


RESEARCH

Open Access

On the realization of the recognition-primed decision model for artificial agents



Syed Nasir Danial, Jennifer Smith, Brian Veitch and Faisal Khan* 

*Correspondence:

fikhan@mun.ca
Centre for Risk, Integrity
and Safety Engineering
(C-RISE), Faculty
of Engineering and Applied
Science, Memorial University
of Newfoundland, St. John's,
NL, Canada

Abstract

This work proposes a methodology to program an artificial agent that can make decisions based on a naturalistic decision-making approach called recognition-primed decision model (RPDM). The proposed methodology represents the main constructs of RPDM in the language of Belief-Desire-Intention logic. RPDM considers decision-making as a synthesis of three phenomenal abilities of the human mind. The first is one's use of experience to recognize a situation and suggest appropriate responses. The main concern here is on situation awareness because the decision-maker needs to establish that a current situation is the same or similar to one previously experienced, and the same solution is likely to work this time too. To this end, the proposed modeling approach uses a Markov logic network to develop an Experiential-Learning and Decision-Support module. The second component of RPDM deals with the cases when a decision-maker's experience becomes secondary because the situation has not been recognized as typical. In this case, RPDM suggests a diagnostic mechanism that involves feature-matching, and, therefore, an ontology (of the domain of interest) based reasoning approach is proposed here to deal with all such cases. The third component of RPDM is the proposal that human beings use intuition and imagination (mental stimulation) to make sure whether a course of action should work in a given situation or not. Mental simulation is modeled here as a Bayesian network that computes the probability of occurrence of an *effect* when a *cause* is more likely. The agent-based model of RPDM has been validated with real (empirical) data to compare the simulated and empirical results and develop a correspondence in terms of the value of the result, as well as the reasoning.

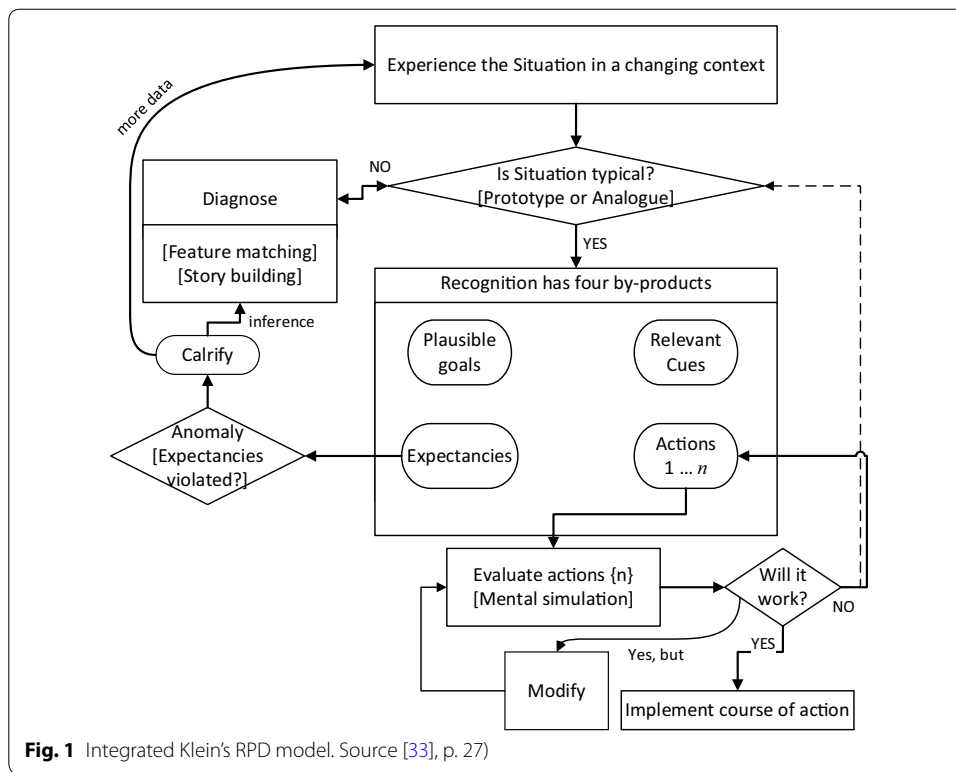
Keywords: Modelling RPDM, Agent modeling, Ontological reasoning, Markov logic network, Feature matching, Mental simulation modeling

Introduction

Naturalistic decision making (NDM) is a relatively new approach to decision-making that relies on situation awareness (SA) [20] rather than having a fixed set of principles from which to choose the best or optimal solution. One of the prominent models of NDM is Klein's [33] *recognition-primed decision model*. RPDM has a descriptive nature, and it requires a thorough understanding of philosophical concepts, such as intuition, perception, and mental simulation. The purpose of this study is to develop a method based on the theory of RPDM that can be implemented in an artificial agent.

There may be many reasons for why an artificial agent based on RPDM should be preferred over those that exploit conventional decision theories. Here are a few concerns important to us. The first being the way how a human mind operates when a decision is to be made. This is even true for cases outside of typical NDM environment that usually is characterized by contextual factors as ill-structured problems, time stress, etc. [67]. For example, in a typical chess play, factors like memory abilities, and the depth of planning (including the number of moves ahead in planning), which are important factors for a decision-making algorithm in terms of comparisons (comparing moves to find the best one in a given state of chess board) and checking alternatives, or finding the best move, have been assessed in experts and novice chess plays [7, 15]. Notice that these factors are important for logical deductions, and so are considered great source of motivations in writing chess programs. The real chess masters, however, have been found to exploit none of these factors, in general, for their mastery in chess playing [35]. de Groot [15] have discovered that novice and expert chess players behave similarly in terms of the overall structure of their thought processes—chess master’s ability to handle the depth of search is almost the same as for the weaker players. The idea that masters can see further ahead than naïve players was dismissed by de Groot’s analysis of verbal protocols, which were obtained when masters and novice players played chess games by thinking a loud in an experiment in 1965. de Groot [15] was unable to pinpoint quantitative differences that could be considered main players for obtaining a mastery in chess except that the masters were found to be able to reconstruct a chess position almost perfectly after viewing it for only 5 s or so [7]. The second reason why RPDM based artificial agents would be better in decision-making lies in the ability to see familiar patterns in the form that could be used to retrieve associated or related information from memory, e.g., the actions performed in a similar situation before, unlike brute force calculations that needs a high-end or a supercomputer to produce desired results by including every bit of information. An example of brute force based calculations used in chess playing was in the IBM Deep Blue that was a supercomputer that defeated the world champion Garry Kasparov in 1996 [48]. The RPDM based agent model has scope in potentially any decision problem, most importantly are those that involve high stakes, and time pressure, such as trading agents, firefighting, and emergency evacuation simulation applications.

RPDM may be considered as a way to develop insight into improving ways to better respond in different operating conditions. However, the model is for experienced people, not for artificial agents. The purpose of this study is to develop a realization of RPDM suitable to be implemented in an agent that is expected to show human-centered artificial intelligence (AI). RPDM (see Fig. 1) explains how human decision-makers plan, in the event of an emergency, to mitigate the aftereffects or to save life and property [68]. The model argues that people are naturally inclined to making a plan based on their experiences [33] and intuition or intuitive knowledge [34], especially when the context has certain important elements such as time stress and high stakes. The nature of Klein’s RPDM model is qualitative or perhaps philosophical where specific details regarding the kinds of methods to use for decision-making and planning have not been specified, which the authors of this work believe would be different for different people. This study identifies tools and methods suitable for the design and development of an agent model that satisfies the RPDM principles to the extent practicable.



Nowroozi et al. [42] proposed a model of RPD called Computational-RPD (C-RPD) and defined the constructs of RPD in *Unified Modelling Language* (UML). Although C-RPD's general form is slightly more detailed than the original RPD, and the authors claim that different sections of their work describe different constructs of RPD. It is unclear how the modeling was performed; C-RPD does not seem to add a scientific methodology that may be considered as a general model covering the concepts in RPD. For example, how can "Evaluate Actions" be done quantitatively, or how can an agent build stories. Will it be a process that incorporates if-then-else conditions, where the consequent comes by interacting with the physical world¹? Or by using an old belief about how the world reacts to when the condition in the if-clause is true? Or will it be a hard-coded knowledgebase where each action has been assigned some pros and some cons, and the agent or the model needs only to fetch the required information? Such questions require a thorough investigation into how each concept in RPD can be modeled separately into different modules, and then how interactions among the modules could be setup so that the overall activity of all modules, combined together, may resemble RPD. Norling [40], and Norling et al. [41] proposed a Belief-Desire-Intention (BDI) based agent model by integrating it with RPD so that the agent can behave more like a human when it comes to deciding something. The agent model can be used to populate a multi-agent simulation environment. Ji et al. [29] proposed an RPD based model that

¹ Consider an agent having a class A fire extinguisher (such as water), willing to apply to a fire due to flammable liquids. If the agent applies fire extinguisher, the result could tell the agent whether that action was good or bad. The fire due to flammable liquids will spread by application of pouring water!.

can be used to analyze drug effects. Based on the experience of how a military commander contributes to decision-making during warfare operations, Sokolowski [58], uses RPDM to capture the dynamics of human mental processes that are involved in decision-making at critical situations.

The authors could not find studies suggesting a rigorous methodology to implement the RPD model. The majority of the literature seen, even where the researchers claim their model as quantitative, present their realization of RPDM as more descriptive or sometimes less formal than RPDM itself [6, 25, 27, 41, 42, 45, 53]. This work aims to add more precisely defined components of a realization of RPDM. For example, the SA part is modeled as an experiential-learning and decision support (ELDS) module, which is based on a Markov logic network (MLN) that needs training to acquire experience. An informational theory based account on modeling SA is given in [17], which is based on Barwise and Perry's [2] situation semantics. A common approach to quantitative modeling of SA involves Bayesian networks (BN) [26, 39]. However, BNs do not support cyclic dependencies that may arise in the causal structure among the factors or conditions on which a situation is dependent. To overcome this limitation Domingos and Richardson [19] proposed Markov logic, whereby a Markov network, which supports cycles, is developed based on information represented in the form of first-order-logic (FOL) rules. The mental simulation component is considered here as a *cause-and-effect* phenomenon [33], p. 89–90), and is proposed to be represented in the form of Bayesian formalism [46]. Lastly, the diagnostic mechanism of RPDM is modeled as an ontology of the domain in which the agent is supposed to operate. An ontology is considered as a tool to represent a set of concepts and their relations in a domain of interest. Sowa [60] exploits ontologies to represent different situations in the world. Because the purpose of OBR module is to diagnose a situation based on common knowledge of the domain of interest, therefore, the choice of using an ontology to represent that knowledge, and thereby suggesting possible matching situations seems reasonable, unlike other approaches to SA that require training (as in MLNs) or prior probabilities (as in BNs).

A recent study [26] exploits RPDM to model human pilot behavior during midair encounters. A fundamental difference between this work and earlier works is in the way SA is modeled. Hu et al. [26] use Bayesian network for situation awareness unlike previous attempts, e.g. [42] where the authors use a direct count on the number of matched features, e.g., by using a similarity criterion, see [22], as a sufficient representation of SA. The pilot models are important to study midair encounter scenarios. The model proposed in [26] simplifies the diagnostic mechanism originally proposed in RPDM by proposing that if a situation is not recognized as typical at the first place, then the model will ask for more information for the recognition of the situation, but the same mechanism, BN, will be used the second time too. Our main concern is why all-important information was not sent to the model in the first place even though it was available through the sensors? Also, what is the criterion to decide how much information will be sufficient for decision-making in the first place? The RPDM says that the diagnostic mechanism should incorporate, at the very basic level, some level of feature-matching [33], p. 91). At the advanced level of diagnosing a situation, a point should come when all the matched features of a situation suggest a larger picture. This is where the authors of the present study think that *story building* should come into play. We also think that

Table 1 A comparison/mapping of the major concepts of Klein’s RPD model with the components in the proposed realization, and in some previous works

Klein’s RPDM	Proposed model	Model A	Model B	Model C
Experience the situation	Input cues	Input cues	Input cues	Input cues
SA based on experience	ELDS module that uses MLN	Bayesian classifier (BC)	Feature by feature matching	Bayesian classifier
SA: Diagnostic mechanism/story building/feature matching	OBR module that uses ontology for story building	Not modeled separately. Diagnostics is done by providing more cues to the BC	Not modeled.	Not modeled
Expectation modelling	Stored as part of plans	Stored as part of solution (plan)	Stored as part of solution (plan)	Not modeled
Action Evaluation: Mental simulation	BN	Specific to midair collision scenarios	Not specified	Not modeled
Action (plan) selection	BDI framework	Not modeled	Not specified	Not modeled
Plan execution	BDI framework	Not modeled	Not specified	Not modeled

Model A is proposed in [26], Model B is proposed in [42], and Model C is proposed in [37]

there must be involvement of an inference mechanism in order to decide which story best suits the matched symptoms or cues of a situation under consideration. The present study exploits ontological-based reasoning (OBR) that uses feature-matching between the available features (not dependent on new or more information) and the ontological knowledge of the agent as opposed to the operational or experience-based knowledge to dig out and give the situation a *name*. Table 1 explains how each concept of Klein’s RPD model may be mapped onto the constructs proposed in the present study. Also, OBR supports inference based on which a recognized situation can be used to suggest a more meaningful interpretation. For example, a situation: “a cat is on a mat”, may mean something about the past of the cat, by interpreting this as, “the cat has taken her meal”. Or by connecting a current situation into a future state, which is the requirement of level 3 SA [20], for example, if the situation “a fire is spreading” is related with another situation “people must escape”, then such a general (domain) knowledge is an important tool for an expression of rational behavior.

“Background concepts” section describes some background concepts, which will help develop an understanding of this study. “Methodology” section explains the methodology proposed here, which includes the development of ELDS and OBR modules. In “Implementing the proposed realization of RPDM model: a case study” section, we present a case study that explains how the methodology of “Methodology” section can be implemented in the form of an agent. The case study in “Implementing the proposed realization of RPDM model: a case study” section is based on an experiment that is used to collect human performance data, which is later used for validating the simulated results from the proposed RPDM based agent model. The ELDS, and OBR modules, which were proposed in “Methodology” section, are developed and explained in detail in this section, and simulation are performed. “Conclusion” section concludes the study with future directions. Some background details and partial computer code are listed in Appendices.

