escalating the information to higher levels? To answer this question, a multi-agent dynamic planning approach is proposed and it will be tested in two different architectures in order to see how much autonomy can be given to an agent and how they coordinate to save the victims.

Keywords

Multi-agent systems, planning and scheduling, uncertainty, coordination.

INTRODUCTION

Due to the sudden occurrence of crises, the response must be as quick and effective as possible to minimise the human loss by providing help and relief to the injured people. We are concerned with securing the area to prevent further events and, with helping and transporting the injured as fast as possible to the appropriate treatment centre that can treat their cases. As this article is a collaboration work between France and Lebanon, a study on the French generic emergency plan, ORSEC plan was conducted. Moreover, meetings with the Lebanese Red Cross were held to obtain more information on Lebanese emergency plans. We notice that the decisions are not made by the operational level's actors. They transmit information to the actors at higher levels in order to make decisions, then, the latter are performed by the actors at the site (operational level). Based on our studies, the list of actors at this level and their roles are identified. Although France and Lebanon have a different way of managing a crisis, the operational actors have same roles and the same objectives but with different labels and strategies specific to each country. Indeed, both countries face many challenges in their response to a disaster such as the coordination between the heterogeneous actors involved in the rescue plan and the crisis dynamic environment that can interrupt the actors' missions on the site. Gathering information such as the number of casualties and their situations, the availability of ambulances, the capacity of hospitals, etc. is necessary to make decisions. Incorrect or insufficient information affects a decision and can alter the course of the response. Therefore, there is a lot of information and uncertainty (traffic, blocked route to a hospital, ambulance accident…) to manage in a dynamic environment.

In planning and scheduling, we are interested in three problems: resource allocation, the duration and the plans effects. The presented work concerns the definition of a new model combining proactive, reactive and progressive approaches and a multi-agent system to solve the planning problem. Our model will be later tested in both centralised and decentralised architectures to determine which one is the best for our problematic.

BACKGROUND

Multi-agent System (MAS)

MAS is an organized corpus of agents in which a certain number of phenomena comes out as the result of interaction between these aforesaid agents. Briot *et al.*, (2001) define a MAS as a set of interacting agents that can organize in a dynamic and adaptive way. One of the interesting features is the distribution of the complexity over several agents. In this work, the MAS is used to model our scenario in different architectures. Several definitions have described the agent because of the variety of contexts and applications for which the agent is designed. Ferber & Perrot, (1995) defined an agent as a physical or virtual entity that can reproduce itself, has its own resources and skills to achieve its goal and tendencies. It can act, communicate and perceive its environment. Its environment's representation is either partial or null. According to Weiss, (2004), it is a computer system located in an environment where it has the ability to act through autonomous actions to achieve its design goals. EL FALOU, (2010) declares other characteristics of a MAS, we mention:

- Collaboration: consists in making all the agents work on the same project. It refers to techniques that allow agents to divide the tasks, the information, and the resources.
- Observability: is all the information that is accessible to an agent.
- Uncertainty: is when the environment of the agents is considered stochastic and the uncertainties are represented using e.g. probability distributions, intervals of values or fuzzy intervals.

Environmental Properties

An environment is where the agent exists, receives information and carries out its activities. The properties of the environment influence the decision-making process of agents. These properties are (El Falou, 2010; Malas, 2017):

- Deterministic vs stochastic: a deterministic environment is an environment in which the future state is totally determined by its current state and by the action executed. The real environment is stochastic because its state does not only depend on the actions of the agents.
- Static vs dynamic: Unlike dynamic environment, an environment is static when its state changes only when an agent takes an action.
- Discrete vs continuous: an environment is discrete if the number of actions to perform by the agent is finite. Otherwise, it is qualified as continuous.
- Fully observable vs partially observable: an environment is fully observable if the agent perceives the complete state and all the information concerning its environment. Generally, this is not the case, because an agent perceives part of its environment through its limited sensors. In this case, it is partially observable.

Planning and Scheduling

Planning aims to determine the various operations to be carried out and the material and the human resources to be allocated to them. While scheduling aims to determine the different dates and resources corresponding to known activities Baki, (2006). According to Ghallab *et al.*, (2004), planning in Artificial Intelligence is concerned with solving problems in several domains. Problem solving in planning is achieving a goal by taking a series of actions. This solution is called a plan. It is found by a planner. According to Lopez & Roubellat, (2001), solving a scheduling problem consists in placing actions in time, taking into account temporal constraints (deadlines, precedence constraints, etc.) and constraints relating to the use and availability of the required resources by the actions. As resource allocation, time and uncertainty are the main challenges of planning in an emergency response, they are considered in this work.

In planning, the number of actions required to achieve the goal is usually unknown. On the contrary, scheduling starts from a set of actions known in advance, which must be positioned in time with respect to each other. In our project, we are between planning and scheduling because in the rescue plan, we can obtain an overall idea of the size of the plan since the number of injured is equal to the number of evacuations and therefore it is equal to the number of actions but we do not know it precisely because of the uncertainties (exact number of injured, situation of injured, number of available ambulances, new attack, etc.).

