

Dynamic Path Planning for Indoor Evacuation Routing: A Universal Planning Framework

Ramzi Djemai¹, Vassil Vassilev², Karim Ouazzane³, and Maitreyee Dey⁴

School of Computing and Digital Media, London Metropolitan University, London N7 8DB, UK

Corresponding author: r.djemai@londonmet.ac.uk

Abstract. In the event of indoor disasters, evacuation processes face a multitude of real-time constraints, including the limited time available for a successful evacuation and the necessity of guiding occupants clearly and safely. These constraints are not limited to office and residential buildings but also apply to airports, hospitals, cruise ships and more. Tragic incidents such as Grenfell Tower's deadly 2017 fire and that of the shopping mall in Russia in 2018 both serve as poignant examples, where sudden and rapidly spreading fires resulted in the loss of +70 and 62 lives respectively. These events highlight the critical need for efficient evacuation plans to ensure quick and safe evacuations. Prior research conducted on this issue, employs various computational disciplines such as heuristic programming, machine learning, and network optimization, with varying degrees of success. This paper introduces a novel reactive planning framework that incorporates real-world knowledge of building topology and planning policies into a knowledge-based planner and an event-driven replanner. This approach aims to enhance the efficiency and effectiveness of evacuation processes by dynamically adapting to changing conditions and providing real-time guidance to occupants.

Keywords: Real-time Path Planning · Semantic Rule Web Language · Ontology Engineering · Heuristic Planning.

1 Introduction

A planning problem is defined by the sequence of steps required to achieve an objective. [1] In environments where conditions are deterministic and fully observable, limited to a single agent, the planning problem can often be represented symbolically and is relatively straightforward. However, complexities arise when the planning process is subjected to uncertainties, either due to the environment being only partially observable or the non-deterministic nature of actions. Under these uncertain conditions, complex frameworks may employ Markov Decision Processes (MDPs) to handle the complexities of planning and achieving multiple goals concurrently. In such scenarios, the planning problem becomes very complex with such solutions being application-specific and/or highly specialised (non-universal). [2] This research considers the environment to be dynamic and

the outcomes of actions are not predetermined, which adds further complexity that's taken into account in our planning process efficiently [3]. [4] In our context, efficiency is achieved by optimally balancing the level of granularity in the problem domain representation, striking the ideal balance between excessive detail and overly vague representations, thus simplifying the planning problem through abstraction and knowledge representation [5]. By identifying and modeling the planning constraints to form a set of rules, which the planner must observe when generating the plan, one can construct a planning algorithm that allows for effective and efficient evacuation. However, to evacuate people indoors, a routing algorithm alone does not suffice. This is partly due to dynamically occurring events that introduce unpredictability and partly due to the lack of indoor communication infrastructure. [6].

2 The Proposed Reactive-Planning Framework

The framework is a universal framework that can compute routes on most kinds of topological structures, including:

- **Open spaces:** concert halls, mining sites, stadiums, airports etc.
- **Closed spaces:** residential, office, industrial and storage buildings, cruise ships, etc.

It integrates a total of four ontological models which are used as input for the planner. These are the topological model, situations, actions and events models.

- **Topological Model:** a model that consists of entities that describe the building and is the static part of the ontology. The situations model consists of all situations an occupant can be in. Situations are indexed to link actions to situations. This model is further expanded on in the next section.
- **Situations Model:** a model consisting of all the situations an occupant can be in during an evacuation. Situations are unary predicates that describe the state of the occupant. See figure 1 below.
- **Actions Model:** This model conceptualises movement through the building and it consists of all actions an occupant can make during a given evacuation. These actions are indexed and classified into navigational actions (prescribing movements in space), corrective actions, and descriptive actions. Actions are modeled as object properties which are binary predicates relating one situation to the next. For example,