Background concepts

Some important concepts are described in this section. Details about RPDM can be found in Appendix A.1. Appendix A.2 describes background information about the simulator used in this study.

Ontology

Ontology is defined as, “The study of the categories of things that exist or may exist in some domain” [60], p. 492). The result of such a study comes in the form of a catalog that contains *types* of things that exist in a domain D from the point of view of a person who uses a language L to talk about D . There are different Conceptual Structures (CS) that can be used to express knowledge about things, in terms of types and relations, in an ontology.

In [30, 32], the authors propose four types of CSs: *type*, *relation-type*, *individual*, and *situation* to define an ontology. Formally, a CS can be defined in terms of a conceptual graph (CG), which is a bipartite graph between concept nodes and the relations among the concepts [59]. Because an ontology provides a context for representing domain knowledge, the present work exploits the formalism of ontology to provide the agent the knowledge about the domain in which it is likely to operate. Using the proposed ontology (“The ontology-based reasoning module” section), the agent would be able to retrieve meaningful knowledge and can reason about it. Also, representation of domain knowledge in the form of a separate ontology would make the system modular in that the operational knowledge, which comes through experience, can be represented in a separate formalism. The separation of operational knowledge from domain knowledge has benefits in many respects, such as analyzing domain knowledge, making domain assumptions explicit, reusing the domain knowledge, and sharing of the domain knowledge [23].

Markov network

A Markov network (MN) is composed of a graph G and a set of potential functions ϕ_k . G has a node for each variable, and MN has a potential function for each clique in G . A clique of a graph G is a *complete subgraph* of G . A potential function is a non-negative real-valued function of the configuration or state of the variables in the corresponding clique. The joint distribution of the variables X_1, X_2, \dots, X_n can be developed to understand the influence of a site, i.e., a variable, on its neighbors [50] as defined below:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{[k]}) \quad (1)$$

where $x_{[k]}$ is the configuration of the k th clique, i.e., the values of the variables in the k th clique. Z is partition function for normalization, $Z = \sum_{x \in \Omega} \prod_k \phi_k(x_{[k]})$.

Markov logic network

Because a random variable assigned with a value can be considered as a proposition [24], p. 58. Domingos and Richardson [19] define MN by first considering the variables

as rules/formulas in a FOL. Unlike FOL, a formula in MLN is assigned a weight (a real number), not just the Boolean `true` or `false`. Formally, an MLN L is defined as a set of pairs (F_i, w_i) with F_i s being the formulas and w_i s being the weights assigned to the formulas.

If $C = \{c_1, c_2, \dots, c_{|C|}\}$ is the set of constants or ground predicates (the facts), then L induces a Markov network $M_{L,C}$ such that the probability distribution over possible worlds x is given by:

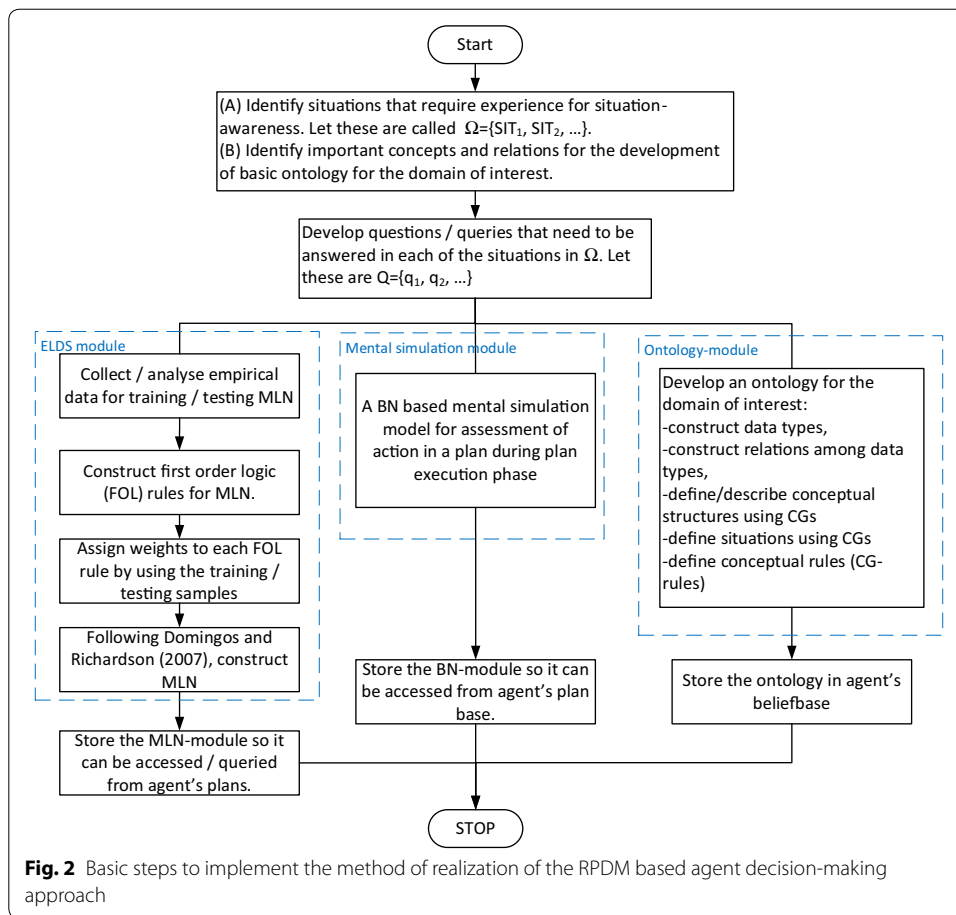
$$P(X = x) = \frac{1}{Z} \exp \left(\sum_i w_i n_i(x) \right) = \frac{1}{Z} \prod_i \phi_i(x_{[i]})^{n_i(x)} \quad (2)$$

where $n_i(x)$ is the number of `true` groundings of F_i in x , $x_{[i]}$ is the state or configuration (i.e., the truth assignments) of the predicates in F_i , and $\phi_i(x_{[i]}) = e^{w_i}$.

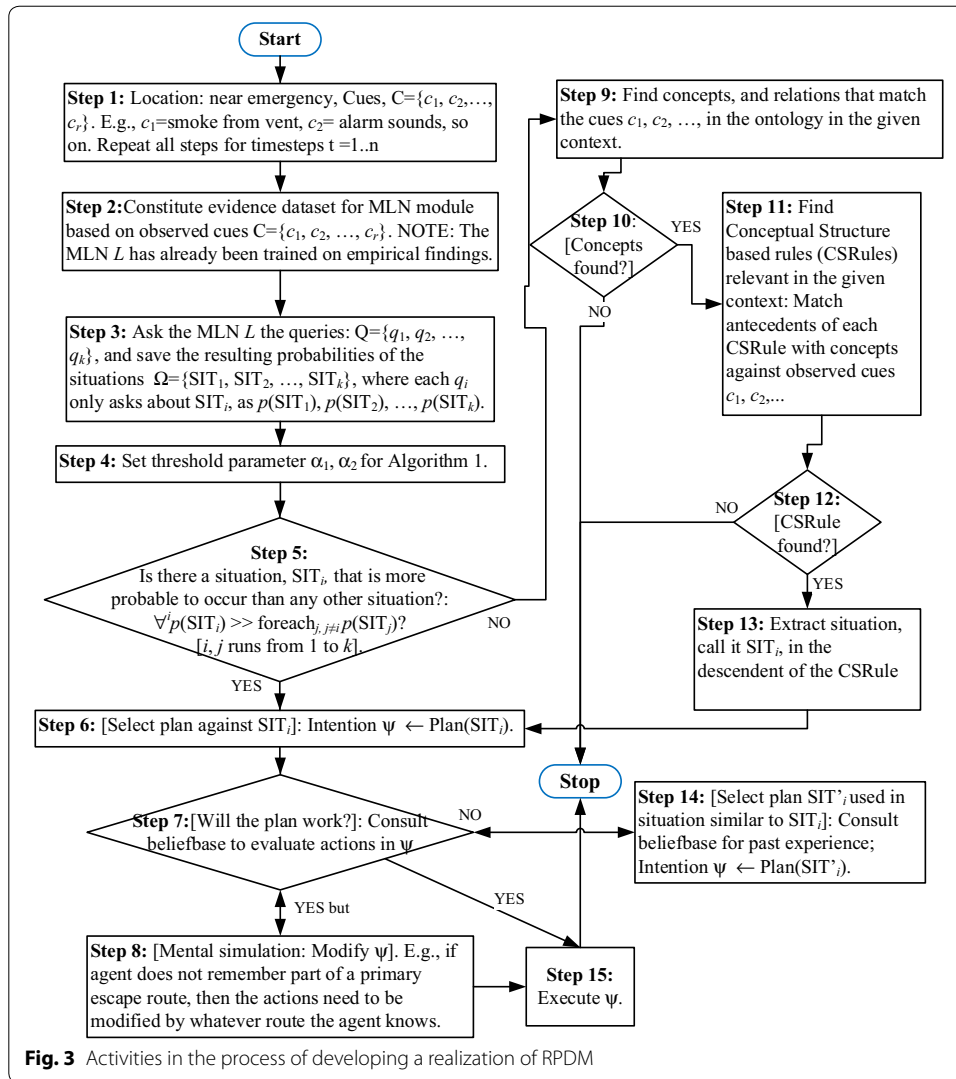
Methodology

The kind of situations suitable for constructing a realization of the RPDM approach for artificial agents should include the ingredients of NDM [43]. At the conceptual level, the agent decision-making process is conceived here in terms of the mental modalities suggested in Bratman's theory of practical reasoning [4]. Specifically, these mental attitudes are a belief, desire, and intention (BDI), which are the basis of the BDI-agent model [51]. The proposed agent model has a *beliefbase* that contains context information, past experiences, an ontology [60] about the domain in which the agent is being operated, and any other kind of information that affects a possible deliberation step. A planning scheme is responsible for matching available cues and a plan to be executed. In simple words, a planning scheme takes all the sensory observations (cues), assesses the situation, selects a plan for execution, and performs mental simulation if necessary. Figure 2 describes the general steps needed to develop the ELDS module based on MLN, OBR module containing the ontology for the domain in which the agent operates, a module to perform mental simulation as a cause-and-effect mechanism using a BN, and where these modules should be stored within the BDI-framework so that upon receiving the sensory data, the agent can have access to each type of knowledge. Figure 3 describes the flow of control, starting from collecting cues in the environment to having a decision for what needs to be done when a situation unfolds demanding action on the agent's part.

The approach of this work involves modeling decision-making at three levels. The first is the situations that are recognized as *typical* by the ELDS-module, i.e., the situations that can be inferred by the MLN inference mechanism. The second is the situations where MLN performs poorly by predicting approximately the same probabilities for more than one situation so that it becomes difficult to distinguish among the candidate situations as being the one currently observed. These are the situations when the agent receives inadequate or conflicting cues at a single time step at a given location in the environment. An agent in such a situation is considered as the one whose experience does not relate well enough to the situation at hand and who has to rely on some basic knowledge to classify/recognize a situation based on perceived cues. This level of decision-making is modeled here in terms of an ontology about possible situations that could arise. These two levels of decision-making—the



one based on experience, and the one involving feature-matching using an ontology—are governed by a third level that decides in what circumstances the agent should select one of these levels. Algorithm 1 describes this higher level of decision-making. Lines # 2–7 deal with decision-values taken from MLN based inference, and lines 8–11 calls a method `DIAGNOSE-SITUATION` that queries the agent’s ontology by using available cues as concepts and then extracts CS-Rules that satisfy the concepts. The working of the `DIAGNOSE-SITUATION` method can be understood as actions taken in steps Steps 9–13 in Fig. 3. For example, if an agent has a visual of smoke in the messhall, and for some reasons it is unable to get other cues, then the agent will take *smoke* and *messhall* as concepts and search the ontology for possible relations. If a relation is found the agent applies inference to explore connected or related situations that contain specific or doable actions. These actions are the final output of the agent. The `DIAGNOSE-SITUATION` method corresponds to Klein’s variation 2 of the RPDM model [33], p. 26) as explained in the preceding section. Steps 1–8 correspond mainly to recognize the given situation, where there are a finite number of observable cues represented as $\{c_1, c_2, \dots\}$, based on MLN L that is developed by using the FOL rules.



Algorithm 1: A general higher-level decision-making.

Assumptions: An MLN can distinguish among m possible situations. An i th situation is SIT_i . These are considered as typical situations for which the agent is trained. $p(SIT_i)$ is the probability of occurrence of SIT_i . $PLAN_{SIT_i}$ refers to a plan associated with the situation SIT_i .

Inputs: $\alpha_1 >> \alpha_2$ and ϵ is a positive real number near zero. Theoretically, $\alpha_1, \alpha_2 \in [0, 1]$.

Output: The decision Φ , which is a plan having actions to perform.

1. for $timesteps = 1$ to n do
2. if $p(SIT_1) > \alpha_1$ && $[p(SIT_2) < \alpha_2, p(SIT_3) < \alpha_2, \dots, p(SIT_m) < \alpha_2]$ then
3. $\Phi \leftarrow PLAN_{SIT_1}$
4. else if $p(SIT_2) > \alpha_1$ && $[p(SIT_1) < \alpha_2, p(SIT_3) < \alpha_2, \dots, p(SIT_m) < \alpha_2]$ then
5. $\Phi \leftarrow PLAN_{SIT_2}$
6. :
6. else if $p(SIT_k) > \alpha_1$ && $[p(SIT_1) < \alpha_2, \dots, p(SIT_{k-1}) < \alpha_2, p(SIT_{k+1}) < \alpha_2, \dots, p(SIT_m) < \alpha_2]$ then
7. $\Phi \leftarrow PLAN_{SIT_k}$
8. else if $|p(SIT_1) - p(SIT_2)| \leq \epsilon$ && \dots && $|p(SIT_{m-1}) - p(SIT_m)| \leq \epsilon$ then
9. $\Phi \leftarrow PLAN_{DIAGNOSE-SITUATION}$
10. return Φ .

