Software agents & human behavior

by

© Syed Nasir Danial

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

Ph.D. in Computer Engineering, Faculty of Engineering and Applied Science

Memorial University of Newfoundland

October 2019

St. John’s, Newfoundland and Labrador, Canada

Abstract

People make important decisions in emergencies. Often these decisions involve high stakes in terms of lives and property. Bhopal disaster (1984), Piper Alpha disaster (1988), Montara blowout (2009), and explosion on Deepwater Horizon (2010) are a few examples among many industrial incidents. In these incidents, those who were in-charge took critical decisions under various ental stressors such as time, fatigue, and panic. This thesis presents an application of naturalistic decision-making (NDM), which is a recent decision-making theory inspired by experts making decisions in real emergencies.

This study develops an intelligent agent model that can be programed to make human-like decisions in emergencies. The agent model has three major components: (1) A spatial learning module, which the agent uses to learn escape routes that are designated routes in a facility for emergency evacuation, (2) a situation recognition module, which is used to recognize or distinguish among evolving emergency situations, and (3) a decision-support module, which exploits modules in (1) and (2), and implements an NDM based decision-logic for producing human-like decisions in emergencies.

The spatial learning module comprises a generalized stochastic Petri net-based model of spatial learning. The model classifies routes into five classes based on landmarks, which are objects with salient spatial features. These classes deal with the question of how difficult a landmark turns out to be when an agent observes it the first time during a route traversal. An extension to the spatial learning model is also proposed where

the question of how successive route traversals may impact retention of a route in the agent's memory is investigated.

The situation awareness module uses Markov logic network (MLN) to define different offshore emergency situations using First-order Logic (FOL) rules. The purpose of this module is to give the agent the necessary experience of dealing with emergencies. The potential of this module lies in the fact that different training samples can be used to produce agents having different experience or capability to deal with an emergency situation. To demonstrate this fact, two agents were developed and trained using two different sets of empirical observations. The two are found to be different in recognizing the prepare-to-abandon-platform alarm (`PAPA`), and similar to each other in recognition of an emergency using other cues.

Finally, the decision-support module is proposed as a union of spatial-learning module, situation awareness module, and NDM based decision-logic. The NDM-based decision-logic is inspired by Klein's (1998) recognition primed decision-making (RPDM) model. The agent's attitudes related to decision-making as per the RPDM are represented in the form of belief, desire, and intention (BDI). The decision-logic involves recognition of situations based on experience (as proposed in situation-recognition module), and recognition of situations based on classification, where ontological classification is used to guide the agent in cases where the agent's experience about confronting a situation is inadequate. At the planning stage, the decision-logic exploits the agent's spatial knowledge (as proposed in spatial-learning module) about the layout of the environment to make adjustments in the course of actions relevant to a decision that has already been made as a by-product of situation recognition.

The proposed agent model has potential to be used to improve virtual training environment's fidelity by adding agents that exhibit human-like intelligence in performing tasks related to emergency evacuation. Notwithstanding, the potential to exploit the basis provided here, in the form of an agent representing human fallibility, should not be ignored for fields like human reliability analysis.

Dedication

To the memories of my father, Syed Noor-ul-Haq (late), and my mother for countless efforts they put in to carve me …

Acknowledgments

In the name of Allah (the God), the Most Beneficent, the Most Merciful. All praise be to Allah (the God) alone, the Sustainer of all the worlds, most Compassionate, ever Merciful, and I send salutations upon His noble prophet Muhammad peace be upon him.

I could not have accomplished this work as presented here had it not been for the expert support and constant encouragement from my supervisors Dr. Brian Veitch and Dr. Faisal Khan. Their enormous research experience and knowledge have brought me to think seriously on the problem I have taken up in this thesis. Also, their diverse interests helped me think over a problem with different perspectives. I would always remember the way they guided me through all this process. It has been a pleasure working under their supervision.

I would also like to thank all of my fellow colleagues at Safety @ Sea group at Memorial University of Newfoundland, for their valuable help and suggestions for improving this work. Thanks to the NSERC-Husky Energy IRC in Safety at Sea, for their financial support. Very especial thanks to Jennifer Smith for sharing her experimental work.