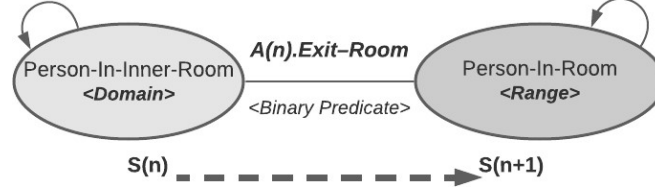


Fig. 1. Framework Process Flow

Consider a scenario where an occupant x is involved in an evacuation. We define two situations and an action as follows:

- **Situation 1:** Occupant x is in an inner room, denoted by the unary predicate $InInnerRoom(x)$.
- **Action 1:** Occupant x takes the action to exit the room, denoted by $ExitRoom(x)$.
- **Situation 2:** As a result of the action, occupant x is now in another room, denoted by $InRoom(x)$.

$$InInnerRoom(x) \wedge ExitRoom(x) \Rightarrow InRoom(x) \quad (1)$$

This equation states that if occupant x is initially in an inner room and takes the action to exit the room, then as a result, occupant x will be in another room.

- **Events model:** This model consists of emergency events which can occur spontaneously during an evacuation. These events are indexed and classified into two categories; Impact Events and Non-Impact Events. The models of both situations and events are part of the dynamic part of the ontology.

To illustrate how various ontological models interact in our reactive-planning framework, Table 1 below outlines the model parameters, classifying them into descriptors, classifiers, quantifiers, and qualifiers. This provides a structured overview of the data each model processes and its contribution to the overall evacuation plan.

Table 1: Parameters of Models in the Conceptual Reactive-Planning Framework

Model	Parameter Type	Description
Topological Model	Descriptors	Define physical attributes of the building space such as doors, offices and obstacles.
	Classifiers	Categorize elements within the topology, e.g., movable/immovable obstacles, physical space, objects etc.

Table 1 Continued from previous page

Model	Parameter Type	Description
	Quantifiers	Measure attributes like the length and capacity of paths.
	Qualifiers	Describe the quality or condition of an element, such as accessibility.
	Temporal	Time-related attributes, e.g., expected durations to traverse between nodes.
Situations Model	Actor	Identifies the individual or occupant within a situation.
	Location	Specifies occupant position within the topology.
	Classification	Categorizes the nature of the situation, e.g., 'Safe' or 'Dangerous'.
	Description	Detailed information about the situation's characteristics.
Events Model	Classifier	Categorizes events by their nature, e.g., 'Impact' or 'Non-Impact'.
	Temporal Param	Duration of an event from start to end of life (EOL).
	Description	Detailed information about the event's characteristics.
Actions Model	Input Param	Start/original situations defining the initial conditions of an action.
	Output Param	Resultant situation after executing an action.
	Condition	Specifies the prerequisites for an action to be viable.
	Effect	Details the consequences or changes brought about by an action.

2.1 Building Topology

In the case of this research, the topology here is that of a building, which can be considered a topological, fixed 3D space within which the planner will compute the evacuation task. This adopted definition pinpoints the similarity between – and applies to– most building structures. This model consists of all the concepts that together constitute a building. Concepts or classes are unary predicates and are representative of the topological information such as <doors>, <corridors>, <floors>, <emergency exits>, <rooms>, etc.