Table 2 The FOL rules for developing the MLN L suitable for emergency response in FIRE and EVACUATE emergencies

#	Rules
1.	$\neg L(ag, al, t) \Rightarrow \neg R(ag, al, t)$
2.	$L(ag, +al, t) \wedge HITR(ag, +mloc, t) \wedge BST(ag, +al, t) \Rightarrow R(ag, +al, t)$
3.	$L(ag, al, t) \Rightarrow HSES(ag)$
4.	$ST(ag, thrt, t) \Rightarrow HSES(ag)$
5.	$(HFO(ag, +p_a, t) \wedge FPA(ag, +p_a, t) \wedge KETPA(+p_a, +emgType)) \vee$ $(ST(ag, +thrt, t) \wedge KETT(+thrt, +emgType)) \vee$ $(L(ag, +al, t) \wedge HITR(ag, +mloc, t) \wedge KETA(+al, +emgType) \wedge BST(ag, +al, t))$ $\Rightarrow HES(ag, +emgType, t)$
6.	$HES(ag, FIRE, t0) \wedge HES(ag, EVACUATE, t1) \wedge Gt(t1, t0)$ $\Rightarrow \neg HES(ag, FIRE, t1)$
7.	$HES(ag, FIRE, t) \Rightarrow \neg HES(ag, EVACUATE, t)$
8.	$HES(ag, EVACUATE, t) \Rightarrow \neg HES(ag, FIRE, t)$
9.	$HFO(ag, +p_a, +t) \wedge FPA(ag, +p_a, +t) \wedge KMLPA(+p_a, +mloc)$ $\Rightarrow HITR(ag, +mloc, +t)$
10.	$L(ag, +al, t) \wedge R(ag, +al, t) \wedge BST(ag, +al, t) \wedge KMLA(+al, +mloc)$ $\Rightarrow HITR(ag, +mloc, t)$

The experiential-learning and decision-making module

The purpose of the ELDS module is to support decision-making based on experience. In the real world, different people consider the same rules differently in terms of how effective they are in assisting a person for deciding on a given situation. That is, there is a diversity among people for adopting a method for a given decision problem. This phenomenon gives rise to people having different experiences about the same or similar situations with different beliefs about the choices they make. Klien's RPDM model considers this diverse nature of experiences among experts by generally describing that a situation recognition task should result in four by-products: relevant cues, typical actions, plausible goals, and expectancies. The RPDM model does not argue as to how the goal of computing the four by-products of recognition should be achieved. The present study argues that an experiential learning technique is a suitable choice to capture the crux of situation recognition in the Endsley's SA model [20], because this way, different agents can have different experiences about a domain of choice. Rules regarding recognition of fire (FIRE) and evacuate (EVACUATE) emergencies are proposed in Table 2. As an example of how agents with different experiences can be made in a real system, consider rule #9 in Table 2:

$$HFO(a_1, +p_a, +t) \wedge FPA(ag, +p_a, +t) \wedge KMLPA(+p_a, +mloc) \Rightarrow HITR(ag, +mloc, +t)$$

This rule says that if an agent (a_1) *Has Focus On* (HFO) the PA (p_a) announcement at some time t , and a_1 is able to understand or *Follow the PA* (FPA), and a_1 knows what to do in that specific PA announcement (the predicate $KMLPA(p_a, mloc)$ is stored as a fact that means the agent knows which muster location is used in which PA), then a_1 should develop an intention (represented by the predicate $HITR$) according to its knowledge about that specific PA and, thereby, the associated weight, w , of the rule. For example, in the case of a PA related to the GPA alarm, a_1 's intention should be to move to the primary muster station; in the case of a PAPA alarm, the intention should be to move to the alternate muster station. However, if an agent keeps repeating a mistake by, say,

Table 3 Joint probability table for possible worlds entails by rule#9

HFO	FPA	KMLPA	$J_1 = \text{HFO} \wedge \text{FPA} \wedge \text{KMLPA}$	$J_2 = \text{HITR}$	$J_1 \Rightarrow J_2$	$p(\cdot)$
¬HFO	¬FPA	¬KMLPA	False	¬HITR	True	e^w/Z
¬HFO	¬FPA	¬KMLPA	False	HITR	True	e^w/Z
¬HFO	¬FPA	KMLPA	False	¬HITR	True	e^w/Z
¬HFO	¬FPA	KMLPA	False	HITR	True	e^w/Z
¬HFO	FPA	¬KMLPA	False	¬HITR	True	e^w/Z
¬HFO	FPA	¬KMLPA	False	HITR	True	e^w/Z
¬HFO	FPA	KMLPA	False	¬HITR	True	e^w/Z
¬HFO	FPA	KMLPA	False	HITR	True	e^w/Z
HFO	¬FPA	¬KMLPA	False	¬HITR	True	e^w/Z
HFO	¬FPA	¬KMLPA	False	HITR	True	e^w/Z
HFO	¬FPA	KMLPA	False	¬HITR	True	e^w/Z
HFO	¬FPA	KMLPA	False	HITR	True	e^w/Z
HFO	FPA	¬KMLPA	False	¬HITR	True	e^w/Z
HFO	FPA	¬KMLPA	False	HITR	True	e^w/Z
HFO	FPA	KMLPA	True	¬HITR	False	$1/Z$
HFO	FPA	KMLPA	True	HITR	True	e^w/Z

The probability $p(\{\text{HFO}, \text{FPA}, \text{KMLPA}, \neg\text{HITR}\}) = 1/Z$ represents the probability of a world that is inconsistent with rule#9. The probabilities for all other possible worlds are equal to e^w/Z , where w is the weight assigned to the rule. The operator ' \Rightarrow ' is for logical implication

attributing GPA to alternate muster station rather than the primary, then in the event of a FIRE emergency this agent will likely move to the alternate muster station even though it is contrary to the required action.

In the current study, the variables p_a , t , and $mloc$ belong to the sets $A = \{\text{PAGPA}, \text{PAPAPA}\}$, $T = \{t0, t1\}$, and $M = \{\text{MESSHALL}, \text{LIFEBOAT}\}$, respectively. This gives rise to eight different permutations resulting from grounding rule#9 for the constants in the sets A , T , and M . As there are four predicates in rule #9, there will be $2^4 \times 8 = 128$ total number of different worlds altogether. For brevity, assume that the variables p_a , t , and $mloc$ belong to sets each having a single constant. So, let $p_a = \{pa\}$, $t = \{t\}$ and $mloc = \{m\}$. Then, there will be $2^4 = 16$ possible worlds, as shown in Table 3, where w shows the weight assigned to the rule, and the table excludes the parameters of each predicate for better readability. The probability that the world that is inconsistent with rule #9 occurs, i.e., the probability $p(\{\text{HFO}, \text{FPA}, \text{KMLPA}, \neg\text{HITR}\})$ is equal to $1/Z$ is less likely than all other probabilities as shown in Table 3, provided $w > 0$. Here Z is the partition function as described in “Background concepts” section. The probability for a world to be true depends on the weight w assigned to each rule. Agents with the same rules differing in respective weights are expected to behave differently.

An explanation of the FOL rules

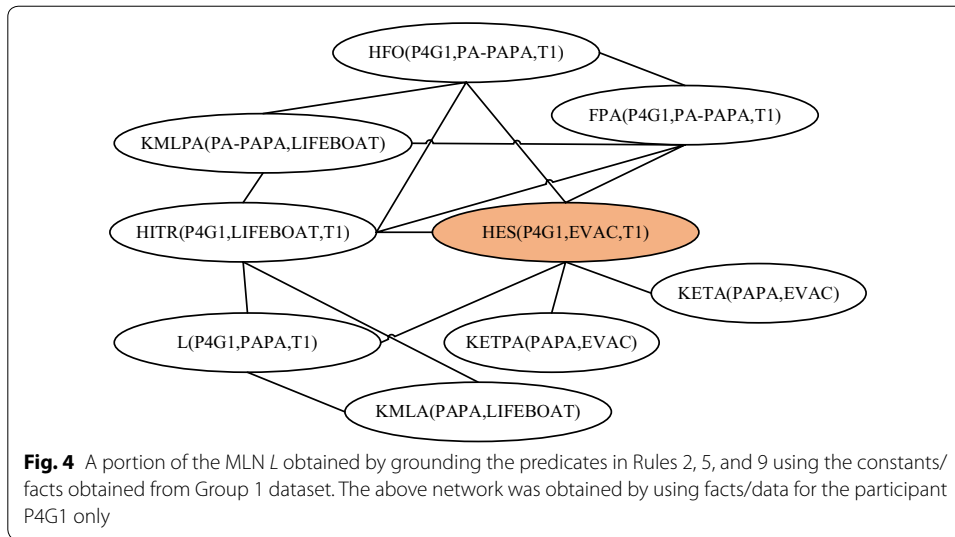
A set of FOL rules are proposed in Table 2 so that an agent can recognize the FIRE and EVACUATE situations in the similar way as a human counterpart recognizes them. The preconditions (antecedents of FOL rules) used here are common among experts and have been suggested in earlier studies [8, 21, 49, 56, 57, 61–65]. Similar work is reported in [38] where the authors constructed decision trees based on some of the preconditions used in this study, such as the presence of hazard, route direction in PA that actually is a byproduct of understanding the PA.

Rule #1 in Table 2 is a hard constraint which signifies the fact that for recognizing a sound, it must have been heard first. Alarms are made to produce loud and clear audio frequencies so that people can hear the alarm sound, but somebody who is hearing an alarm sound does not necessarily pay attention to it. Several studies [49, 63, 66] show that people need training to be vigilant about alarm sounds.

There are several sources that give *intention* a vital role in deliberation [4, 62]. In rule #2, an agent must be listening to an alarm, which means she is paying attention to the alarm, and at the same time developing the deliberative intention [4, p. 56], due to deliberation that involves carefully listening the alarm, to moving to a (particular) muster location. Because the agent has formed the intention just after listening to the alarm at time t , and the deliberation involving the act of listening or the formation of intention is done before having to see a visual cue about a possible threat (the predicate `BST` ensures that the intention was formed before seeing a threat), it clearly means that the alarm has been recognized at the same time. Nonetheless, the agent cannot act upon the intention unless recognition of the alarm is made, because deliberation requires the location of the muster station, which can only be decided after recognition of the alarm. Therefore, as in rule #2, if the intention is made before recognition, then it needs to be updated with the value of the muster location (i.e., `MESSHALL` or `LIFEBOAT`) at some later time, say t_2 , before performing the actions in the intention and according to the result of the recognition of the alarm. Rule #2 thus models `HITR` as a policy-based intention as explained in the literature [4, p. 56], that is, the agent will form a general intention of moving to a muster location right at the time of listening to an alarm, and will later determine which muster station is the right choice.

Rules #3 and 4 have the same descendent: *Has Some Emergency Situation*, which is referred to here by the predicate name `HSES` (see Table 2). A true value of `HSES` means that the agent knows there is some emergency. Having `HSES` true does not necessarily tell the agent-specific details about the kind of emergency that has occurred. Rules # 3 and 4 say that an agent will be aware of ‘some’ emergency situation if it just listens to an alarm or observes a threat.

PA announcements are important cues in a developing situation [8, 21, 61, 65]. PAs are verbal announcements with clear words detailing the situation with the type and location of a hazard, other affected areas, and possible plan to assist evacuation. An agent needs to focus on PA wordings in order to gain advantage of the message in a developing emergency. Stress is considered a factor that influences focus of attention in offshore environments [57]. In short, the predicate `HFO` is `true` when the agent has a focus on a PA being uttered. An agent that is engaged in all activities except what is communicated in the PA is defined to have no focus, whereas one that suspends its current engagements and begins performing the required actions is considered to have focus on the PA. Similarly, if an agent, while moving, suddenly changes its course because of instructions given in the PA a moment before, this also considered to have exhibited a clear sign of deliberative intention [4] in response to the PA. This deliberative intention is captured in rule #9 by the predicate `HITR` when the agent considers `HFO` and `FPA`, and has a prior knowledge about possible deliberation steps (the predicate `KMLPA` that stands for *Knows Muster Location according to PA*). The predicate `FPA` is used to demonstrate the requirement of following the PA. If `HFO` is `true`, but `FPA` is `false`, it means that,



though the agent had focus on the PA's words, it is confused or does not have understanding of the situation, and therefore, the agent is unable to follow the PA. Rule #5 is a disjunction of three different rules: the first determines SA about the emergency based on focus and understanding of PA, the second uses direct exposure to the threat/hazard, and the third is based on the recognition of alarms. This last disjunct in rule #5 uses the predicate KETA to link an alarm to the corresponding situation or emergency type because that is needed to conclude in the consequent predicate HSES.

Rule #6 uses time as factor for ignoring an earlier understanding about a FIRE situation when FIRE is escalated to EVACUATE. That is, if an agent has awareness about a FIRE at t_1 , and at some later time t_2 the situation escalates to EVACUATE, then there is no need to keep the impression of FIRE situation because the agent needs to act according to EVACUATE situation. Rules #7 and 8 are to ensure that FIRE and EVACUATE are two distinct types of situations, besides that EVACUATE may occur because of a fire [8, 61]. Rule #10 determines a formation of intention to move to a muster location by listening to an alarm (the predicate L), recognition of alarm (the predicate R), and belief about what is needed in that particular alarm type (the predicate KMLA that stands for Knows Muster location against the Alarm). In this case, the formation of deliberative intention [4] is based on deliberation about the act of listening and recognizing the alarm type.

Training the ELDS module

The dataset Tr is used for training the ELDS module, and the dataset Te is used as testing/evidence while querying the ELDS's MLN L . The model is trained by employing a discriminative learning method [18, 55] using the software package Alchemy [1] so that weights can be assigned to the rules presented in Table 2. A fragment of the MLN L is depicted in Fig. 4. The nodes in Fig. 4 are obtained for each possible grounding of each predicate appearing in a formula. An edge between two nodes means that the corresponding ground predicates have appeared at the same time in at least one grounding of one formula in L .