Dynamic Approaches

Bidot *et al.*, (2008) presented four different approaches for planning under uncertainty with regard to the balance between the generation and the execution. The first one is creating a plan, while the second one is the step at the moment this plan is executed online. The four approaches are:

- Proactive: has a knowledge base about uncertainty in order to have the power to decide offline. Since the solution is insensitive to perturbations, it is not revised during execution. Knowledge about uncertainty is included to create more robust and reliable schedules.
- Reactive: generates a complete schedule and if during execution the solution differs from the observed situation, then the plan can be changed through online schedule re-generation.
- Progressive: with this approach, we are able to both plan and execute incrementally online. They generate the plan only in the short term. New steps are generated online either on a specific timestamp or whenever a condition expressing that some uncertainties are resolved is satisfied.
- Mixed: combines at least two pure-generation techniques.

Temporal Networks

According to Dechter *et al.*, (1991) Simple Temporal Network (STN) is a set of time points and a single interval constraint between each pair of these time points (V_i, V_j). Each edge is labeled by an interval [a_{ij}, b_{ij}] representing the time interval between a_{ij} and b_{ij}. It represents a constraint $a_{ij} \leq V_j - V_i \leq b_{ij}$ that can be expressed as a pair of inequalities: $V_j - V_i \leq b_{ij}$ and $V_i - V_j \leq -a_{ij}$. In addition, V_0 is a reference point in STN. A solution to a STN is a complete set of assignments to variables such that all temporal constraints are satisfied. Having at least one solution means that a STN is consistent. The simplest solution to solve a STN is to convert the directed constraint graph into a directed edge-weighted graph called a distance graph.

STN do not take uncertainty into account. Thus, by adding uncertainty to STN, a new extension is created called the Simple Temporal Network under Uncertainty (STNU). It is defined by a triple where V and C have the same definitions as in STN, and L is a set of contingent links (Morris *et al.*, 2001; Hunsberger *et al.*, 2013; Cairo *et al.*, 2018). A contingent link represents actions with uncertain durations which means they are uncontrollable but bounded Morris *et a*l., (2001). It has the form where $0 < a_{ij} < b_{ij} < \infty$, V_i is an activation time point assigned by an agent and V_j is a contingent time point controlled by Nature. Moreover, when STNU has a strategy to execute online the executable (non-contingent) time points by satisfying every constraint in the network no matter what is the duration of the contingent link, it is called dynamically controllable.

Although STNU take uncertainty into account, it is not designed for a multi-agent system. Therefore, another extension of STN is MaSTN where time-points are controlled by a set of agents and each variable is assigned an owner Casanova *et al*., (2015). Despite the fact that an agent maintains its private STN, it shares constraints with other agents. We differ between inter-agent constraints where some time-points that belong to different agents are coupled and intra-agent constraints belonging to the agent itself Mogali *et al.*, (2016).

None of the above temporal networks take into account both the multi-agent system and the uncertainty. Casanova *et al*., (2016) proposed a Multi-agent STNU (A, V, E, C) where (V, E, C) is a STNU and A is a set of agents. The reference-point is shared by all agents. As in MaSTN, a variable is owned by a unique agent. A MaSTNU is dynamically controllable when it has an execution strategy that is valid, dynamic and distributed.

FORMAL FRAMEWORK

Definitions

Agent

An agent has a sequence of actions to perform in order to achieve its goal. In a formal way, $A=\{a_1^z,\ldots, a_n^z\}$ is a set of non-instantiated actions for an agent z where |A| = n and $i \in \{1,\ldots,n\}$. It needs to choose among them, those that serve it in order to achieve its goals. The chosen actions are called instantiated actions which means the actions that are inserted in a plan. An executed plan is a plan where all its actions are executed.

Action

It is the task performed by an agent over a period of time. To an action a_i^z is associated the list <$t^-_{a_i^z}$, $t^+_{a_i^z}$, pcond(a_i^z), eff(a_i^z) > such that:

- $t^-_{a_i^z}$ is the date on which the agent starts the execution of a_i^z.
- $t^+_{a_i^z}$ is the date on which the agent finishes the execution of a_i^z.

- pcond(a_i^z) are the preconditions of an action a_i^z. They are the properties of the world required for an action. An action cannot be performed if its preconditions are not satisfied. Thus, preconditions let an agent know when to execute the action.
- eff(a_i^z) are the results of executing an action a_i^z .

Standard Plan

In real world, when someone calls an ambulance center for help, they have standard plans for several types of accidents. They will choose among them the appropriate plan for the situation. Thus, when an event occurs, they will take further tasks. In this context, we differ between a standard plan and a local plan. Based on this idea, we consider that an agent z has a library of standard plans where $Ʃ^z$ = {$Ʃ_1^z$, $Ʃ_2^z$, …, $Ʃ_p^z$}; |Ʃ|=p. Using standard plans instead of actions helps agents to save time in generating their local plans. A standard plan is a non-empty set of sequential actions that have the same order in every local plan and the same constraints. In addition, it has a goal. We consider a standard plan $Ʃ_i^z$ as a septuplets <pcond($Ʃ_i^z$), eff($Ʃ_i^z$), B_k, ≺, g , $t^-_{Ʃ_i^z}$, $t^+_{Ʃ_i^z}$> where:

- pcond($Ʃ_i^z$) are the properties of the world required for an $Ʃ_i^z$.
- eff($Ʃ_i^z$) are the results of executing a $Ʃ_i^z$.
- B_k is a nonempty set of actions; $|B_k|$ = y; $B_k \subset A$. Nonempty set is a set that contains at least one element. A set is called singleton set if and only if it contains only one element.
- ≺ represents precedence constraints; ≺ = ∅ if and only if B_k is a singleton set.
- g: is the goal of $Ʃ_i^z$
- $t^-_{Ʃ_i^z}$: is the time on which the agent starts the execution of $Ʃ_i^z$
- $t^+_{Ʃ_i^z}$: is the time on which the agent finishes the execution of $Ʃ_i^z$

A standard plan composed of one action, has the same preconditions and effects of the action. Otherwise, if a precondition of an action is not established as an effect of a previous action, it is a precondition of the standard plan. If an effect of an action is not canceled by a subsequent action, it is an effect of the standard plan.

Each agent in our architecture has a library of standard plans. Despite the fact that a heterogeneous multi-agent system exists in this work, in a homogeneous set of agents, all agents share the same library but each agent may have a local plan different from the others. For example, all ambulances share the same library, but two ambulances may have two different local plans. This depends on the environment of the ambulance. In addition, each type of agent has a library of standard plans that is different from the others.

Local Plan

It is the combination of several instantiated standard plans. It is executed by an agent in order to reach its goals from its initial task. Local plan $ʠ^z$=(i, G, $D^+_{ʠ^z}$, S=(V_S, C_S, L_S)) where i is the initial task, G is a set of the goal tasks, $D^+_{ʠ^z}$ is the latest time on which the agent must finish the execution of $ʠ^z$ and S is a Simple Temporal Network with uncertainties (STNU). $ʠ^z$ is the local plan ʠ associated to agent z.

Global Plan

It is the combination of the local plans executed by heterogeneous agents to achieve the main goal. For example, if the agents of our scenario are one security agent *s*, one fire brigade *f* and one ambulance *m*, the number of local plans is three ʠ= {$ʠ^s$, $ʠ^f$, $ʠ^m$}. Global plan Γ = (ʠ, ≺, $t^-_Γ$, $t^+_Γ$) where ʠ= {$ʠ^z$, …, $ʠ^{z+n}$}, ≺ represents the precedence constraints. $t^-_Γ$ and $t^-_Γ$ are the time when the global plan starts and finishes its execution, respectively.

Uncertainty

It is a notable occurrence at a given time and it could perturb the schedule of a plan. We could have an uncertain duration of a plan due to uncertain duration of a task and uncertainty on tasks effects. In planning and scheduling, uncertainty about duration, resources and effects are considered. Note that all of the above durations are represented by a temporal network.

Architecture

It is the structure in which our scenario is described and modeled. It contains all our agents, their plans and their environment. It could be centralized or decentralized. A centralized architecture is when the planning decision is made by a unique agent. Whereas a decentralized architecture is when all the agents make decisions. It is not a single agent that controls.

TERRORIST ATTACK SCENARIO

Problem Description

The most lethal form of terrorism is suicide, where the attackers' objective is to kill by committing suicide. In this work, we chose to work on a suicide bombing attack. An attack is characterised by its location, time and damage. For example, an attack in a church on Sunday morning during mass has a much greater impact than an attack in the same place on Wednesday evening. Note that neither volunteers (associations and individual) nor emotional and psychological trauma are taken into account in this work. Therefore, only critical and serious injuries are considered. Furthermore, as we are working on the operational level of an emergency plan, only actors at this level are considered.

On the basis of the ORSEC plan in France, and our meetings with the Lebanese Red Cross in Lebanon, we note that the actors at this level are the same but with different nomenclatures. Therefore, we decide to use terms that are understandable by both countries. Thus, our scenario's agents are: fire brigades, security agents, ambulances and hospitals. As the first three agents are attached to their centres, we also consider each centre as an agent. A centre represents the administration that takes decisions and the call centre that receives and informs the agents concerned. We note that the ambulance is the only vehicle used to transfer victims to hospitals. Moreover, in our work, we are only interested in transferring victims to an appropriate hospital that is capable of treating their injuries and has an available place. Thus, this work does not take into account how this hospital manages the casualty queue during a crisis.

Regardless of the architecture's type, the decisions that should be made to save the greatest number of injured are:

- The number of agents required to respond. This is related to the number of victims, which is unknown at the beginning of the crisis.
- The injured who needs to be transferred first.
- The hospital to which he/she is to be transferred.
- The ambulance that is responsible for transferring him.
- The route to be taken by an ambulance.

Agents Roles

The agents below are used in both architectures, except for the central agent which is only used in the centralized architecture. In addition, certain roles can be added to or removed from certain agents depending on the architecture's type.