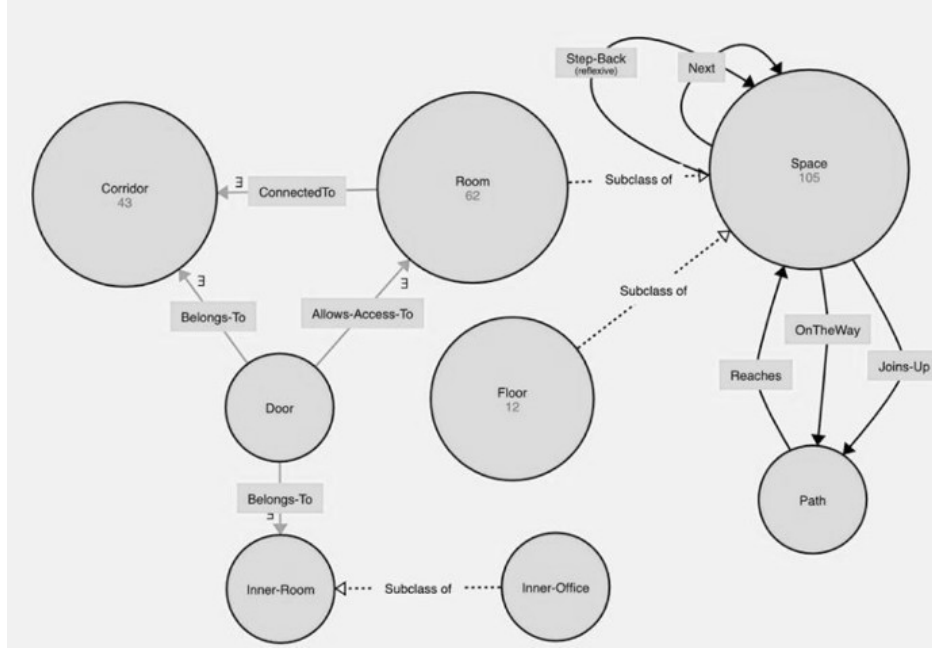


Fig. 2. Topological Domain Representation

2.2 Evacuation Process

The Evacuation Process Model (EPM) uses the ontological models described above, governed by the planning rules and processed by the controller. Each ontological model provides its respective information, and they are governed by the rules, and the controller implements the ontological models. The rules engine processes the rules, the heuristic engine implements the rules, the events processor processes the events, and the interpreter simulates the evacuation, which is then translated into a state space search graph via a dashboard interface [7].

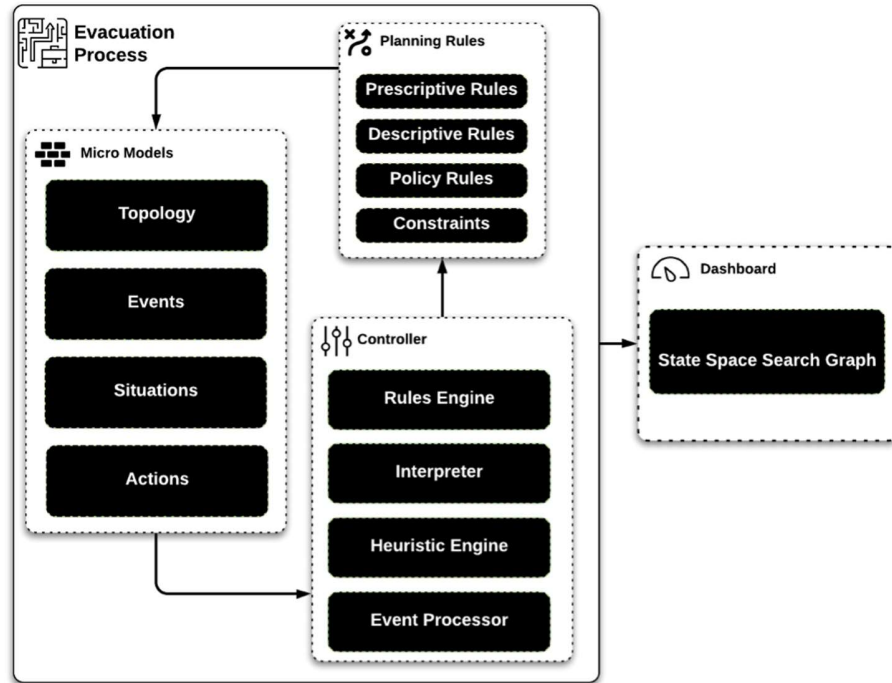


Fig. 3. Topological Domain Representation

Policy rules are the building rules that must be complied with during emergencies and act as limitations or constraints on the evacuation planning process. They are indexed for use during the evacuation. The heuristics model contains two types of rules: prescriptive and descriptive which are loaded into the rules engine [8]. The inference engine generates inferences driven by several factors, including but not limited to the asserted rules. The planner works independently (all occupants use one and the UI), meaning from a user's point of view, the planning is done for one person, from one app for one evacuation [9]. The diagram above (figure 3) illustrates the evacuation process where the dashboard is the front-end sitting on the client side that uses the three components from the server side to process the state space search graph and provides live routing and guidance during the evacuation [10].