In the last, I would like to extend my deepest and sincere thanks to my elders, Muhammad Hanif Shah, my late father — Syed Noor-ul-Haq, my mother, Seema and other sisters, and Shariq bhai and my other brothers for their moral support, and thousands of prayers. I know without their love, conviction and prayers I would never be able to perform this scholarly activity.

I would like to thank my wife Bushra for her constant support, love, and care, for all the late nights and early wakeups to keep me calm over the course of this study. I thank you Bushra for being my sounding board. I owe you everything. In the last, I would like to thank my children Ahmed and Yaseen for their rhythmic talks that used to soothe me after long hours of hard working.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms & Notations

ACO	Ant Colony Optimization.
ACT	The concept type for an act, used in ontology.
ADC-IDAC	Accident Dynamic Simulator-Information, Decision, and Action in a Crew.
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.
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.
CPN	Colored Petri-Net.
C-RPD	Computational-RPD.
CS	Conceptual Structure.
CTMC	Continuous time Markov Chain.
D1	The dataset from Group 1 participants.
D2	The dataset from Group 2 participants.
DBN	Dynamic BN.
DPG	Discretized path graph.
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.
GSPN	Generalized stochastic Petri-Net.
GSPNRL	The GSPN based model of Route Learning.
`Gt`	The predicate "Greater".
H_2S	Hydrogen sulphide.
HAC-ER	Human Agent Collectives-Emergency Response.

HES	The predicate "Has emergency situation".
HFO	The predicate "Has focus on".
HITR	The predicate "Has intention to reach".
HRA	Human Reliability Analysis
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 a PA".
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".
LA	Learning Alarms scenario.
LE	Learning scenario.
LF	The *Linear Form* of a CG.
LH	Learning hazard scenario.
LIFEBOAT	The secondary or alternative muster location.
MCMC	Markov chain Monte Carlo algorithm.
MC-SAT	Markov Chain SATisfiability algorithm.
MESSHALL	The primary muster station.
MLN	Markov logic network.
MN	Markov network.
NC	Navigation command.
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).
OWL	Web Ontology Language.
P(.)	A probability function.
PA	Public address.
PAPA	Prepare to abandon platform alarm.
PER	Primary Escape Route.
PN	Place-Transition Net.

R	The predicate "Recognizes".
ROI	Region of Interest.
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).
SPN	Stochastic Petri-Net.
ST	The predicate "Sees threat".
STO	Situation theory ontology.
TA	Training/practice Alarm scenario.
Te	Testing data set.
TE	Testing scenario.
TH	Training hazard scenario.
Tr	Training data set.
UML	Unified Modelling Language.
VE	Virtual environment.
Z	The partition function for normalization.

Notations

α	Transition firing rate in a Petri-Net.
τ	Identifier name for time duration.
γ	The difficulty level of a landmark.
Λ	The set of transition rates.
θ_k	The rate of a transition, where k is some integer subscript.
Υ	The subset of S such that sites are occupied; for $S - \Upsilon$ the sites are not occupied.
π	Steady state distribution vector of a CTMC.
$\tau 0$	Constant for time duration in the context of FIRE situation.
$\tau 1$	Constant for time duration in EVACUATE situation.

λ_k	The rate of *k*th transition in a Petri-Net model.
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	Conceptual relation "agent", not agent in AI, see definition 6.1.
A_k	The *k*th place in the GSPNRL model.
al	Identifier name for an alarm value.
attr	The conceptual relation "attribute", see definition 6.2.
chrc	The conceptual relation "characteristic", see definition 6.3.
$\mathfrak{D}$	A domain for the type of things in the context of an ontology.
E	The set of edges for the graph *G*.
$e(p_i)$	The average % of wrong decisions per decision point.
emg_type	Identifier name for an emergency type, FIRE or EVACUATE.
expr	The conceptual relation "experiencer", see definition 6.4.
$f_X(.)$	Sojourn time density function.
$F(x)$	The distribution function of firing time.
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*.
$\mathcal{L}$	A language people use to talk about things in the domain $\mathfrak{D}$.
L	The set of landmarks.
LBH or M2	A constant for LIFEBOAT used as parameter in predicates.
M_X	The X*th* marking of CTMC, or the process is in state X.
M_0	Initial marking of a Petri-Net.
mloc	Identifier name for muster location.
MSH or M1	A constant for MESSHALL used as parameter in predicates.
N1	The trainer model in the proposed GSPNRL model.
N2	The action generator and learning model in GSPNRL.
N3	The route model in GSPNRL.
$n_i(x)$	The number of true groundings of *i*th formula.
NUM	A datatype for some places in the GSPNRL.
NUMLEVELS	A datatype for some places in the GSPNRL.
obj	The conceptual relation "object", see definition 6.6.