- Security agent aims to clear the road for the transfer of victims, seal off the crisis area and secure the zone by killing and arresting the attackers.
- Security centre agent checks its capacity, communicates with the corresponding agents and triggers the necessary security agents.
- Ambulance agent seeks to transfer wounded to hospitals.
- Ambulance centre agent checks its capacity, communicates with the corresponding agents and triggers the necessary ambulance agents.
- Hospital agent aims to communicate its capacity with the corresponding agent.
- Fire brigade agent locates, pick-ups and transfers victims to the Victims' Gathering Point (PRV). In addition, it sorts and provides first aid to the injured.
- Fire department agent checks its capacity, communicates with the corresponding agents and triggers the necessary fire brigade agents.
- Central agent is responsible for generating local plans to enable the relevant actors to respond to a crisis.

Model Description

We want our architectures to be close to reality in order to know how agents can coordinate with each other by executing their private plans to achieve the main goal. Furthermore, we want to know how an agent can make a decision without affecting the local plan of other agents.

In reality, apart from ambulance agents, not all homogeneous agents on the site have the same local plan. For example, the missions are distributed among the fire brigades, which means that they are divided into groups. There is a group to locate victims, another to provide first aid, etc. In our architectures, unlike ambulance agents, the fire brigades are divided into 3 groups:

- F1: to locate and pick-up the victims
- F2: to take them to the PRV
- F3: to aid, sort and prioritise them

In addition, all the homogeneous centre agents involved in the response to a crisis coordinate together. Since this work focus more on the actors acting at the site, the homogeneous centre agents are grouped together, meaning that there is a centre agent for each agent type representing all the centres. For example, for the two security centre agents sc_1 and sc_2, 10 security agents are available in each of them. sc_1 and sc_2 are 30 minutes and 20 minutes away from the site, respectively. This information is used to allocate agents to the site (resource allocation). In planning and execution, the homogeneous agents sc_1 and sc_2 are represented by SC agent.

Moreover, in real life, communication between homogeneous agents and their centre is in broadcast mode, which means that the information is shared between all agents. For example, if a security agent communicates with its centre, all other security agents hear the conversation. In this work, although security agents on site have different missions but their overall objective is to secure the area so that the other agents can intervene. In this context, all the security agents are also grouped into one agent. Since patient management in the hospital is not considered, a single agent represents all hospitals in our architecture.

Crisis

It is characterized by:

- Physical location, longitude and latitude coordinate r (x_r, y_r).
- Number of injured people: d.
- Event time: t.

Injured People

X={x_1, x_2,…, x_n} is a set of injured people where n ∈ ℕ. Each injured is characterised by a medical diagnosis and triage tag that represents his/her situation. Therefore, there are four colours to describe the injured situation:

- Black colour means the victim is dead.
- Red colour identified injured people with critical or serious injuries.
- Yellow colour means that the injured person has a minor injury.
- Green colour represents unharmed victims; they have no physical injuries but may have psychological injuries.

Only injured with red colour tags are considered. They should be transferred to a hospital immediately or before a delay which is maximum six hours, depending on the injury situation Foucher *et al.*, (2018).

Security Centre Agent

SC={sc_1, sc_2,…, sc_i} is a set of security centre agents where |SC| = n, n ∈ $\mathbb{N}^*$ and i={1, …, n}. It has a library of standard plans Л^{SC}= {Л_1^{SC}, Л_2^{SC}, …, Л_j^{SC}}; |Л^{SC}|=p; p ∈ $\mathbb{N}^*$; j={1…,p} and a local plan η^{SC}.

sc_i is characterised by:

- Physical location, longitude and latitude coordinates sc_i (x_{sc_i}, y_{sc_i}).
- Set of available security agents: S^{sc_i}= {$s_1^{sc_i}$, $s_2^{sc_i}$,…, $s_j^{sc_i}$}; |S^{sc_i}|=m; m ∈ ℕ (it can be empty); j={1…,m}.

Security Agent

$s_j^{sc_z}$ is the security agent j that belongs to the security centre agent sc_z. S={S^{sc_1}, S^{sc_2},…, S^{sc_i}} is an agent representing all the available security agents involved in the actual rescue plan where |S|=p; p ∈ ℕ* and i={1, …, p}. S has a library of standard plans $Ꭻ^S$ = {$Ꭻ_1^S$, …, $Ꭻ_j^S$}; |$Ꭻ^S$|=p; p ∈ ℕ*; j={1…,p} and a local plan ɥ^S.

Ambulance Centre Agent

MC={mc_1, mc_2,…, mc_i} is a set of ambulance centre agents where |MC|= n, n ∈ ℕ* and i={1, …, n}. It has a library of standard plans $Ꭻ^{MC}$= {$Ꭻ_1^{MC}$, …, $Ꭻ_j^{MC}$}; |$Ꭻ^{MC}$|=p; p ∈ ℕ*; j={1…,p} and a local plan ɥ^{MC}.

mc_i is characterised by:

- Physical location, longitude and latitude coordinates mc_i (x_{mc_i}, y_{mc_i}).
- Set of available ambulance agents: M^{mc_i} ={$m_1^{mc_i}$, $m_2^{mc_i}$,…, $m_j^{mc_i}$}; |M^{mc_i}|=k; k ∈ ℕ; j={1…,k}.