This means that the planning is not coordinated; it works independently, but some routes may overlap in some cases. What changes from one occupant to another is the starting situation, the different events they experience during the evacuation and the actions they take. The rules engine is fed data from the input models. This is due to the evacuation is dynamic, and any rule may be triggered depending on unravelling events. As such, the rules engine recommends actions to the occupants. The rules are written in a semantic ontology format (SWRL)

and are queried using the Semantic Web Query Language.[5] The following example is a snippet of simpler rules, with some prescribing actions and others being descriptive.

$$\begin{aligned} \mathbf{Person-In-Room(?pir) \wedge Exit-Room(?por) \rightarrow} \\ \mathbf{Person-Outside-Room(?por)} \end{aligned} \quad (2)$$

$$\begin{aligned} \mathbf{Smoking(?s) \wedge Room(?r) \wedge Occupant(?o) \wedge} \\ \mathbf{Taking-place-in(?s, ?r) \wedge is-in(?o, ?r) \wedge} \\ \mathbf{Person-Outside-Room(?por) \rightarrow Exit-Room(?o, ?r)} \end{aligned} \quad (3)$$

$$\begin{aligned} \mathbf{Person-In-Corridor(?pic) \wedge Exit-Corridor(?pic, ?piv) \rightarrow} \\ \mathbf{Person-in-Vestibule(?piv)} \end{aligned} \quad (4)$$

$$\begin{aligned} \mathbf{Staircase(?s) \wedge Room(?r) \wedge contains(?F, ?C) \wedge} \\ \mathbf{Corridor(?C) \wedge Floor(?F) \wedge Connects(?s, ?F) \wedge} \\ \mathbf{AccessVia(?R, ?C) \rightarrow Indirectly-Connect(?s, ?R)} \end{aligned} \quad (5)$$

The distinction between the concept of Path and Route is vital from an ontology and heuristic estimation context. Although the two are interchangeable in many existing pieces of literature, the way the framework is developed, the two are distinct and, as a result, non-interchangeable [9] and [10]. Path sits at the top-most level of the ontology and is made up of the following characteristics:

- A physical concept related to the location of spaces.
- Fixed and non-dynamic in nature.
- Independent from individuals or occupants.
- A sequence of connected spaces, ensuring continuity from one space to the next.

On the other hand, the concept of Route is defined as:

- Dynamic, subject to changes (re-rerouting) based on situations and triggered events.
- Made up of several paths.
- Generated over existing paths.
- The route is not subject to obstruction by obstacles, but rather, the path may be blocked, in which case the route over this path is not possible.

3 Planning Policies

Planning or evacuation policies are an integral component of our reactive planning framework. These are rules which are indexed and classified according to the following types which govern the evacuation planning process under various conditions:

- **Prescriptive Rules:** These are rules that prescribe actions to occupants during the evacuation. These dictate what action an occupant must take. For example, navigate to floor 9, turn left, exit room, walk down a corridor, etc. These actions can be generated on an atomic basis (a1, a2, a3, a4, etc) or it may be more efficient to recommend a composite action instead of planning “next” each time. This is discussed in more detail in the Planning and Replanning Section.
- **Descriptive Rules:** These are rules that describe the environment [11]. I.e., These rules are based on the logic of the building topology and drive the inference engine. For example, behind x door, there’s y room.