P	A finite and non-empty set of places in a Petri-Net.
$\wp$	The power set notation.
`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.
p_k	The *k*th decision-point on the route R1.
P*i*G1	Where $i \in \{1,2, \ldots, 17\}$. Codes for participants used in the experiment.
Q	A transition rate matrix, or infinitesimal generator.
Qr	The set of questions or queries.
R1	The primary escape route in AVERT simulations.
`req`	The conceptual relation "require", see definition 6.8.
Rj	The set of rules containing j number of rules, where $j > 0$.
S	The set of sites containing objects or empty [objects could be formulas representing situations].
`send`, `ack`	Places to maintain asynchronous communication between N1 and N2 in the GSPNRL model.
$\mathtt{SIT}_i$	The *i*th situation.
T	A finite, nonempty set of transitions in a Petri-Net.
`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`.
$\mathtt{t}_k$	The *k*th transition in a Petri-Net model.
$\mathtt{Tr}_k$	The *k*th place in N1.
U	A variable name for the MLN.
w_i	The weight for an *i*th element.
W_k	The set of k weights, $w_1, w_2, \ldots, w_k$.

Operators

I^-, I^+	Backward and forward incidence operators, respectively.
$\neg$	Logical negation
$\Rightarrow$	Logical implication

List of Appendices

Chapter 1

Introduction, overview, and co-authorship statement

1.1 Problem statement

With the advent of computers in c.1950, the world is now witnessing an ever-increasing pace in the technological advancement and expansion in industries whose products range from simple pencils to sophisticated fighter jets that can perform powerful and computer assisted maneuvering during typical flight operations. Some machines like smartphones would like to talk as well! There are lucid and varying opinions among scientists as to which algorithms or machines should be called intelligent and which should not, what reasoning is legitimate or correct for a smart behavior to fulfill the requirements for the status of an intelligent being, and what reasoning lacks this status. One school of thought considers 'conscious thought process' as a fundamental agency of the human mind that has no parallel in terms of representing it as an algorithm, and, therefore, at least some conscious thinking must be non-computational, and hence cannot be represented by a Turing machine (Lucas, 1961; Penrose, 1989, 1991). Words like *soul* (Al-Ghazālī, 1998; Alanen, 2014 p. 88; Descartes, 2015; S. A. Khan, 1989) or *mind* (Penrose, 1989) have been used to better explain this eccentricity in some attitudes, because these attitudes, such as consciousness, intention, beliefs and much like, do not seem to fit in a metric space. The discussion on the well-known mind-body problem is one example where eminent philosophers, such as Aristotle (384-322 BC), had a view that mind has no physical form (Aristotle, 2001), otherwise it would have been affected in a limited way (by physical objects) just as an eye, which is affected by only photons of light and not by the sound waves. The other school of thought contends this theory and argues that it is possible to make machines that can think, not in the sense of by consuming masses of data in the name of learning[1] (Hassler, 2016), as is the case of deep learning, but in the way a real solution to machine-intelligence problem makes sense (Minsky, 1988, 1991).

The present thesis does not go into issues concerning the theological or ideological interpretations of how intelligence should be defined. The proposed approach takes a practical way by aiming to develop models (of mechanisms as they are understood today) for some mental attitudes that could be used by agent programs to make them behave *human-like* in some acceptable sense in a limited scope. This line of research is favored in several works related to human factors engineering. As an example, consider Reason's (1990) model of a fallible machine that is based on the assumption that human error has origin in certain mental processes that generate it inadvertently.