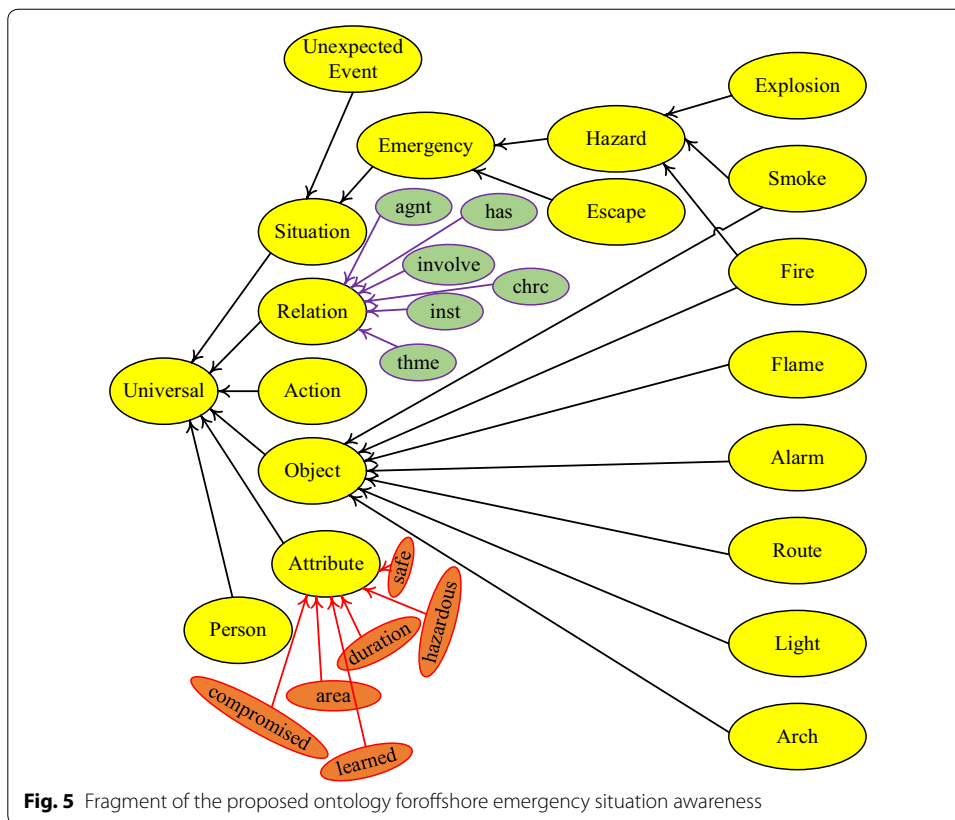


Fig. 5 Fragment of the proposed ontology for offshore emergency situation awareness

The ontology-based reasoning module

The OBR module incorporates the need for basic concepts that may come into one’s mind when an emergency is encountered that involves fire, smoke, evacuation, or escape. These basic concepts and those derived from them have been defined in the ontology by exploiting the formalism of Sowa [59, 60], that is, by using CGs. Figure 5 shows a fragment of important concepts represented in the proposed ontology for offshore emergency situations.

The conceptual relations: *agent* (agnt), *attribute* (attr), *characteristic* (chrc), *experiencer* (expr), *instrument* (inst), *object* (obj), and *theme* (thme) are used here as defined in [59], p. 415–419). The concept agnt does not refer to the concept of agent as defined in AI literature, rather it is a relation used in conceptual structures to refer to a relation that links an [ACT] to and an [ANIMATE], where the ANIMATE concept represents the actor of the action. The concept of ACT is defined as an event with an animate agent.

Definition 3.2.1 The relation agnt links the concept [ACT] to [ANIMATE], where the ANIMATE concept refers to an actor of the action. Example: A CG for “A Man moves to a destination” in the linear form (LF) will be represented as:

$$\begin{aligned}
 [\text{MoveTo}] - (\text{agnt}) &\rightarrow [\text{Person}], \\
 &- (\text{attr}) \rightarrow [\text{Destination}].
 \end{aligned}$$

Definition 3.2.2 The relation *attr* links [Entity: *x] to [Entity: *y], where *x has an attribute *y. Example: Fire has flame. The CG is: [Fire] \rightarrow (*attr*) \rightarrow [Flame] such that Fire and Flame are represented as two concepts of type Entity, and Fire has an attribute Flame.

Definition 3.2.3 The relation *chrc* links [Entity: *x] to [Entity: *y] such that *x has a characteristic *y. Example: *Emergency is a danger to people and property*. The CG is: [Emergency] \rightarrow (*chrc*) \rightarrow [danger] \rightarrow [Person_Property].

Definition 3.2.4 The relation *expr* links a [State] to an [Animate], who is experiencing that state. For example, because Emergency is defined here as a situation as well as a state, therefore, the concepts in the sentence, “Emergency is experienced by people”, are described as CG by [Emergency] \rightarrow (*expr*) \rightarrow [Person].

Definition 3.2.5 The relation *inst* links an [Entity] to an [Act] in which the entity is causally involved. For example, the CG [Fire] \leftarrow (*obj*) \rightarrow [Produce] \rightarrow (*inst*) \rightarrow [Combustion] reflects a causal relationship between the chemical process of combustion and the birth of a fire.

Definition 3.2.6 The relation *obj* links and [Act] to an [Entity], which is acted upon. For example, in the event of an emergency “a person moves to the secondary muster station (LIFEBOAT)”, is represented in the ontology as an descendent of a CS-Rule as:

Antecedent part

[MESSHALL] – (*attr*) \rightarrow [Compromised],
– (*expr*) \rightarrow [Person].

Descendent part

[MoveTo] – (*agnt*) \rightarrow [Person],
– (*attr*) – [Destination] – (*obj*) \rightarrow [LIFEBOAT].

Definition 3.2.7 The relation *thme* is to represent a thematic role. For example to express the intent in the sentence, “Muster station has hazard”, one can write the CG as [MusterStation] – (*thme*) \rightarrow [Hazard] (see [60] for a detail account on thematic roles in ontologies).

Definition 3.2.8 The relations *require* (*req*) and (*involve*) links a [Person] to an [Action], and an [Action: x] to an [Action: *y], respectively where *x involves *y. As an example the descendent in following CS-Rule represents the use of *req* relation.

Antecedent part:

[Place] – (*thme*) \rightarrow [Hazard],
– *expr* \rightarrow [Person].

Descendent part:

[Person] – (*req*) \rightarrow [ImmediateAction] – (*involve*) \rightarrow [RaiseAlarm],
 \leftarrow (*agnt*) – [MoveOut].

Definition 3.2.9 The concept `Combustion` is defined as an act of burning. The CG is:

$$[\text{Combustion}] - (\text{actOf}) \rightarrow [\text{Burning}].$$

Definition 3.2.10 The concept `Fire` is defined as an entity that has attributes of heat, light, flame and that is produced as a result of combustion. The CG is:

$$\begin{aligned} [\text{Fire}] - (\text{attr}) &\rightarrow [\text{Heat}], \\ &- (\text{attr}) \rightarrow [\text{Flame}], \\ &- (\text{attr}) \rightarrow [\text{Light}], \\ \leftarrow (\text{obj}) - [\text{Produce}] - (\text{inst}) &\rightarrow [\text{Combustion}]. \end{aligned}$$

Definition 3.2.11 The concept `Smoke` is defined as a child concept of `[Hazard]` that is produced as a result of combustion. The CG is:

$$\begin{aligned} [\text{Produce}] - \\ &- (\text{inst}) \rightarrow [\text{Combustion}], \\ &- (\text{obj}) \rightarrow [\text{Hazard} : \text{super}]. \end{aligned}$$

Definition 3.2.12 The concept of muster station is defined as a place of temporary refuge during an emergency. It is represented as:

$$\begin{aligned} [\text{MusterStation}] - (\text{attr}) - \\ &\rightarrow [\text{TemporaryRefugeArea}] - (\text{attr}) \rightarrow [\text{Duration}] - \\ &- (\text{involve}) \rightarrow [\text{Emergency}]. \end{aligned}$$

Definition 3.2.13 The concept of emergency is classified as a situation, and as a state. It is formally defined in terms of a CG as:

$$\begin{aligned} [\text{Emergency}] - \\ &- (\text{isa}) \rightarrow [\text{UnexpectedEvent}], \\ &- (\text{isa}) \rightarrow [\text{Situation} : \text{super}], \\ &- (\text{req}) \rightarrow [\text{ImmediateAction}], \\ &- (\text{attr}) \rightarrow [\text{Duration}], \\ &- (\text{attr}) \rightarrow [\text{Area}], \\ &- (\text{chrc}) \rightarrow [\text{Danger}] \\ &- (\text{to}) \rightarrow [\text{Person_Property}], \\ &- (\text{involve}) \rightarrow [\text{Hazard}], \\ &- (\text{expr}) \rightarrow [\text{Person}], \\ &- (\text{notifiedBy}) \rightarrow [\text{Alarm}]. \end{aligned}$$

Definition 3.2.14 The following CS-rules are stored for memory-based inference:

CS-rule #1

If a muster station, x , gets a hazard, then the muster station, x , will be considered as *compromised*.

Antecedent:

[MusterStation: *x] - (thme) → [Hazard].

Consequent:

[MusterStation: *x] -
 - (attr) → [Compromised],
 - (expr) → [Person].

CS-rule #2

If a person finds the MESSHAL compromised, then the person should move to the LIFEBOAT station.

Antecedent:

[MESSHAL] -
 - (attr) → [Compromised],
 - (expr) → [Person].

Consequent:

[Person] ← (agnt) - [MoveTo] -(attr) → [Destination] - (obj) →
 [LIFEBOAT].

CS-rule #3

If a person finds the LIFEBOAT station compromised, then the person should escape from the platform as quickly as possible.

Antecedent:

[LIFEBOAT] -
 - (attr) → [Compromised],
 - (expr) → [Person].

Consequent:

[Person] ← (agnt) - [Escape] - (actOf) → [ImmediateAction] - (involve) →
 [EMERGENCY].

CS-rule #4

If a person finds a hazard at some location, then the Person should raise alarm and move out of that location.

Antecedent:

[Place] -
 - (thme) → [Hazard],
 - (expr) → [Person].

Consequent:

[Person] –
 ← (agnt) – [MoveOut],
 – (req) → [ImmediateAction] – (involve) → [RaiseAlarm].

Implementing the proposed realization of RPDM model: a case study

A general methodology to prepare a working model of RPDM for agents is described in Figs. 2 and 3. It is not possible to proceed with it unless there are specific modules for ELDS, OBR, and mental simulation. These modules, in turn, require situation-specific data so that rules can be outlined on the basis of which ELDS-module for SA is made, and an ontology for basic terms and general principles can be designed. In this section, we will discuss how the concepts explored in “[Methodology](#)” section can shape a working model for an artificially intelligent agent that makes decisions in the sense of the theory behind the RPDM model as explained in [33]. We will describe an experiment that has been used here for developing some situations in which the proposed methodology of “[Methodology](#)” section may be implemented. Moreover, subsequent subsections will discuss how the insight developed in the experiment is used to develop the ELDS-module and an ontology for basic domain knowledge.

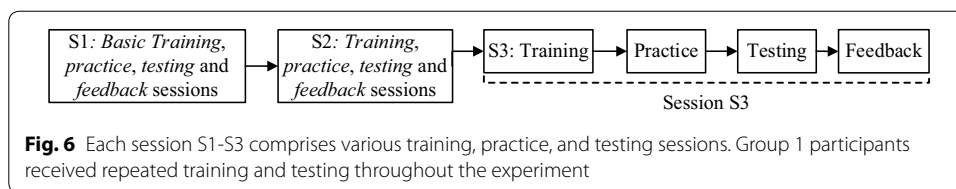
Human-competence measurement in a virtual environment

Smith [56] performed an experiment to assess how training in a virtual environment (VE) for emergency response affects human competence in different emergency egress scenarios. Emergency response training is a regulated part of industrial safety. For example, SOLAS Chapter II-2 Regulation 13 [28] describes specific guidelines about the use of exit signs in escape routes on offshore petroleum platforms. The OSHA fact sheet [44] describes operational features of all escape routes and urges at least two routes for rapid and safe evacuation in an emergency. A thorough investigation into different kinds of accidents, hazards, emergencies, and required responses is given in [8, 10]. Smith’s experiment involved 36 participants divided into two groups: Group 1 containing 17 and Group 2 containing 19 participants. Group 1 participants were trained in several training sessions, and Group 2 participants received only a single basic training exposure (Fig. 6).

Evacuation scenarios and decision tasks

The training curriculum of Smith’s study [56] targeted six learning objectives: (1) establish spatial awareness of the environment, (2) alarms recognition, (3) routes and mapping, (4) continually assess situation and avoid hazards on routes, (5) register at temporary safe refuge, and (6) general safe practices. In the present study, Group 1 participants’ data from cabin-side scenarios are used for validating the simulation results from the agent model proposed in “[Methodology](#)” section. The agent is supposed to operate given the same input as was perceived by participants in the Smith’s experiment.

The participants were tested throughout three separate sessions: S1, S2, and S3, each comprising various training and testing sessions involving a range of activities. The testing sessions were recorded as replay video files so they can be watched later using



AVERT. In cabin side scenarios, session one (S1) comprised two learning (LE2, LE3) and two testing (TE1, TE3) scenarios. At the beginning of S1, the participants were given a 30-min video tour (named LE1) to get acquainted with the virtual platform. As the participants were trained in S1 and S2 prior to S3 it means a compounding training effect from S1 and S2 was already present in S3. Session 2 (S2) targeted training and testing for emergency alarm recognition during muster drills. For cabin-side scenarios, S2 contained two training (LA2, LA3) and testing (TA1, TA3) scenarios. The purpose of S2 was to train the participants for alarm recognition in fire and evacuation emergencies so that they can decide, upon listening to an alarm, which type of emergency has occurred. Session 3 (S3) was developed to train and test the participants for muster drills for fire and evacuation emergencies. In these drills, participants listen to platform alarms followed by public address (PA) announcements, and encounter fires and smoke hazards. S3 comprises two training (LH3, LH4) and two testing (TH1, TH2) scenarios.