Ambulance Agent

$m_j^{mc_z}$ is the ambulance agent j that belongs to the ambulance centre agent mc_z. It has a library of standard plans $Ꭻ^{m_j^{mc_z}}$ = {$Ꭻ_1^{m_j^{mc_z}}$, …, $Ꭻ_j^{m_j^{mc_z}}$}; |$Ꭻ^{m_j^{mc_z}}$|=p; p ∈ ℕ*; j={1…,p} and a local plan ɥ$^{m_j^{mc_z}}$. Each ambulance agent has a capacity *cap* which can be 1 or 2, which means an ambulance can transfer either one or two casualties at a time.

Hospital Agent

H = {$h_1, h_2, \dots, h_i$} is a set of hospitals where |H|= n, n ∈ ℕ* and i={1, …, n}. It has a library of standard plans $Ꭻ^H$= {$Ꭻ_1^H$, …, $Ꭻ_j^H$}; |$Ꭻ^H$|=p; p ∈ ℕ*; j={1…,p} and a local plan ɥ^H. A hospital has a maximum injured capacity. This latter corresponds to the hospital's services, medical team, available operating rooms, emergency rooms and beds, etc. h_i is characterised by:

- Physical location, longitude and latitude coordinates h_i (x_{h_i}, y_{h_i}).
- Capacity; *cap* ∈ ℕ.

Fire Department Agent

FC={fc_1,…, fc_i} is a set of fire department agents where |FC|= n, n ∈ ℕ* and i={1, …, n}. It has a library of standard plans $Ꭻ^{FC}$= {$Ꭻ_1^{FC}$, …, $Ꭻ_j^{FC}$}; |$Ꭻ^{FC}$|=p; p ∈ ℕ*; j={1…,p} and a local plan ɥ^{FC}.

fc_i is characterised by:

- Physical location, longitude and latitude coordinates fc_i (x_{fc_i}, y_{fc_i}).
- Set of available fire brigades in real time: F^{fc_i}= {$f_1^{fc_i}$, $f_2^{fc_i}$,…, $f_j^{fc_i}$}; |F^{fc_i}|=k; k ∈ ℕ; j={1…,k}.

Fire Brigade Agent

$f_j^{fc_z}$ is the fire brigade agent j that belongs to the fire department agent fc_z. There are three groups of fire brigades' agents on site. Fire brigade agent belongs to only one group; $f_j^{fc_z}$ ∈ F_i; i={1,2,3}. F_i= {$f_1^{fc_j}$, $f_1^{fc_y}$,…, $f_k^{fc_j}$, $f_p^{fc_y}$}where j, y, k, p ∈ ℕ*, |F_i|=b, k, p={1,…,b} and j≠ y. F_i has a library of standard plans $Ꭻ^{F_i}$= {$Ꭻ_1^{F_i}$, …, $Ꭻ_j^{F_i}$}; |$Ꭻ^{F_i}$|=p; p ∈ ℕ*; j={1…,p} and a local plan ɥ^{F_i}.

Central Agent

c is the decision-maker of the centralized architecture. It has all the libraries of all the agents. Note that in a centralised architecture, the above agents do not have a library of standard plans.

Centralised Architecture

Figure 1 shows the interaction between our model's agents of this architecture.

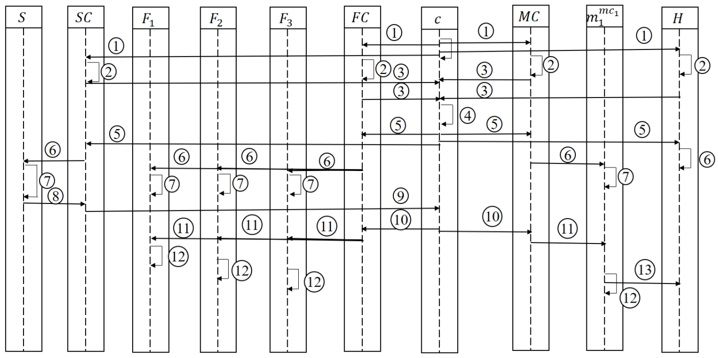

Figure 1. Interaction between agents

The steps below describe Figure 1:

1. When central agent receives an alert, it generates its local plan using the progressive approach, which means that part of its plan is generated offline and the following steps are carried out during execution. It starts to execute its private plan. It sends to each centre agent (including H) its local plan and demands its capacity.
2. They start implementing their local plans. Note that they use a proactive approach as their plans are generated offline.
3. They send their capacities to c.
4. c adds the necessary standard plan to its local plan and begins assigning agents to site and casualties to hospitals.
5. It informs SC, FC, MC and H of its decision and sends the local plans of the agents concerned.
6. SC, FC and MC trigger the necessary agents and send local plans to them. While H prepares to accept casualties.
7. S, F_1, F_2, F_3 and $m_1^{mc_1}$ execute their local plans. As for their centres, they use a proactive approach.
8. No agent can intervene without the area being secured to avoid further unexpected events. When S secures the site and completes the execution of its local plan, it informs its centre.
9. SC transmits the information to c.
10. c informs FC and MC.
11. FC and MC inform their agents.
12. Agents continue their local plans' execution. Note that F_2 cannot take casualties to PRV unless F_1 finds at least one injured person. Moreover, F_3 cannot continue to implement its plan without at least one casualty in the PRV. In addition, $m_1^{mc_1}$ can transfer an injured if it is in PRV.
13. $m_1^{mc_1}$ informs H of its expected time of arrival and the situation of the injured person.