In earlier days, computers would perform tasks that were tedious due to lengthy calculations. Now machines are needed so as to assist humans in their complex multitude of jobs — requiring a significant cognitive workload — starting from medical diagnostics to computer games, virtual reality, crime investigation, assisting jurisprudence, accident investigation, training using high fidelity simulators and virtual environments, commanding a firefighting team, piloting a fighter airplane, and what not. Machines, which can assist people in diagnosing a patient's disease, or

---

[1] see, for example, how the computer program, AlphaGo, was trained for playing the game *Go* (Borowiec & Lien, 2016).

helping trainees to learn better responses, should fulfill the requirements of artificial intelligence (AI), but above all there must exist a relationship of trust between the machine and the human participants involved so that the participants can show faith in the machine's output. A foundational test for machine intelligence is termed as the Turing test (Turing, 1950), which says if a person cannot differentiate between a machine's response and a human's response during an interaction, then the machine should be given a fair credit of being the one showing some intelligence. This study argues that the conventional AI techniques lack trust in the end users' eyes because the way computations are performed is superficial or extraneous in that the machine — or the algorithm involved — does not seem to take into account the human mind's psyche. Also, the machine does not incorporate the human problem-solving techniques to the extent a human does, for instance, a medical practitioner exploits his problem-solving skills to conclude a patient's critically deteriorating health, or a fire commander comes to decide a course of action from a plethora of experiences he/she has had at his/her disposal.

The main goal of this thesis is to introduce methods or models for agents, targeting mental agencies of learning, situation-awareness, and decision-making, where the computation is performed in the way it is thought to be performed by human beings. For simplicity, the present thesis focuses on route-learning instead of general learning. Models of route-learning, situation awareness, and decision-making are presented in the context of offshore emergencies. In other words, a coherent picture of the agent modeling, as proposed here, contains models for different but related capabilities for an agent. The agent will be able to learn routes in the same fashion a human being would learn — some locations will be easy, and some will be difficult for learning.

The agent will be able to develop an understanding of an emergency by classifying which emergency is of which type. Finally, the agent will be able to make decisions based on its experience as to what actions are needed to respond to the current emergency.

1.2 Research objectives

The present work focuses on developing an agent model that produces human-like behavior when subjected to a similar context in which a person interacts. The following are the research objectives:

(a) To develop an agent model that simulates how much of a route will be remembered after the agent is exposed to a route for the first time. The model should simulate human-like behavior of remembering a route when a person traversed it the first time.

(b) To develop agents with different skills of remembering parts of a route.

(c) To develop agents with different route knowledge in an environment.

(d) To develop agents with different experiences of recognizing fire and evacuation emergencies.

(e) To develop agents with different ability to make decisions as to what needs to be done in an emergency situation in a similar way people make decisions.

1.3 Overview of agent modeling

Agent field is attributed to John von Neumann's work on cellular automata and game theory[2] (Dyson, 2012; Shoham & Leyton-Brown, 2009). The notion of agent seems like a newborn in computing ever since its inception in the field perhaps nearly half a century ago. Surely, it is after World War II and the advent of computers and AI that anything like autonomous agents began to come to light. According to Alan Kay (1984), the idea of agent goes with John McCarthy (1927-2011) in the mid-1950s, and the term was coined by Oliver G. Selfridge[3] (1926-2008) a few years later.

In contemporary times, agent modeling is considered as an area of investigation where a system is designed in a way that it can solve problems, while at the same time contains basic and relevant capabilities, such as autonomous behavior and intelligence. This strand of agent research is motivated through a study by Nwana (1996). In such a modeling work, the focus remains on the individual's attributes, properties, and functionalities so that their collective embodiment may have features essential to define an agent. On the other hand, the term agent-based modeling has a history of being used in disciplines where instead of individual behaviors, a coarse-grained or global behavior is the main focus of study. Such a global view that is generated by interactions of individual agents is important in a number of areas, including game theory, complex adaptive systems, nonlinear systems, complexity,

[2] See *The Theory of Games and Economic Behavior by* John von Neumann & Oskar Morgenstern (1944).
[3] See Selfridge (1988).

cybernetics (Heath & Hill, 2008), sociology (D. Klein, Marx, & Fischcach, 2018), and perhaps to some extent statistical mechanics.