The hazards block part of the primary escape route and compromise the primary muster location in TH1. A detailed account on these training/testing scenarios is available in [56]. In scenario TH1, initially a fire broke out in a galley, and a general platform alarm (GPA) began sounding to notify personnel of a FIRE situation. The GPA alarm was also followed by a PA announcement that told the participants the kind of hazard, the location of the hazard, and possible actions needed (where to muster, the primary or the secondary muster station). The protocol instructed the participants to leave their cabins immediately and proceed to the primary muster station and register there by moving the T-card from the *steady* to *mustered* state. After some time, the fire escalated, and the situation turned from FIRE to EVACUATE emergency. This was signaled by a change in the alarm sound from GPA to Prepare-To-Abandon-Alarm (PAPA) followed by another PA announcement. Participants need to decide which muster location is the right choice and which egress route to follow in case the primary escape route becomes inaccessible.

All training and testing scenarios were recorded, and a log file for each participant was maintained that contained specific information about the way the participant proceeded in a scenario towards making a required decision. Factors that play important roles in deciding about the kind of emergency (FIRE or EVACUATE), recognizing alarms, and developing an intention to move to a particular muster location using an escape route are listed in Table 4.

Data collection

All observations were collected in the form of Boolean variables or predicates reported in Table 4. Table 5 reports a sample of data collected through knowledge elicitation that involves breaking each participant’s session into two parts. The first part concerned with the question of recognizing a FIRE emergency and then deciding upon accordingly. The

Table 4 Variables (and corresponding predicate names) to be used in the ELDS module development along with parameter types and description are shown

Factor/variable	Predicate	Parameters	Descriptions
Alarm Recognition	R	(agent, alarm, t)	An agent recognizes an alarm as being GPA or PAPA during time interval t
Focus of attention or has focus on	HFO	(agent, pa, t)	An agent has focus on pa message ^a during time t
Encounters or sees a threat or hazard	ST	(agent, thtType, t)	An agent has seen a hazard of type, thtType, during t
Follows a PA	FPA	(agent, pa, t)	An agent understands the wording in PA.
Intention to move to a specific muster location	HITR	(agent, musterLoc, t)	An agent has (developed) an intention during t to reach a specific muster location
Situation awareness of emergency	HES	(agent, emgSitType, t)	An agent got awareness about the situation type, emgSitType, during time t
Paying attention to alarm	L	(agent, alarm, t)	An agent listens to an alarm during time interval t
Assessment of alarm recognition based on listening alarm	BST	(agent, alarm, t)	An agent listens to an alarm before seeing the threat. This predicate is used in conjunction with others in rules 2, 5, 9 to assess if the alarm recognition is done <i>before seeing a threat</i> (BST). This concludes that the alarm is recognized otherwise the SA might be due to some other factor such as watching a fire
Sensing of an emergency	HSES	(agent)	Based on the antecedent in rules 3 and 4 an agent will get a sense of some emergency without getting further details

^a The PA was made by a verbal announcement of the message, "Attention all personnel! This is the offshore installation manager speaking. We have report of a fire in a galley."

second part involves recognition of EVACUATE emergency and act accordingly. The methodology to collect data for each of the predicate is based on "Observing participants' performing tasks" [9], p. 14–15). There are three methods to perform this type of data collection. The first is based on the approach called *theory* (TT) that says, given the information about a person's observed behaviors, or gestures, an attributor can make inferences about the person's intentions, beliefs and goals [14]. The second approach to mind reading, called *rationality theory* (RT), exploits the use of principles of rationality [16] to attribute different states to others based on their behavior. The third approach used in this work to collect data through re-play videos is referred in cognitive science literature [3, 54] as simulation theory (ST). Appendix B described a set of assumptions that are made about the participants of Smith's experiment.

In order to show how data pertaining to each predicate is gathered from the re-play videos of participants, we present here, for brevity, only the procedure adopted to collect data for the predicates HFO and FPA. The primary way to determine whether a participant had focus on PA wordings was to see if the participant's movement has changed starting with the PA. For instance, if it is observed that as soon as the PA begins the participant starts getting slowed down in speed, or stopped, or

Table 5 A sample of empirical observations for the decision choices made by the five participants

Participant	Predicates									
	L (., GPA, t0)	R (., GPA, t0)	BST (., GPA, t0)	HITR (., MSH, t0)	ST	HFO (., PAGPA, t0)	FPA (., PAGEPA, t0)	HES (., FIRE, t0)		
P4G1	True	False	False	False	True ^a	True	False	True	True	
P5G1	True	True	True	True	True ^b	True	True	True	True	
P6G1	True	True	True	False ^c	True ^d	True	True	True	True ^e	
P7G1	True	True	True	False	True ^f	True	True	True	True	
P10G1	True	True	True	True	True ^g	True	True	True	True	
Participant	Predicates									
	L (., PAPA, t1)	R (., PAPA, t1)	BST (., PAPA, t1)	HITR (., LBS, t1)	ST	HFO (., PAPAPA, t1)	FPA (., PAPAPA, t1)	HES (., EVAC, t1)		
P4G1	True	False	False	True	False ^h	True	True	True	True	
P5G1	True	True	True	True	True ⁱ	True	True	True	True	
P6G1	True	True	True	True	True ^j	False	False	True ^k	True	
P7G1	True	True	True	True	True ^l	False	False	True ^m	True	
P10G1	True	False	False	False	True ⁿ	False	False	False ^o	False	

The time interval t0 starts with the beginning of the scenario until the GPA alarm ends. The interval t1 starts when the PAPA alarm begins sounding, i.e., just after the GPA ends and the situation escalates from FIRE to EVACUATE. It ends at the end of the scenario

^a Parameters observed are: P4G1, SMK_VENT, and t0

^b Parameters observed are: P5G1, SMK_STAI, and t0

^c The intention for moving to the messhall was developed late during the beginning of the interval t1

^d Parameters observed are: P6G1, SMK_STAI, and t0

^e P6G1 kept the impression of FIRE until it recognizes EVACUATE situation in t1

^f Parameters observed are: P7G1, SMK_STAI, and t0

^g Parameters observed are: P10G1, SMK_MSHA, and t0

^h P4G1 did not see any threat during t1

ⁱ Parameters observed are: P5G1, SMK_MSHA, and t1

^j Parameters observed are: (P6G1, SMK_MSHA, t1), and (P6G1, SMK_STAI,t1)

Table 5 (continued)

- ^k P6G1 kept impression of FIRE emergency during the beginning of the interval t_1 , later understood escalation of situation from FIRE to EVACUATE, thus HES has three groundings: HES(P6G1,FIRE,t0), HES(P6G1,FIRE,t1), and HES(P6G1,EVACUATE,t1)
- ^l Parameters observed are: P7G1, SMK_MSHA, and t_1
- ^m P7G1 behaved same as P6G1 and kept impression of FIRE during some time in the beginning of the interval t_1
- ⁿ Parameters observed are: P10G1, SMK_MSHA, and t_1
- ^o P10G1 could not recognize EVACUATE situation and kept the impression of FIRE situation till the end of the scenario

kept walking slowly as if trying to listen to the words. Only one participant ignored PA for FIRE situation. This participant ignored all other cues too. This participant's behaviors were tracked in other scenarios, not reported here, and it is found that he had developed a tendency to move to the lifeboat station, irrespective of any situation. Four other participants were found who only ignored or did not focus on PA related with EVACUATE situation as their gestures showed no change in their pace of their previously selected actions. For instance, all of them were heading towards the messhall when the GPA alarm turned to PAPA and the PA related with PAPA started being announced. But none of them re-routed to show their understanding or vigilance with the new demands in the PA. That is the reason, the authors inferred that these four participants did not put their attention on the PA. So, in all these five cases the predicate HFO was assigned with a Boolean `false` value.

For the predicate FPA, if a person does not focus on PA wordings, no actions according to the PA should be expected unless another cue triggers the same. In all cases, where a participant did not focus on the PA wording, we assigned FPA a false value. Also, there were four cases where the participants showed focus on the PA by pausing their activities, which they were engaged in before the PA announcement began, and then resuming after the PA is over, but they did not act according to the PA wordings. These PAs were related to EVACUATE situation but none of these participants re-routed immediately after listening to the PA. Therefore, we have assigned `false` values to FPA for these cases too with corresponding HFO having `true` values. In the rest of the cases FPA takes a `true` value.

The dataset for all 17 participants of Group 1, each participant having data for two situations, the FIRE and the EVACUATE situation, was collected and split into the training sample, Tr , containing 80% of the data, and the testing/evidence sample, Te , containing the remaining 20%. The testing/evidence data is used here for making inferences from the trained ELDS module.

Simulation results

An agent has been programmed using the concepts proposed in the work. The agent program is made using three technologies: (1) an object-oriented design pattern for the autonomous agent programming language called OO2APL [13], which is available as a Java API, (2) Alchemy 2.0 [1] that supports Markov logic network development, and (3) the Amine platform [30–32] for the design and development of ontology. This section reports the results obtained after executing the agent program, and a comparison is performed between the simulated scenarios, which are the results of the query predicates R , $HITR$, and HES , and the empirical observations. MC-SAT [47] inference algorithm is used for querying the ELDS module. Table 6 reports the simulated results along with the evidence data Te that is used to make inference from the MLN in ELDS module.

Situation # 1A

In this situation, the agent was provided with the same factors that were available when the participant P1G1 was performing the test scenario TH1 during the first half of the

total testing time, i.e., the interval t_0 , in which a GPA alarm begins sounding followed by the relevant PA while the participant was in the cabin. The agent's ELDS module was set with the values of the predicates L , BST , ST , HFO , and FPA as evidence as mentioned in Sit#1A in Table 6. MC-SAT algorithm was executed with queries $?R$, $?HITR$, and $?HES$ (with required arguments) and the probabilities that these predicates are true are found to be 0.87 for recognizing the alarm (i.e., the predicate R), 0.66 for developing the intention to move to $MESSHALL$ during t_0 , and 0.46 for moving to $LIFEBOAT$ station during t_0 (i.e., $HITR(. , LIFEBOAT, t_0)$),

where the parameter values $MESSHALL$ and $LIFEBOAT$ represent the primary and alternate muster locations, respectively. The probabilities for the agent to recognize and be aware of $FIRE$ and $EVACUATE$ emergencies during t_0 are found to be 0.94 and 0.64 respectively. As there are two sets of probabilities for each of the queried predicate, the agent needs to decide which value to use. Algorithm 1 has been implemented to resolve this issue. The parameter α_1 has been set to 0.6, and the value of α_2 has been set to 20% of the value of α_1 . These values were obtained so that the simulated results are found to be as close to the empirical values as possible. The result of Algorithm 1 based on its implementation in Appendix C.1, determines that during t_0 the agent will move to the primary muster station. This result is the same that was observed in the empirical finding where the participant chose to move to the primary muster station (see the value `true` in the last column of row 1A in Table 6) during interval t_0 .

Situation # 1B

The empirical findings during the second half of the testing scenario for participant P1G1 is reported in Sit#1B in Table 6 by using the Boolean (`true` or `false`) values. The numeric parenthesized values are obtained by running the simulation using the agent. The agent was provided with the same evidence that was perceived by the participant P1G1. The evidence formed the collection of Boolean values for the predicates L , BST , ST , HFO , and FPA . Although, P1G1 was able to form the intention of moving to the right muster station, i.e., the $LIFEBOAT$ station during t_1 , despite the fact that P1G1 was not found to focus on listening to the PAPA alarm and following relevant PA. The moment when P1G1 was entering into the $MESSHALL$ during t_0 , the interval t_0 had ended and the PAPA alarm started sounding. The presence of smoke was a visual cue that has a dominance [52] over the other cues like audio signals (such as listening to the PAPA alarm and PA), therefore, we argue that P1G1 could not utilize the PAPA alarm and the relevant PA to come to form the intention of moving to the $LIFEBOAT$ station. The only cue that was used during t_1 was the presence of smoke in the $MESSHALL$. P1G1 made intention to move to the $LIFEBOAT$ station because he found the $MESSHALL$ compromised already. The simulation results for this part of the emergency are given here as under:

Because the rules where $HITR$ is consequent (rule#9, 10 in Table 2) are based on HFO , FPA , L , R , and BST . All of these predicate values were set to `false` because of the inability of P1G1 to perceive the corresponding cues. The probability that $HITR(P1G1, M2=LIFEBOAT, t_1)$ results `true` has been found to be 0.5. This value is inconclusive based on Algorithm 1. While the agent is present in the $MESSHALL$ (due to the decision in Situation 1A as reported in "SITUATION # 1A" section), and smoke was in the $MESSHALL$,

Table 6 The evidence/test data collected as 20% of the empirical observations. Columns with predicate names having a preceding '?' contain simulated results, which are the probabilities these predicates are true given the evidence data

St#	Predicates	L (., GPA, t0)	?R (., GPA, t0)	BST (., GPA, t0)	?HITR (., M1 M2, t0)	ST	HFO (., PAGPA, t0)	FPA (., PAGPA, t0)	?HES (., H1 H2, t0)
1A	True	True (0.87)	True	True (0.66 _{M1} , 0.46 _{M2})	False	True	True	True ^a (0.94 _{H1} , 0.64 _{H2})	
2A	True	True (0.87)	True	True (0.66 _{M1} , 0.43 _{M2})	True ^b	True	True	True (0.96 _{H1} , 0.26 _{H2})	
3A	False	False (0.0)	False	False (0.51 _{M1} , 0.51 _{M2})	True ^c	False	False	False (0.44 _{H1} , 0.24 _{H2})	
St#	Predicates	L (., PAPA, t1)	?R (., PAPA, t1)	BST (., PAPA, t1)	?HITR (., M1 M2, t1)	ST	HFO (., PAPAPA, t1)	FPA (., PAPAPA, t1)	?HES (., H1 H2, t1)
1B	False	False (0.0)	False	False (0.46 _{M1} , 0.50 _{M2})	True ^d	False	False	False (0.01 _{H1} , 0.38 _{H2})	
2B	True	True (0.9)	True	True (0.44 _{M1} , 0.92 _{M2})	False	True	True	True (0.07 _{H1} , 0.96 _{H2})	
3B	True	True (0.9)	True	True (0.43 _{M1} , 0.94 _{M2})	False	True	True	True (0.14 _{H1} , 0.96 _{H2})	