The last two steps are repetitive until there are no more wounded on site. Figure 2 shows the interaction between our model's agents by adding uncertainty. It is assumed that the road is suddenly blocked during the transfer of an injured to a hospital by $m_1^{mc_1}$.

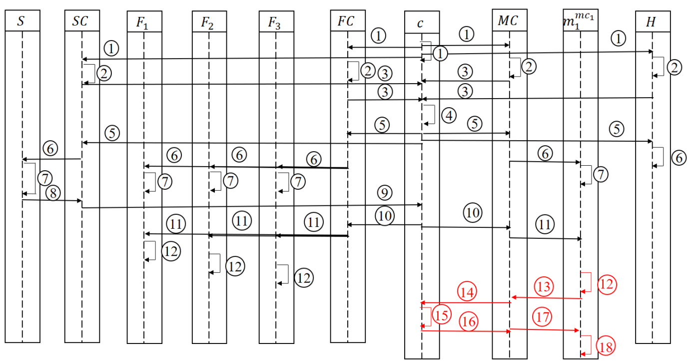

Figure 2. Interaction between agents under uncertainty: Effect

The 12 first steps are the same as above. The ambulance's local plan's effect is different from the expected result. Thus, it stopped the execution of its local plan (12 in red). The following steps describe the additional tasks required to resolve the blocked road event:

13. $m_1^{mc_1}$ informs MC.
14. MC informs c.
15. c takes further tasks. It finds another road for the hospital and adds new standard plans to the ambulance's local plan.
16. c sends them to MC.
17. MC informs $m_1^{mc_1}$
18. $m_1^{mc_1}$ continue its local plan's execution by performing the new standard plan added to its plan.

We have given an example of uncertainty about a plan's effect. Next, we assume that a new casualty is found at time t and $m_1^{mc_1}$ is the only agent available (see Figure 3).

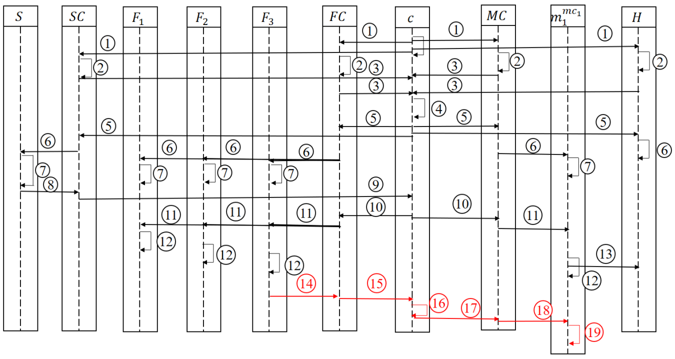

Figure 3. Interaction between agents under uncertainty: resource

The 13 first steps are the same as Figure 1. To solve this problem:

14. F_3 informs FC about the new casualty.
15. FC informs c.
16. c assigns a hospital and an ambulance to the injured; we assume it is $m_1^{mc_1}$. In addition, it adds the necessary standard plan to ambulance's local plan.
17. c informs MC.
18. MC informs $m_1^{mc_1}$
19. $m_1^{mc_1}$ executes its local plan.

For the last example, a new casualty is found at time t and there are two available ambulances $m_1^{mc_1}$ and $m_2^{mc_1}$ (see Figure 4).

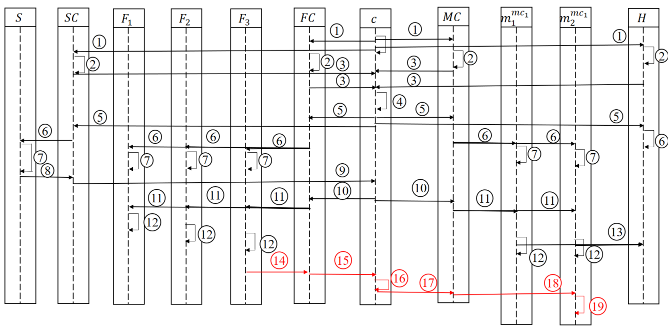

Figure 4. Interaction between agents uncer uncertainty: duration

Figure 4 has the same steps as Figure 3 but with different assignment where $m_2^{mc_1}$ is the appropriate ambulance to transfer the new casualty (step 16). It is assumed that the injured person has to be transferred to the hospital in a time interval that is less than the time it takes $m_1^{mc_1}$ to arrive at the site and then transfer the wounded and greater than the duration needed by $m_2^{mc_1}$.

Decentralised Architecture

Figure 5, Figure 6, Figure 7 and Figure 8 show the interaction between the agents of this architecture in the same four cases as in the previous architecture.