Nonetheless, the importance of an individual agent cannot be undermined because of a few interesting concerns. The first concern is a rather theoretical debate on the number of and nature of minimal attributes that are required to define an agent (Minsky, 1988, 1991; Penrose, 1989; Wooldridge, 2009). The second concern is related to the tools that can best represent the attributes or agencies of an agent. It turns out that there are many models, with support for representing different agencies, for an individual agent. An interesting model for the present study is the Belief-Desire-Intention agent model (Rao & Georgeff, 1995) that is based on Bratman's theory of *practical reasoning* (Bratman, 1987). A BDI agent (see Figure 1.1) has beliefs about the world it is acting in. The *desires* form a set of motivational factors or knowledge for the agent. An agent might be developed that uses certain strategies

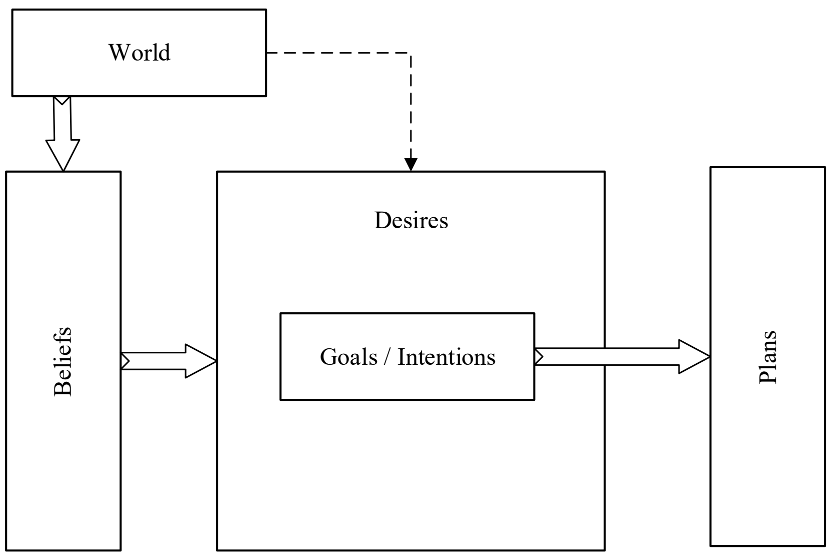


Figure 1.1. A general design of the components in a typical BDI agent architecture.

during interaction with other agents (in competition or cooperation) that maximize the wellbeing of the interacting agents (Parsons & Wooldridge, 2002). Such a strategy or policy is an expression of desire the agent will have. Many practical implementations of the BDI architecture use *goals* to mean *desire* as well as a target that the agent wants to achieve. Intention, on the other hand, is a goal that the agent is committed to deliberate. As shown in Figure 1.1, a BDI agent receives inputs from the environment to stay current. Based on current percepts, the agent selects a goal — in BDI logic, this goal is called intention — and decides a course of actions. The course of actions related to an intention is called a *plan* (Rao & Georgeff, 1995). BDI agents are rational agents, which means they are designed to select the best course of action against an intention.

At last, is a concern pointed out in (Kay, 1984) as, "… current artificial intelligence techniques contain the seeds of architecture from which one might construct some kind of mentality that is genuinely able to learn competence", where mentality and competence refer to that of human beings. Contrary to Kay's belief about AI in 1984, the author thinks that the seeds to construct human-like behavior does not contain only the genes from AI, but contributions of other fields cannot be neglected. Cognitive Psychology and Human Factors have played an important role in our understanding of the human thought process.

1.3.1 What is an agent?

As the notion of agent is used in many disciplines, from computer science to economics (Ross, 1973), the meaning of the notion remains vague until a specific account of agenthood is manifested. In a broader AI sense, an agent is an entity that is situated in an environment, and that has the ability to act autonomously in order to meet its delegated objectives (Wooldridge, 2009). Perhaps being autonomous is the unanimously accepted criteria of agency for an agent.

There are different views among scientists for a definition of the notion of agent. A few of these are listed below:

1. Wooldridge & Jennings (1995) define the notion by using two separate titles, the *weak notion of agency* and *the strong notion of agency*. These forms are defined as:
   (a) The Weak Notion of Agency is a general way to use the term 'agent' to mean a hardware or software that should possess the following properties:
      - Autonomy: Agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
      - Social ability: Agents interact with other agents (and possibly humans) via some kind of agent-communication language;
      - Reactivity: Agents perceive their environment and respond in a timely fashion to changes that occur in it;
      - Pro-activeness: Agents do not simply act in response to their environment; they are able to exhibit goal-directed behavior by taking the initiative.

   (b) The Stronger Notion of Agency: According to this, the notion of 'agent', in addition to having the properties associated with the weak notion of agency, is either conceptualized or implemented using concepts that are

more usually applied to humans. For example, it is quite common in AI to characterize an agent using mentalistic notions, such as knowledge, belief, intention, and obligation (Shoham, 1993). Some AI researchers have gone further, and considered emotional agents.