Fig. 4. Framework Process Flow

- **Constraints:** constraint-based rules are restrictive rules that express what cannot be done and come from the building policy. In effect, rules take into account constraints. In other words, rules are formulated with constraints in mind. For example, if the capacity of a path is too small or turns into a bottleneck, a constraint can be formulated as the following:

$$\text{PathCap} = \text{full} \iff x(\text{InstanceOfPerson}) > y \quad (6)$$

Where:

- **PathCap** represents the status of the path capacity.
- **$x(\text{InstanceOfPerson})$** represents the number of people currently on the path.
- **y** represents the maximum capacity of the path.

When such a condition holds, the heuristic function used by the planner stops further guidance to this path, suggesting an alternative route instead. Rules are processed and managed by a rules engine within the planning framework and play a crucial part in the design of our evacuation scenarios.

4 Planning and Re-planning in State Space

Given that the route generation is dependent on the topological model, there are two primary methods for planning an evacuation. To illustrate, consider the goal is to get to the stairs, and an occupant is in a given room. There are two methods of planning for this – the first type of planning is when each next action is planned atomically (a1, a2, a3, etc.), one at a time which may be unnecessary and repetitive. The second and better way is to abstract the goal state in such a way that actions leading to it are grouped. This allows for planning several actions at a time as a composite action. A composite action can be considered a composite function in that each next action is the continuation of the previous. It means the planner can formulate one composite action at a time, which contains a sequence of actions, see Figure 5 below.

Fig. 5. Framework Process Flow

However, in either scenario, events may occur and change the action – in which case, the rules recommend rerouting actions which are actions taken to reroute the occupant’s route to a safe exit. Another fundamental assumption is that whilst the planning is computed for all occupants, each plan is occupant-specific [10].

State space search models use a graph, which is usually an abstract representation of things such as states, processes, locations on a map, etc. These graphs are cast from real-world problems in applications such as Google Maps (finding the least congested route), robotics (Amazon’s order pick robots), and search and rescue (time-critical operations) [14]. State spaces are searched differently according to their application and indicated problem-solving aim. Each search algorithm uses a particular search strategy. Graphs generally consist of nodes and edges, and they can be linear or non-linear, direct or undirected, and edges can have weights. Graph traversal algorithms are used to search through

the graph to find a goal solution to a given problem. They outline an order with which to search through the states or nodes of the graphs, and each set of visited but not explored nodes is referred to as a frontier at each iteration; the visited node is ejected from the frontier set, and its neighbour is added subsequently.

[7] In our case, the aim is to find the shortest, safest path within the graph, which is modelled after a physical building [13]. This is specified by the ontological model, which is a representation of the world's vocabulary, consisting of the physical entities (offices, rooms, etc.) and situations, events, and actions [12].

Each graph consists of an initial state and a goal state. However, in our case, goal states are not necessarily final (fixed). When a goal state is found, it gets designated as the "final" exit from which occupants make it to safety, and the evacuation process is considered finished. However, the goal state may change depending on various factors affecting and governing the route planning, such as:

- **Uncontrolled events:** Exit no longer accessible due to a hazard.
- **Rules:** The route to the exit is blocked or not usable, in which case, the next best goal state is chosen according to the algorithm (meta-rules or strategy)

The search strategy adopted is an informed state space search since the search is guided by the heuristic function, which uses pre-existing knowledge of the domain. This means our strategy can use problem-specific knowledge that goes beyond the problem's definition. The 3rd edition book by Stuart Russel et al., titled "AI, A Modern Approach", suggests that this "best-first" type of search approach can work out solutions more efficiently than an uninformed search strategy. Our planning algorithm (currently being tested) is an adaptation of the LPA* heuristic search algorithm. LPA* is of the same family of heuristic search algorithms and is a modified version of the A* algorithm. However, unlike A*, LPA* can be used in dynamic environments where the conditions can change over time. [4] In an evacuation scenario, the hazards and other factors that affect the optimal evacuation path can change dynamically, and LPA* is used to update the path accordingly. As such, our planning is in effect, knowledge-based and our replanning is event-driven. [8] It uses a simple heuristic function which is needed when there's more than one action to consider, i.e., when there's a choice of routing/replanning, there are several factors that must be considered, and this is governed by the h function which is an integral part of our dynamic planning framework.