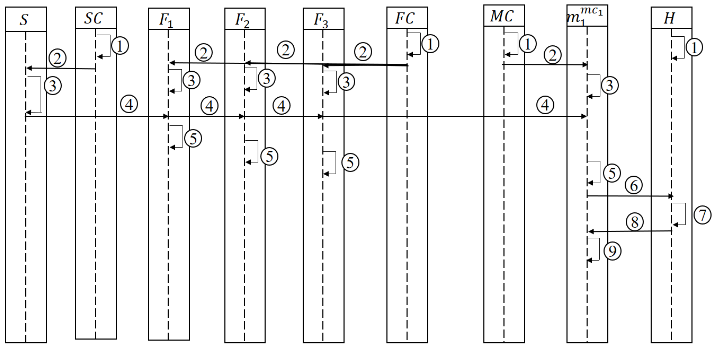

Figure 5. Interaction between agents

The steps below describe Figure 5:

1. When SC, FC, MC and H receive an alert, they generate their local plan using the proactive-reactive-progressive approaches. It means their plans are generated offline and will be executed online. If an event disturbs their plans, they use the reactive or the progressive approach depending on the type of uncertainty to solve the problem. With the exception of H, centre agents assign the necessary agents to go to the site.
2. Each centre informs its agents.
3. When they receive the decision, they generate their private plans and start to execute it.
4. To avoid further unforeseen events, S informs all agents present on site when the area is secured.
5. Idem 12
6. $m_1^{mc_1}$ informs H of the injured situation.
7. H assigns a hospital to the injured person.
8. H informs $m_1^{mc_1}$.
9. $m_1^{mc_1}$ executes the rest of its local plan.

Note that steps 6 to 9 are repetitive for every casualty. Thus, progressive approach is the best to use in this case. Now, if the road is suddenly blocked, $m_1^{mc_1}$ uses the reactive approach and searches for another road to the hospital (Figure 6: step 9 in red).

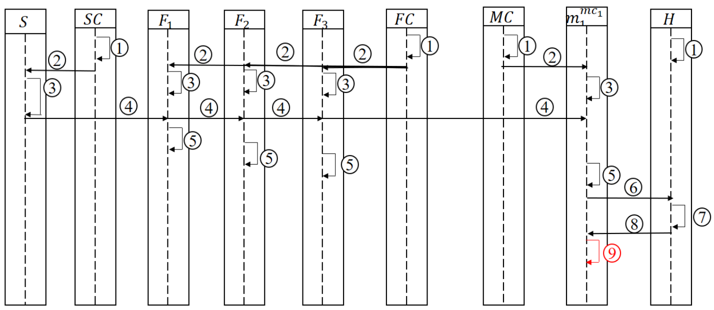

Figure 6. Interaction between agents under uncertainty: effect

If new casualty is found at time t and $m_1^{mc_1}$ is the only available agent:

10. F3 informs all the ambulances. In this example, there is only one ambulance.
11. $m_1^{mc_1}$ adds a new standard plan that allow it to return to site and executes it.
12. It also adds a new standard plan to transfer the injured to a hospital. It executes it.
13. Idem 6
14. Idem 7
15. Idem 8
16. Idem 9

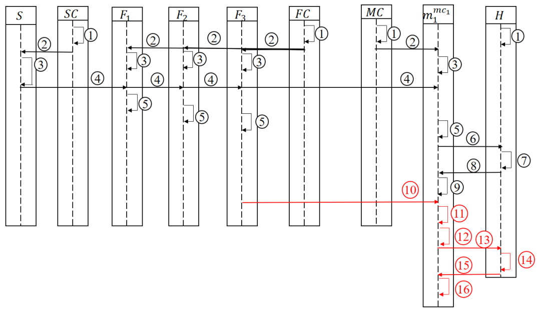

Figure 7. Interaction between agents under uncertainty: resource

If there are two available ambulances for the new casualty:

10. Idem 10
11. The nearest ambulance to the site will transfer the injured to the hospital. It informs the other ambulances. In the following steps, $m_1^{mc_1}$ is replaced by $m_1^{mc_2}$
12. Idem 11
13. Idem 12
14. Idem 13
15. Idem 14
16. Idem 15
17. Idem 16

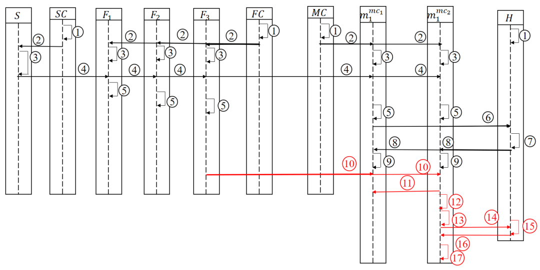

Figure 8. Interaction between agents under uncertainty: duration

Decentralised Architecture Vs Centralised Architecture

Table 1 shows the decision-makers in each architecture and the decisions they make to help casualties. Although we have not simulated our scenario, based on all the figures above, we notice that the communication is higher in centralised architecture than in decentralised architecture when there is no uncertainty. The difference between the two architectures concerning communication to choose the most suitable available ambulance to transfer a casualty is that the communication in decentralised architecture depends on the number of ambulances. Moreover, despite the fact that *c* needs a large data space to save all the libraries of all the agents, it browses them all to find the requested one to build the plan of the requested agent. In addition, it can handle one request at a time. In other words, it collects all the requests in a list, and treats them one at a time using FIFO (First In, First Out) method. Yet, what happens if there is one demand that is more important than the others? For example, we assume that the route is blocked when $m_1^{mc_1}$ and $m_2^{mc_1}$ transfer x_1 and x_2 to h_1 and h_2, respectively (see Figure 9). x_1 must be transferred in a time interval shorter than x_2, but, the request of $m_2^{mc_1}$ is the first one received by *c* ($t_1 < t_2$). In this case, *c* treats the demand of $m_2^{mc_1}$ before $m_1^{mc_1}$ and there is a risk of losing x_1. The time for re-planning and allocation can be very long. The distribution of decisions between agents in the decentralised architecture solves this problem. Furthermore, each agent has its own library which can access it quickly. Yet, the data space depends on the agents' number involved in a decentralized architecture.