(a) M1 is short for messhall, the primary muster station, and M2 is short for the alternate muster station, the lifeboat station
 (b) True and false are the empirically observed states for the predicates in each column
 (c) A predicate starting with a '?' is the one that has been queried in the ELDS module. The remaining predicates are used as evidence/test data in the query process. ELDS module that comprises MLN is queried by employing the MC-SAT inference algorithm
 (d) The results of querying for R, HITR, and HES are reported as the probability that the predicates are true given the evidence data. These probabilities are reported as values in parenthesis under the respective columns. A probability value with a subscript M1 stands for the result related with the primary muster station, whereas the one having a subscript M2 refers to the probability that the predicate is true involving the alternate muster station
 (e) H1 refers to FIRE emergency, H2 refers to EVACUATE emergency. The probability that a FIRE emergency is occurred is referred to by ρ_{H1} , and the probability that an EVACUATE emergency has occurred is referred to by ρ_{H2} , where $0 \leq \rho \leq 1$
^a The participant kept impression of FIRE during some time in the beginning of the interval t1
^b Observed SMK_VENT during t0
^c Observed SMK_VENT during t0
^d SMK_MSHA, SMK_STAI, SMK_VENT all observed during t1

the agent perceived the smoke, determined its current position (which was MESSHALL), and passed this information in the form of the following CG:

$$[\text{MESSHALL}] - (\text{thme}) \rightarrow [\text{Smoke}] \quad (3)$$

to the OBR-module (reported in “The ontology-based reasoning module” section). A match of the CG in (3) was made with the antecedent of CS-rule#1 because MESSHALL is a subtype of MusterStation, and Smoke is a subtype of Hazard. The inferred consequent that comes from CS-rule#1 is:

$$\begin{aligned} [\text{MESSHALL}] - \\ - (\text{attr}) \rightarrow [\text{Compromised}], \\ - (\text{expr}) \rightarrow [\text{Person}]. \end{aligned} \quad (4)$$

The above CG (4) has further been considered as antecedent of CS-rule #2, and the final inferred output is the following CG:

$$\begin{aligned} [\text{MESSHALL}] - \\ - (\text{attr}) \rightarrow [\text{Compromised}], \\ - (\text{thme}) \rightarrow [\text{Smoke}], \\ - (\text{expr}) \rightarrow [\text{Person}] \leftarrow (\text{agnt}) - [\text{MoveTo}] - \\ - (\text{attr}) \rightarrow [\text{Destination}] - (\text{obj}) \rightarrow [\text{LIFEBOAT}]. \end{aligned} \quad (5)$$

This final CG (5) contains the relevant cues, which are Smoke that was present at the MESSHALL, and the destination to be reached, which is the LIFEBOAT station. The above CS-rule, during the simulation, has been used to form the intention to move to the *destination*: = LIFEBOAT station, and the BDI framework executes the plan associated with *moving to the lifeboat* station, which was the required action when the primary muster station is engulfed in a hazard.

Situation # 2A

The situation reflects a participant, P2G1, in his cabin when the GPA alarm begins sounding. In the next second, the PA announces that there is a FIRE in the galley. The participant clearly listened to the GPA, understood the PA announcement and made an intention to move to the primary muster location. In this situation, as shown in Table 6 (line 2A), P2G1 has perceived all the cues that led all the predicates to true. During simulation, the agent, was provided with the evidence predicates, L, BST, ST, HFO, and FPA, all having the Boolean values true. The ELDS module computed the probability of forming intention to move to the MESSHALL as 0.66. At the same time, the probability of moving to the LIFEBOAT station was found to be 0.43. Algorithm 1 decides the MESSHALL as the destination location during the interval t0 because the probability of HES has been calculated as 0.96 for the FIRE emergency. As the agent knows the plan about what to do in case of FIRE emergency, which is to move to the MESSHALL, the agent performs the action of moving to the MESSHALL.

Situation # 2B

Continuing the situation 2A, during the next half interval of time, i.e., t1, P2G1 received a PAPA alarm with the relevant PA, and perceived correctly all the available cues

corresponding to the predicates as shown in Table 6 (line 2B). The participant decided to move to the LIFEBOAT station during t1. The agent, in simulating the participant P2G1, was given the same values of the predicates as was perceived by the participant, and the ELDS module arrived at the same result by computing the probability of moving to the LIFEBOAT station as 0.96.

Situation # 3A

In this situation, the participant P3G1 did not pay attention to the GPA alarm when it started sounding while the participant was in the cabin. Right from the beginning, P3G1 made an intention to move to the LIFEBOAT station. By watching P3G1's replay video, no rationale could be found that explains why P3G1 did this, except that this behavior was dominant throughout all the scenarios in which P3G1 participated. The repeated use of the same decision irrespective of what a scenario demands may be considered as an example of similarity-matching and frequency bias [52] because all emergency scenarios considered here have similarities in terms of the cues, like smoke, fire, and alarms. Because P3G1 did not use the cues for decision-making, the predicates L, R, BST, HITR, HFO, FPA and HES are assigned the value *false* during t0, as shown in Table 6 (line 3A). The ELDS module (during simulation), correspondingly, resulted in low probabilities that ultimately brought the OBR-module in action. Here, the agent exploits the only available cue, which was the observation that there was smoke coming out of the MESSHALL vent, and therefore, the agent determined that MESSHALL is compromised. The CG: [MESSHALL]-(thme)→ [Smoke] is used to initiate memory-based inference on the OBR-module. This CG is matched with the antecedent of CS-rule#1, which is a more general form in the ontology, and the consequent was generated as:

$$\begin{aligned}
 &[\text{MESSHALL}] - \\
 &\quad - (\text{attr}) \rightarrow [\text{Compromised}], \\
 &\quad - (\text{expr}) \rightarrow [\text{Person}].
 \end{aligned} \tag{6}$$

This result was further matched with other CS-rules. Since the antecedent of CS-rule#2 is matched with the above result, therefore, the final inference is made in the form of the CG in (7).

$$\begin{aligned}
 &[\text{MESSHALL}] - \\
 &\quad - (\text{attr}) \rightarrow [\text{Compromised}], \\
 &\quad - (\text{thme}) \rightarrow [\text{Smoke}], \\
 &\quad - (\text{expr}) \rightarrow [\text{Person}] \leftarrow (\text{agnt}) - [\text{MoveTo}] - \\
 &\quad - (\text{attr}) \rightarrow [\text{destination}] - (\text{obj}) \rightarrow [\text{LIFEBOAT}].
 \end{aligned} \tag{7}$$

which has clear instruction to move to the LIFEBOAT station during t0.

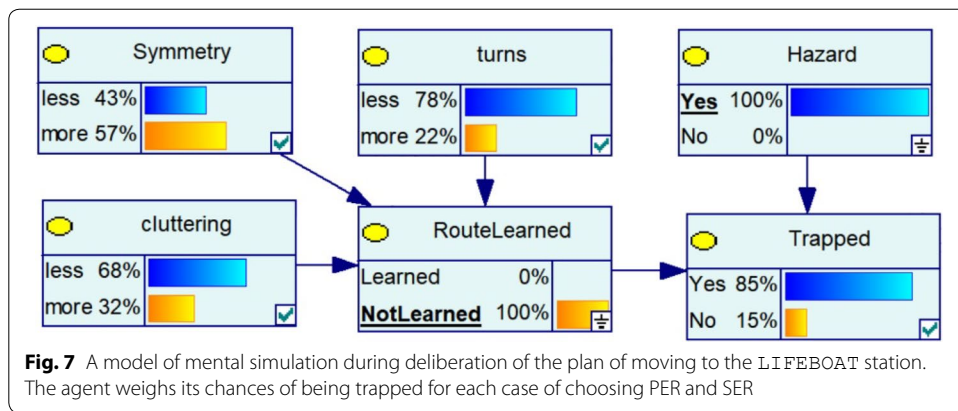
Situation # 3B

The situation 3A turns to 3B when t0 ends and t1 began. At this time, the PAPA alarm began sounding followed by the relevant PA announcement. This happened right after the time when the decision was made as described by the CG (5). Since we have given

the agent all the cues that were observed by the participant P3G1 during t1, using the ELDS module, the agent was able to hold the initial decision of moving to the LIFEBOAT station using the primary egress route. In other words, during situations 3A and 3B, the agent came up with the same decision of moving to the muster station. When the second decision was being made, the plan of the first decision was not yet complete. The BDI framework, as implemented in OO2APL, allows only one plan against a single trigger, therefore, two same decisions of moving to the LIFEBOAT station did not execute two plans, but a single plan corresponding to moving to the LIFEBOAT station was executed. Also, the decision was implemented using a plan that was executed by first setting 'moving to LIFEBOAT' station as a goal, and then fetching a plan that is associated with this goal. During the course of following actions in the plan, the agent kept observing and found smoke in the stairwell. This is a typical situation in which the agent needs to modify the plan by adding or dropping some actions according to the current situation. In a general sense, Klein [33] demonstrates the need to modify actions in a plan by a label "Evaluate Actions-Mental simulation" that follows "Modify" block. A typical plan that performs on-the-fly modification is given in Appendix D, where a plan of moving to a muster station is considered as a goal that is made up of other goals such as *MoveTo*, *TraverseEdge*, *Seek* and *Arrive*, which are the standard steering behaviors [5, 36] used to perform various actions in a plan's execution process. The agent mustered at the LIFEBOAT station. The corresponding probabilities for the queried predicates R, H1TR, and HES have been found to be 0.9, 0.94, and 0.96 respectively (see Table 6, line 3B). In order to show how the process of mental simulation works in accordance with RPDM literature, the agent's beliefbase has been slightly modified by setting the primary escape route (PER) as 'not learned'. The problem of learning by remembering waypoints along a route considering landmarks as opportunities for better retention is considered in [11, 12]. Now what are the consequences, in a hazard, when the agent adapts a route that it does not know? For the present case, the agent exploits a Bayesian network (see Fig. 7) to assess the consequences of choosing PER and the secondary escape route (SER) under current circumstances when a hazard has already been recognized and the agent did not know the primary escape route. The probability of being trapped is found to be higher in choosing PER than that of SER in case PER is not remembered or has not been learned. Therefore, the agent acts on the plan of moving to the LIFEBOAT station using the secondary escape route.

Conclusion

The present work proposes a model that has potential to be used as a realization of Klein's recognition-primed decision model for human decision-making in emergencies. The present work proposes, for the first time, concrete scientific methods that can be used to address the modelling of philosophical modalities of RPDM in a pragmatic setting by also providing a case study as an application. There are two major components of the RPDM that are focused upon here. The first is the SA modelling using experience. This part is modelled in the form of an experiential learning and decision-making module that comprises a Markov logic network L . The network is trained by using empirical data collected by estimating human performance in a VE for different offshore emergency situations involving fire and evacuation. Coupled with the ELDS-module is a



feature matching module that comes into play when the agent’s experience could not recognize a given situation. The feature-matching module is based on an ontology of concepts related with fire and evacuation situations, and this part is the second component of RPDM that is modelled here.

The results show that the model outputs are similar to the decisions made by human participants given the same input cues. Several examples serve to illustrate. In situation 1A the agent recognizes the GPA alarm and has SA about a FIRE situation, forms intention to move to the primary muster station, and initiates a plan to muster at the primary muster station. Situation 1A was the situation the agent had experience about during the training session, so the decision was made because of the agent’s experience. Situation 1B was new to the agent because the agent had no training session in which all cues were absent except a visual of a smoke hazard. The agent exploited the visual cue, that is, smoke in the primary muster station, and used its general knowledge about how to react in case of smoke at a location. Both of the situations 2A and 2B are found to be typical as the agent was able to be aware about the emergency and was able to make decisions as required. In situation 3A, there is a deviation in terms of the reasons behind the decision the agent made, and the decision made by the participant P3G1. P3G1 was found to have used no known cues for his/her intention of moving to the alternate muster station, the LIFEBOAT station. We think that the participant made the choice based on his/her training sessions that show the same trend of moving to the LIFEBOAT station no matter what the circumstances demand. On the contrary, during simulation, when the agent was given the same input cues as was perceived by P3G1, the agent used the only available cue, smoke coming out from the MESSHALL vent, and decided to move to the LIFEBOAT station. In situation 3B, the agent retained the initial decision that it made during interval t0 in situation 3A.

The proposed model performed well on the evidence data (Te dataset) collected. Further work is needed to improve the results. RPDM has many dimensions, such as the use of mental simulation for determining if a certain (already decided) course of actions would work or not. We have simulated a version of this strand of thinking by providing a mechanism right within a plan in the BDI framework that could be used to avoid or mitigate anything wrong that was not expected. For example, one can think that if a wrong choice of a route is made, the repercussions, during an emergency, might be life-threatening. If that is considered as a *violation of expectancies* then the relevant plan

should make sure such a choice would never be made. Appendix D described a pseudocode for a plan used in this study that has a capability to avoid violation of expectancies about the choice of a route after a decision about where to muster has been made. Future work should aim to verify the agent's responses in more complex and demanding environments for which human performance data is available.