5 Conclusion and Further Work

Path planning is central to the idea of navigation and goal-state reachability. However, as far as this research is concerned, path planning alone does not suffice. This is due to the nature of the environment in which the planner is expected to

operate. Hence a need for a topological, constraint-based path planner and where people occupy a limited, controlled space, the dynamics of the evacuation remain the same. universal planning framework considers the different events that may occur during the evacuation process, and the planning itself is continuous until an event occurs. In this case, the replanning takes effect to reroute occupants to safety. In the future, the framework will be validated using various events, which will be fed into a simulator to simulate different evacuation scenarios – from the start state (where an occupant is situated) to the next, circumnavigating different obstacles and rerouting (if an event triggered) until the goal state (exit location) is reached.

References

1. Penberthy, J.S. and Weld, D.S., 1992. UCPOP: A Sound, Complete, Partial Order Planner for ADL. *Kr*, 92, pp.103-114.
2. Liu, J., Rojas-Cessa, R. and Dong, Z., 2016, April. Sensing, calculating, and disseminating evacuating routes during an indoor fire using a sensor and diffusion network. In *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)* (pp. 1-6). IEEE.
3. Bode, N.W. and Codling, E.A., 2013. Human exit route choice in virtual crowd evacuations. *Animal Behaviour*, 86(2), pp.347-358.
4. Claßen, J., Hu, Y. and Lakemeyer, G., 2007, July. A situation-calculus semantics for an expressive fragment of PDDL. In *AAAI* (Vol. 7, pp. 956-961).
5. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. and Dean, M., 2004. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, 21(79), pp.1-31.
6. Kang, W.S., Yun, S., Kwon, H.O., Choi, R.H., Son, C.S. and Lee, D.H., 2015, April. Stable path planning algorithm for the avoidance of dynamic obstacles. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings* (pp. 578-581). IEEE.
7. Grimm, S., 2009. Knowledge representation and ontologies. In *Scientific data mining and knowledge discovery: Principles and foundations* (pp. 111-137). Berlin, Heidelberg: Springer Berlin Heidelberg.
8. Kou, J., Xiong, S., Fang, Z., Zong, X. and Chen, Z., 2013. Multiobjective optimization of evacuation routes in stadium using superposed potential field network based ACO. *Computational intelligence and neuroscience*, 2013, pp.2-12.
9. Aguilar, L., Lalith, M., Ichimura, T. and Hori, M., 2016, August. Automatic evacuation management using a multi agent system and parallel meta-heuristic search. In *International Conference on Principles and Practice of Multi-Agent Systems* (pp. 387-396). Cham: Springer International Publishing.
10. Djemai, R., Vassilev, V., Ouazzane, K. and Dey, M., 2024. Knowledge-based reactive planning and replanning. *IEEE Conference on Artificial Intelligence (IEEE CAI 2024)*.
11. Mirahadi, F. and McCabe, B., 2019. A real-time path-planning model for building evacuations. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 36, pp. 998-1004). IAARC Publications.
12. Ünal, A.E., Gezer, C., Pak, B.K. and Güngör, V.Ç., 2022, September. Generating emergency evacuation route directions based on crowd simulations with reinforcement learning. In *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1-6). IEEE.

13. Gul, F., Mir, I., Abualigah, L., Sumari, P. and Forestiero, A., 2021. A consolidated review of path planning and optimization techniques: Technical perspectives and future directions. *Electronics*, 10(18), p.2250.
14. Ferguson, D., Likhachev, M. and Stentz, A., 2005, June. A guide to heuristic- based path planning. In *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)* (pp. 9-18).