Table 1. Decision-maker in both architectures

	Centralised	Decentralised
Agent S	-	
Agent SC	-	Number of agents required to respond
Agent F_1	-	-
Agent F_2	-	-
Agent F_3	-	Injured priority
Agent FC	-	Number of agents required to respond
Agent MC	-	Number of agents required to respond
Agent $m_1^{mc_1}$	-	The route and which injured to transfer
Agent H	-	Appropriate hospital
Agent c	All the decisions	-

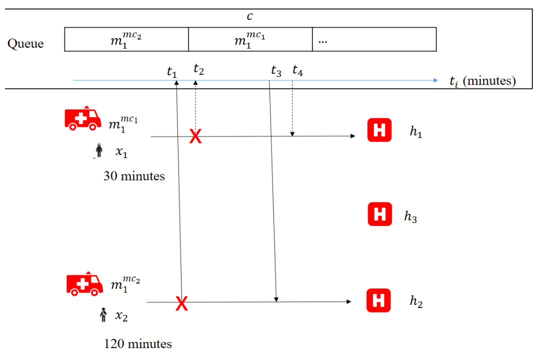

Figure 9. Example of how central agent handles requests

CONCLUSION

The ORSEC plan gives an overview of the actors' missions and the coordination between them to save the victims without taking into account the sudden events that can disrupt their missions. Operational level actors face many uncertainties in achieving their goals. In this context, a multi-agent model for the operational level is proposed in this work to show the coordination of actors at this level and their ability to cope with these uncertainties. It is presented in a theoretical way in a centralized architecture. Some examples of how an agent can behave in this architecture to deal with a sudden event are explained. The agents of the scenario and the distribution of decisions in the two architectures is also discussed.

To validate proposed model, future work should simulate the two architectures to automatically generate plans for agents using a mixed planning approach. Then, they will be tested and compared to indicate which one is the best for our problematic.

REFERENCES

Briot, J., Demazeau, Y., & Gasser, L. (2001). *Principes et architecture des systèmes multi-agents* (IC2 information - commande - communication Informatique et systèmes d'information).

Ferber, J., & Perrot, J. (1995). *Les systèmes multi-agents vers une intelligence collective* (IIA, Informatique, intelligence artificielle).

Weiss, G. (2004). Multiagent systems. *In Morgan kaufman Publishers.*

El Falou, M. (2010). Contributions à la composition dynamique de services fondée sur des techniques de planification et diagnostic multi-agents

Baki, B. (2006). Planification et ordonnancement probabilistes sous contraintes temporelles.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning - theory and practice*.

Lopez, P., & Roubellat, F. (2001). *Ordonnancement de la production* (IC2 Information, Commande, Communication Série Productique). Paris: Hermès science publications.

Bidot, J., Vidal, T., Laborie, P., & Beck, J. (2008). A theoretic and practical framework for scheduling in a stochastic environment. *Journal of Scheduling, 12*(3), 315-344.

Malas, A. (2017). Contributions à la résolution du transport à la demande fondées sur les systèmes multi-agents. (Contributions to on-demand transport resolution based on multi-agent systems).

Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence, 49*(1), 61-95.

Morris, P.H., Muscettola, N., & Vidal, T. (2001). Dynamic control of plans with temporal uncertainty. *IJCAI.*

Hunsberger, L. (2013). Magic Loops in Simple Temporal Networks with Uncertainty – Exploiting Structure to Speed Up Dynamic Controllability Checking. *ICAART*.

Cairo, M., Hunsberger, L., & Rizzi, R. (2018). Faster Dynamic Controllability Checking for Simple Temporal Networks with Uncertainty. *TIME*.

Casanova, G., Pralet, C., & Lesire, C. (2015). Managing Dynamic Multi-Agent Simple Temporal Network. *AAMAS*.

Mogali, J.K., Smith, S.F., & Rubinstein, Z.B. (2016). Distributed Decoupling of Multiagent Simple Temporal Problems. *IJCAI*.

Casanova, G., Pralet, C., Lesire, C., & Vidal, T. (2016). Solving Dynamic Controllability Problem of Multi-Agent Plans with Uncertainty Using Mixed Integer Linear Programming. *ECAI*.

Foucher, S., Le-loch, J., Desbrest, A., Gabilly, L., Lefort, H., & Tazarourte, K. (2018). Catastrophe avec nombreuses victimes en milieu urbain : plan blanc et difficultés de gestion des places hospitalières, 213–223.