Abbreviations and symbols

ACT: the concept type for an act, used in ontology; AI: artificial intelligence; ANIMATE: the concept type for an animate being, used in ontology. An actor of the action; AVERT: all-hands virtual emergency response trainer; BC: Bayesian classifier; BDI: belief-desire-intention (an agent model); BN: Bayesian network; BST: The predicate "Before seeing a threat"; C: the set of ground predicates or facts; CG: conceptual graph; C-RPD: computational-RPD; CS: conceptual structure; ELDS: experiential learning and decision support; EVACUATE: an evacuate situation; FIRE: a fire situation; FOL: first-order logic; FPA: the predicate "Follows public address"; GPA: general platform alarm; Gt: the predicate "Greater"; HES: the predicate "Has emergency situation"; HFO: the predicate "Has focus on"; HIR: the predicate "Has intention to reach"; HSES: the predicate "Has some emergency situation"; IMO: international Maritime Organization; KETA: the predicate "Knows emergency type for an alarm"; KETPA: the predicate "Knows emergency type for aPA"; KETT: the predicate "Knows emergency type for a threat"; KMLA: the predicate "Knows muster location for an alarm"; KMLPA: the predicate "Knows muster location for a PA"; L: The predicate "Listens"; L: The name of the MLN developed in this work; LF: The *Linear Form* of a CG; LIFEBOAT: The secondary or alternative muster location; MC-SAT: Markov Chain SATisfiability algorithm; MESSHALL: The primary muster station; MLN: Markov logic network; MN: Markov network; NDM: naturalistic decision-making; OBR: ontology-based reasoning; OO2APL: Object-oriented 2APL (An agent programming language); OSHA: Occupational Safety and Health Administration (an agency in the United States Department of Labor); P(): a probability function; PA: public address; PAPA: prepare to abandon platform alarm; PER: primary escape route; R: the predicate "Recognizes"; RPD: recognition primed decision; RPDm: recognition primed decision model; SA: situation awareness; SER: secondary escape route; SMK_MSH: constant for smoke in MSH; SMK_STAI: constant for smoke in stairwell; SMK_VENT: constant for smoke coming out from a vent in MSH; SOLAS: safety of life at sea (an international convention); ST: the predicate "Sees threat"; Te: testing data set; Tr: training data set; UML: unified modelling language; VE: virtual environment; Z: the partition function for normalization; ϕ_k : the potential function for the k th clique of the MN comprising G ; a_1 : an arbitrary agent; ag: identifier name for the proposed agent; agnt: a conceptual relation "agent", different from AI's agent. Please see the Definition 6.1; a1: identifier name for an alarm value; attr: the conceptual relation "attribute", see Definition 6.2; charc: the conceptual relation "characteristic", see Definition 6.3; emg_type: identifier name for an emergency type, FIRE or EVACUATE; expr: the conceptual relation "experiencer", see Definition 6.4; G: the graph used in developing the MN; inst: the conceptual relation "instrument", see Definition 6.5; involve: The conceptual relation "involve", see Definition 6.9; k : the running variable used in numbering the cliques of graph G ; LBH or M2: a constant for LIFEBOAT used as parameter in predicates; mloc: identifier name for muster location; MSH or M1: a constant for MESSHALL used as parameter in predicates; $n_i(x)$: the number of true groundings of i th formula; obj: the conceptual relation "object", see Definition 6.6; p_a: identifier name for PA; PAGPA: a constant for PA related to GPA alarm used as parameter in predicates; PAPAPA: a constant for PA related to PAPA alarm used as parameter in predicates; PiG1: where $i \in \{1, 2, \dots, 17\}$. Codes for participants used in the experiment; req: the conceptual relation "require", see Definition 6.8; SIT; i th situation; t: identifier name for time duration; t0: constant for time duration in FIRE situation; t1: constant for time duration in EVACUATE situation; thme: the conceptual relation "theme", see Definition 6.7; thrt: identifier name for a threat or hazard, possible values are SMK_MSH, SMK_STAI and SMK_VENT.

Acknowledgements

The authors would like to thank Dr. Adil Kabbaj of INSEA in Morocco, who is the principal investigator of AminePlatform, Dr. Mehdi Dastani and Dr. Bas Testerink who are the founders of APL language and who provided source files to their newly developed object oriented design pattern based framework for BDI agent modelling, and Professor Mieczyslaw Kokar for his valuable suggestions regarding the use of ontologies.

Authors' contributions

SND performed the development and implementation of the proposed agent model and performed knowledge elicitation from Smith's [56] experiment for validation of the model. JS performed the experiment presented in [56] and verified the data extracted from the experiment. FK supervised the technical development of the proposed agent model. BV supervised the entire study and performed the editorial process. All authors read and approved the final manuscript.

Funding

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)-Husky Energy Industrial Research Chair in Safety at Sea, and the Canada Research Chair Program in Offshore Safety and Risk Engineering.

Availability of data and materials

The training and testing samples used for the validation are available upon request to the authors.

The replay videos used to create the training and testing samples have restricted access because AVERT simulator is not available for public use.

Competing interests

The authors declare that they have no competing interests.

Appendix

Appendix A

A.1 *The recognition-primed decision model*

The RPDM fuses two types of processes. The first is the process that decision makers use to recognize a given situation and come up with a reasonable plan to carry out a course of action. The second is a subtle problem that involves imagining how the course of actions, which having been decided upon, would make sense in the current situation. Will it really make sense? Or does it need modification? These questions are addressed in RPDM at the conceptual level by conditioning that the model output should make sense in situations where a decision-maker does not have time to work on and evaluate all possible logical options by comparing one with others [33]. In RPDM, a decision-maker should pick one option and evaluate if this will work in the current scenario.

There are three variations of RPDM, which should be applied to three distinct types of scenarios a decision-maker may encounter. Interested readers should consult [33, 34] for details about these variations. Figure 1 shows the Klein's RPDM integrated version that combines all features of the three variations of RPDM. In this first variation of RPDM, a decision-maker, due to his/her experience, discovers that the situation is familiar with one that had been solved earlier. The decision-maker is aware of important cues to consider, the expected outcome, plausible goals, and the plan of action. This situation is the most straight forward because the solution to the problem in the given context is already known. Therefore, not much information is needed.

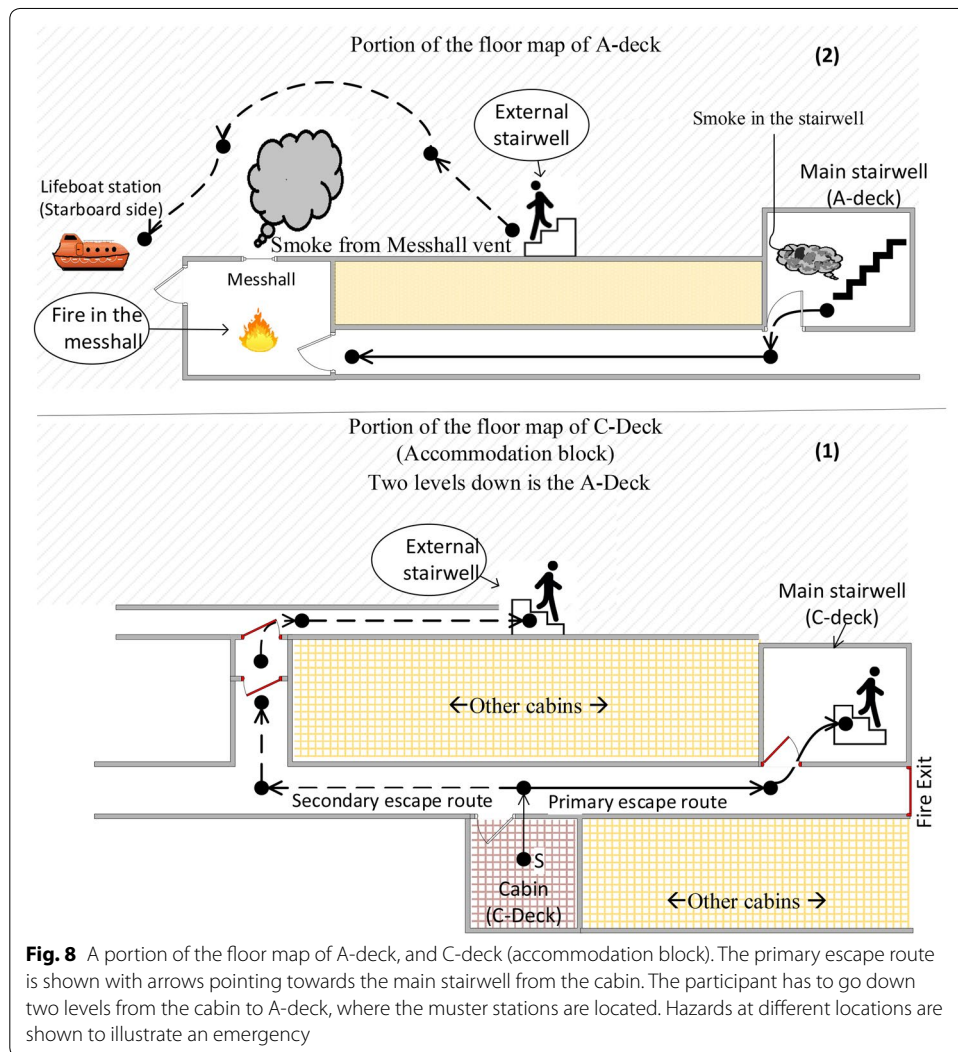
In the second variation of RPDM, a decision-maker comes across a situation that needs more focus on the recognition part. The decision-maker needs more cues to diagnose the nature of the problem to suggest a remedy. The situation may arise due to the reason that the cues do not match clearly with a single typical case or may map onto more than one typical case. This situation makes use of the possibility of making a misinterpretation of some cues until the decision-maker realizes that some expectancies have been violated. The time when the violation of expected outcome is realized, the decision-maker should respond to the anomaly by trying to build a story (by doing mental simulation) to address the discrepancies.

The last variation of RPDM focuses on evaluating a course of actions that have been decided as a solution to the problem in the given situation. The evaluation process exploits mental simulation, which is a process "that weaves together different events into a story that shows how the causes led to the effects" [33], p. 89–90). The purpose of the evaluation process is to make sure the decided course of action would work in a complex situation where there is a doubt about the actions in the plan of handling the situation.

A.2 *AVERT simulator*

This study used a VE called *All-Hands Virtual Emergency Response Trainer* (AVERT). AVERT is a first- person vantage point simulator of an offshore platform that is intended to train workers in emergency egress in offshore platforms. AVERT simulator offers a high-fidelity VE that simulates many things that make up a real offshore facility, such as the control room, the engine room, the steering gear room, stairwells, different sorts of machines, a helipad, muster stations, exit signage, escape ladders, TV-lounge, messhall,

and so on. Each participant was allocated to a cabin and a worksite. There were two muster locations: a primary muster location at the messhall, and a secondary or alternative muster location called the lifeboat station; both are located at the starboard side at A-deck of the vessel. Lifeboats are the primary means of marine evacuation. There were two escape routes from cabins at C-deck to both muster stations (primary or alternative). These routes are called the primary and the secondary escape routes. A schematic of the primary escape route from the cabin to the primary and alternative muster stations is shown in the floor map diagram in Fig. 8.



Appendix B: Assumptions for knowledge elicitation

Because each participant has performed the scenarios as explained in “[Evacuation scenarios and decision tasks](#)” section, the following assumptions seems reasonable about the participants of Smith’s [56] experiment.

1. All participants can recognize the primary and secondary muster stations by seeing these stations. This assumption is based on the fact that each participant has already visited these muster stations many times (at least 4–5 times) before appearing in the session (TH1) used in this study.
2. If a participant recognizes GPA (by listening an audible alarm sound), he/she will know which muster station to move, which is the messhall (primary muster station) in this case. The same is true for the PAPA alarm, which asks to move to the secondary muster station or the lifeboat station.
3. If a participant understands or follow the PA related with the GPA alarm, he knows where to muster and which path to take.
4. If a participant understands or follow the PA related with the PAPA alarm, he knows where to muster and which path to take.
5. All participants know GPA alarm means a FIRE situation, and PAPA alarm means the EVACUATE situation.
6. All participants know that the PA announcement during the GPA alarm is for FIRE emergency and that of during the PAPA alarm is for EVACUATE emergency. The only thing that matters is whether a participant understands the PA or not.
7. All participants know that smoke in the stairwell is caused by a fire in the galley, and that situation is a FIRE situation.
8. All participants know that if smoke comes out of the messhall vent then the messhall is at FIRE. The messhall is, therefore, compromised and this situation is the EVACUATE situation.
9. All participants know that a fire seen somewhere not inside the messhall is a FIRE situation unless the PAPA alarm is ringing.
10. All participants know that a fire or smoke in the messhall means EVACUATE situation.

Appendix C

C.1 An implementation of Algorithm 1 in OO2APL based BDI agent

Let p_1 , p_2 are probabilities for two competing situations (FIRE and EVACUATE) during time interval T .

Let p_3 , is the probability that the agent has developed intention to move to the primary muster station, i.e., the messhall during the time interval T .

Let p_4 , is the probability that the agent has developed intention to move to the secondary muster station, i.e., the lifeboat station.

Assume that the agent knows which muster station to move in case of which emergency type, FIRE or EVACUATE.

```
1.  if(p1>=alpha1&& p2<=alpha2)
2.      exp.arrDecision.add(DecisionalVal.MoveToPMS);
3.  else if(p2>=alpha1&& p1<=alpha2)
4.      exp.arrDecision.add(DecisionalVal.MoveToSMS);
5.  else{
6.      if(p3>=alpha1&& p4<=alpha2)
7.          exp.arrDecision.add(DecisionalVal.MoveToPMS);
8.      else if(p4>=alpha1&& (p3<=alpha2)
9.          exp.arrDecision.add(DecisionalVal.MoveToSMS);
10. else
11.     exp.arrDecision.add(DecisionalVal.Diagnose);
12. }
```

The above code fragment implements a simple conflict resolution scheme in order to decide which situation should be taken as most promising based on the obtained probabilities from MLN inference. If a conflict between two probabilities can not be resolved the trigger *Diagnose* is added and the BDI framework will call *DiagnosePlan* method where ontological reasoning will be used to figure out what actions should be taken in the situation given the available cues.

C.2 The Java class for DiagnosePlan in OO2APL BDI framework

```

public class DiagnoseSituationPlan extends RunOncePlan {
    final Experience currObs;//contains current cues and previous experiences
    public DiagnoseSituationPlan(Experience currObs) {
        this.currObs=currObs;}
    @Override
    public void executeOnce(PlanToAgentInterface planInterface) throws
        PlanExecutionError
    {
        Ontology ontology=(Ontology)planInterface.
            getContext(BeliefBase.class).getOnt();
        Lexicon lexicon= planInterface.getContext(BeliefBase.class).
            getLex();
        System.out.println("Diagnosing Plan");
        for(int i=0;i<currObs.arrCue.size();i++)
        {
            :
            //check if smoke is visible in the messhall
            if(currObs.arrCue.get(i).getST_SMK_MSHA()==true) {
                try {
                    CG cg=CG.parseLF("[PMS]-\n"+
                        "-thme->[Smoke]",lexicon);
                    MemoryDeductiveInference m= new MemoryDeductiveInference(ontology,
                        lexicon);
                    //perform ontology based reasoning using strict //inference algorithm
                    CG cgRslt = m.strictInferenceChain(cg);
                    System.out.println(cgRslt.toString(lexicon));
                }catch(Exception e) {
                    e.printStackTrace();
                }
            }
            :
            //write code for other observed cues
        }
    }
}

```

Appendix D: Actions evaluation and modifications

Let $x [1], x [2], \dots, x[n]$ represent successive waypoints in a selected route that lead to a muster station.

A typical plan for moving to a muster station

```
SelectESCroute //Imagine using BN (see Figure7) what is more likely in each case of
                //selecting PER and SER.
p(trapped) = Calculate probability of being trapped given the current context information and agent's
              belief about PER and SER.
if( p(trapped|PER, hazard) > p(trapped|SER,hazard)
  SetRoute(SER)
Else SetRoute(PER)
MoveTo(destination:= x[n]) //move to x[n] from the current position, a goal
FollowPath //subgoal of MoveTo
  TraverseEdge(x[1]) //x[1] is the first waypoint //Subgoal of FollowPath
  TraverseEdge(x[2])
  :
  TraverseEdge(x[i]) //ith waypoint
  AssessEdgeForBlockage // subgoal of TraverseEdge; an edge is in the navigation graph
  If(route is blocked at x[i])
    Estimate how many waypoints beyond x[i] (including x[i]) be avoided let this be k.
    Find a detour that leads to the x[i+k]th waypoint, let the detour is D.
    SetRoute(D)
    MoveTo(destination:= x[i+k])
  Else
    SetTargetToSteer(x[i])
    If(x[i] is the last waypoint in the current route)
      Arrive(x[i])
    Else
      Seek(x[i])
  :
  TraverseEdge(w[n])
```

Received: 16 June 2019 Accepted: 26 September 2019

Published online: 11 October 2019

References

1. Alchemy (2012) Alchemy: a software for statistical relational learning and probabilistic logic inference based on Markov logic representation. Washington DC
2. Barwise J, Perry J (1983) Situations and attitudes. MIT Press, Cambridge
3. Blakemore S-J, Decety J (2001) From the perception of action to the understanding of intention. *Nat Rev Neurosci* 2(8):561–567. <https://doi.org/10.1038/35086023>
4. Bratman M (1987) Intention, plans, and practical reason. Harvard University Press, Cambridge
5. Buckland M (2004) Programming game AI by example. Jones & Bartlett Learning, Burlington
6. Canellas MC, Feigh KM (2016) Toward simple representative mathematical models of naturalistic decision making through fast-and-frugal heuristics. *J Cogn Eng Decis Making* 10(3):255–267. <https://doi.org/10.1177/1555343416656103>
7. Chase WG, Simon HA (1973) THE MIND'S EYE IN CHESS. In: Visual information processing, pp 215–281. <https://doi.org/10.1016/B978-0-12-170150-5.50011-1>
8. Chowdhury S (2016) Optimization and business improvement: studies in upstream oil and gas industry. Wiley, New Jersey
9. Crandall B, Klein GA, Hoffman RR (2006) Working minds. MIT Press, Cambridge
10. Crowl DA, Louvar JF (2011) Chemical process safety: fundamentals with applications, 3rd edn. Pearson Education Inc., Boston
11. Danial SN, Khan F, Veitch B (2018) A generalized stochastic petri net model of route learning for emergency egress situations. *Eng Appl Artif Intell* 72:170–182
12. Danial SN, Smith J, Khan F, Veitch B (2019) Human-like sequential learning of escape routes for virtual reality agents. *Fire Technol* 55(3):1057–1083. <https://doi.org/10.1007/s10694-019-00819-7>
13. Dastani M, Testerink B (2014) From multi-agent programming to object oriented design patterns. In: Delpiaz F, Dix J, van Riemsdijk MB, eds. Engineering multi-agent systems. EMAS 2014. Lecture Notes in Computer Science, Vol 8758, pp. 204–226. https://doi.org/10.1007/978-3-319-14484-9_11
14. Davies M, Stone T (eds) (1995) Folk psychology: the theory of mind debate. Blackwell Publishers, Oxford,
15. de Groot AD (1965) Thought and choice in chess, 1st edn. Mouton Publishers, The Hague
16. Dennett DC (1987) The Intentional Stance. MIT Press, Cambridge

17. Devlin KJ (1991) Logic and information. Cambridge University Press, Cambridge
18. Domingos P, Lowd D (2009) Markov logic: an interface layer for artificial intelligence. In: Brachman T, Dietterich RJ (eds) Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool Publishers, Seattle
19. Domingos P, Richardson M (2007) Markov logic: a unifying framework for statistical relational learning. In: Getoor B, Taskar L (eds) Introduction to statistical relational learning. MIT Press, Cambridge
20. Endsley M (1988) Design and evaluation for situation awareness enhancement. Proc Hum Fact Ergonom Soc Ann Meeting. <https://doi.org/10.1177/154193128803200221>
21. ExxonMobil (2010) OIMS: System 10-2 Emergency preparedness and response. https://www.cnsopb.ns.ca/sites/default/files/inline/12450_so41877_1_spill_response_soei_0.pdf
22. Fan X, McNeese M, Sun B, Hanratty T, Allender L, Yen J (2010) Human-agent collaboration for time-stressed multi-context decision making. IEEE Trans Syst Man Cybern 40(2):306–320. <https://doi.org/10.1109/TSMCA.2009.2035302>
23. Hadzic M, Wongthongtham P, Dillon T, Chang E (2009) Current issues and the need for ontologies and agents. Ontology-based multi-agent systems. Springer, Berlin, pp 1–14
24. Halpern JY (2003) Reasoning about uncertainty. MIT Press, Cambridge
25. Hassard ST (2009) The variations of recognition primed decision-making and how it informs design decision-making. In: NDM'09 proceedings of the 9th Bi-annual international conference on naturalistic decision making, 57. Swindon: BCS Learning & Development
26. Hu Y, Li R, Zhang Y (2018) Predicting pilot behavior during midair encounters using recognition primed decision model. Inf Sci 422:377–395. <https://doi.org/10.1016/j.ins.2017.09.035>
27. Hutton RJB, Warwick W, Stanard T, McDermott PL, McIlwaine S (2001) Computational model of recognition-primed decisions (RPD): improving realism in computer-generated forces (CGF). Proc Hum Fact Ergonom Soc Ann Meeting 45(26):1833–1837. <https://doi.org/10.1177/154193120104502607>
28. IMO (2009) SOLAS: Consolidated Edition, 5th edn. International Maritime Organization, London
29. Ji Y, Massanari RM, Ager J, Yen J, Miller RE, Ying H (2007) A fuzzy logic-based computational recognition-primed decision model. Inf Sci 177(20):4338–4353. <https://doi.org/10.1016/j.ins.2007.02.026>
30. Kabbaj A (2006) Development of intelligent systems and multi-agents systems with amine platform. In: Hitzler P, Øhrstrøm P (eds) Conceptual structures: inspiration and application. ICCS 2006. Lecture Notes in Computer Science, vol 4068. Springer, Berlin
31. Kabbaj A, Bouzouba K, El Hachimi K, Ourdani N (2006) Ontologies in amine platform: structures and processes. In: Schärfe H, Hitzler P, Øhrstrøm P, eds. Conceptual structures: inspiration and application. ICCS 2006. Lecture Notes in Computer Science, pp 300–313. https://doi.org/10.1007/11787181_22
32. Kabbaj A, Bouzoubaa KM, Soudi A (2005) Amine platform: an artificial intelligence environment for the development of intelligent systems. First Information and Communication Technologies International Symposium ICTIS, Tetuan
33. Klein GA (1998) Sources of power. MIT Press, Cambridge
34. Klein GA (2004) The power of intuition. Doubleday, New York
35. Means B, Salas E, Crandall B, Jacobs TO (1993) Training Decision Makers for the Real World. In: Gary AK, Orasanu J, Calderwood R, Zsombok CE (eds) Decision making in action: models and methods. Ablex Publishing Corporation, Norwood, pp 306–326
36. Millington I, Funge J (2009) Artificial intelligence for games, 2nd edn. Morgan Kaufmann Publishers, Burlington
37. Mueller ST (2009) A Bayesian Recognition decision model. J Cogn Eng Decis Making 3(2):111–130. <https://doi.org/10.1518/155534309X441871>
38. Musharraf M, Smith J, Khan F, Veitch B (2018) Identifying route selection strategies in offshore emergency situations using decision trees. Reliab Eng Syst Saf. <https://doi.org/10.1016/j.ress.2018.06.007>
39. Naderpour M, Lu J, Zhang G (2014) An intelligent situation awareness support system for safety-critical environments. Decis Support Syst 59:325–340. <https://doi.org/10.1016/j.dss.2014.01.004>
40. Norling E (2004) Folk psychology for human modelling: extending the BDI paradigm. In: AAMAS'04 proceedings of the third international conference on autonomous agents and multiagent systems. IEEE Computer Society, Washington DC
41. Norling E, Sonenberg L, Rönnquist R (2000) Enhancing multi-agent based simulation with human-like decision making strategies. In: Moss S, Davidsson P (eds) Multi-agent-based simulation. MABS 2000. Lecture Notes in Computer Science, vol 1979. Springer, Heidelberg
42. Nowroozi A, Shirri ME, Aslanian A, Lucas C (2012) A general computational recognition primed decision model with multi-agent rescue simulation benchmark. Inf Sci 187:52–71. <https://doi.org/10.1016/j.ins.2011.09.039>
43. Orasanu J, Connolly T (1993) The reinvention of decision making. In: Klein G, Orasanu J, Calderwood R, Zsombok CE (eds) Decision Making in action: models and methods. Ablex Publishing Corporation, Norwood, pp 3–20
44. OSHA (2018) Emergency exit routes factsheet. https://www.osha.gov/OshDoc/data_General_Facts/emergency-exit-routes-factsheet.pdf. Accessed 22 Apr 2019
45. Patterson R, Fournier L, Pierce B, Winterbottom M, Tripp L (2009) Modeling the dynamics of recognition-primed decision making. In: Proceedings of NDM9, the 9th international conference on naturalistic decision making. British Computer Society, London
46. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inferences. Morgan Kaufmann, San Mateo
47. Poon H, Domingos P (2006) Sound and efficient inference with probabilistic and deterministic dependencies. In: Proceedings of the 21st national conference on artificial intelligence, Volume 1, pp 458–463. <http://dl.acm.org/citation.cfm?id=1597538.1597612>
48. Press G (2018) The Brute Force of IBM Deep Blue and Google DeepMind. <https://www.forbes.com/sites/gilpress/2018/02/07/the-brute-force-of-deep-blue-and-deep-learning/#3729cb5c49e3>
49. Proulx G (2007) Response to fire alarms. Fire Protect Eng 33: 8–14. http://www.cfaa.ca/Files/flash/CODES/FIREA_LARMARTICLESFORTHEAHJ/FireAlarmResponse.pdf

50. Raedt L De, Kersting K, Natarajan S, Poole D (2016) Statistical relational artificial intelligence: logic, probability, and computation. In: Synthesis Lectures on artificial intelligence and machine learning, Vol 10. <https://doi.org/10.2200/S00692ED1V01Y201601AIM032>
51. Rao AS, Georgeff MP (1995) BDI agents: from theory to practice. In: Lesser V, Gasser L, eds. Proceedings of the first international conference on multi-agent systems (ICMAS-95), pp 312–319
52. Reason J (1990) Human error. Cambridge University Press, Cambridge
53. Resnick M (2001) Recognition primed decision making in E-commerce. In: Proceedings of the human factors and ergonomics society annual meeting. Sage Journals Ltd, Santa Monica
54. Shanton K, Goldman A (2010) Simulation theory. *Wiley Interdiscip Rev* 1(4):527–538. <https://doi.org/10.1002/wcs.33>
55. Singla P, Domingos P (2005) Discriminative training of markov logic networks. In: Proceedings of the 20th national conference on artificial intelligence, vol 2, pp 868–873. <http://dl.acm.org/citation.cfm?id=1619410.1619472>
56. Smith J (2015) The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios. Memorial University of Newfoundland, St. John's
57. Sneddon A, Mearns K, Flin R (2013) Stress, fatigue, situation awareness and safety in offshore drilling crews. *Saf Sci* 56:80–88. <https://doi.org/10.1016/j.ssci.2012.05.027>
58. Sokolowski JA (2003) Enhanced decision modeling using multiagent system simulation. *Simulation* 79(4):232–242. <https://doi.org/10.1177/0037549703038886>
59. Sowa JF (1984) Conceptual structures: information processing in mind and machine. Addison-Wesley, Reading
60. Sowa JF (2000) Knowledge representation: logical, philosophical and computational foundations. Brooks/Cole Thomson Learning, Pacific Grove
61. Spouge J (1999) A guide to quantitative risk assessment for offshore installations. CMPT Publication, Aberdeen
62. Thilakarathne DJ (2015) Modelling of situation awareness with perception, attention, and prior and retrospective awareness. *Biol Inspired Cogn Arch* 12:77–104. <https://doi.org/10.1016/j.bica.2015.04.010>
63. Tong D, Canter D (1985) The decision to evacuate: a study of the motivations which contribute to evacuation in the event of fire. *Fire Saf J* 9(3):257–265. [https://doi.org/10.1016/0379-7112\(85\)90036-0](https://doi.org/10.1016/0379-7112(85)90036-0)
64. Tutolo D (1979) Attention: necessary aspect of listening. *Language Arts* 56(1): 34–37. <http://www.jstor.org/stable/41404756>
65. Wankhede A (2017) Different types of alarms on ships. *Marine Insight*. <https://www.marineinsight.com/marine-safety/different-types-of-alarms-on-ship/>
66. Winerman L (2004) Fighting fire with psychology. *Mon Psychol* 35(8): 28. <https://www.apa.org/monitor/sep04/fighting>
67. Zsombok CE (1997) Naturalistic decision making: where are we now? In: Zsombok CE, Klein G (eds) Naturalistic decision making. Lawrence Erlbaum Associates, Publishers, Mahwah, pp 3–16
68. Zsombok CE, Klein GA (1997) Naturalistic decision making. Lawrence Erlbaum Associates, Publishers, Mahwah

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.