2. The KidSim Agent (D. C. Smith, Cypher, & Spohrer, 1994) is defined as a persistent software entity dedicated to a specific purpose.
3. Russel & Subramanian (1995) define the notion of agent as, “An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors”.
4. Shoham (1993) argues that “An agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments. These components are defined in a precise fashion, and stand in rough correspondence to their commonsense counterparts. In this view, therefore, agenthood is in the mind of the programmer: What makes any hardware or software component an agent is precisely the fact that one has chosen to analyze and control it in these mental terms.”

   Put differently, Shoham is of a belief that any entity, be it a computer program, or hardware, can be called an agent as long as it has been ascribed mental qualities. It will, however, remain a question for investigation as to whether such an ascription of mental attitudes is meaningful in a given context (McCarthy, 1979).

5. “Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so, realize a set of goals or tasks for which they are designed.” (Maes, 1995).

6. “Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions” (Hayes-Roth, 1995).

The attributes of agency are what makes an entity, software, or hardware, an agent. Thus, the weak notion of agency limits the attributes to four: autonomy, social ability, reactivity, and pro-activeness. In the stronger notion of agency, several mental qualities such as beliefs, desires, or intentions, are required to be ascribed to the agent, and this ascription is considered legitimate when it expresses the same information about the machine that it expresses about a person (McCarthy, 1979).

1.3.2 Intelligence, Fallibility and agents

Intelligence is attributed as a property of fallible beings. Arguably, this is because fallible beings are endowed with the brilliance to prefer something over something else that often is based on a good reason; and for that matter, there are infrequent miscalculations (Turing, 1947).

Agents are now a part of many types of decision-support systems. Computer games, virtual-reality, evacuation simulation, complex adaptive systems, and planning such as generating a sequencing plan and conflict-free trajectories for a set of aircraft attempting to land at a given airport (Man, 2015), are but a few examples. The kind of agency that is addressed here must show coherence with the notion of fallibility, which is a fundamental characteristic of human beings, in order that it could be regarded as a sort of being showing *intelligence*. The present work aims at developing a model of a fallible machine, called an artificially intelligent agent, or simply an agent, that in some way is inspired by corresponding human mental attitudes.

Fallibility should not be regarded as an outcome arising on the surface by a failing component of the involved computing machinery, but it reflects an effort that is unsuccessful this time, and that must be remembered in all future attempts. Reason (1990) says that fallibility is inherent within the methods that are responsible to perform the job for which the method was employed. So, the process of modelling fallibility in mental attitudes, is not straight-forward. The target is to achieve agent behavior that can be considered ‘similar’ to human response as long as the context remains the same.

1.4 Mental qualities of an agent in emergencies

Harman (1976) says that reasoning is distinguished from logic in that it is a process that modifies premises about concluding something by adding or deleting some conditions in the antecedents. In this respect, reasoning has no premises and it does not conclude anything. The theory of reasoning is important in decision-making because when people deliberate, they make intentions to achieve some goal. Beliefs and desires interplay in making of intentions in a given context (Bratman, 1987). The account on beliefs, intentions, and then decision-making in traditional approaches to decision-making are much suited for optimal or algorithmic style selections among different candidate choices in well-structured or laboratory settings. Even if the algorithmic style strategies follow the theory of probability, the principals of expected utility theory, and the Bayesian formalism, people are found to rather rely on certain heuristics and biases in making decisions in real life (Klein, 2008). For example, in their (Tversky & Kahneman, 1974) proposal to three biases[4] that invalidate or overturn people's ability to depend on optimal selection, in case they have it, the authors explain if people are asked to evaluate probability that a person belongs to a class of, say, engineers or lawyers, and a brief personality description is shown to them, then people will only use prior probabilities of such events if the personality description has not been shown before asking. However, if people are exposed to even a brief personality description that will act as a temptation, and they will use the features in the personality description to match those they already know about engineers and lawyers and come up with an estimated probability based on representativeness or similarity.

In short, there are pitfalls pointed out in (Kahneman, Slovic, & Tversky, 1982) that decision-making, as real people do, is not based on the standards set by the laws of probability theory, it is rather based on simple heuristics and biases that render no need for people to compare available options with one another (Klein, 1998, 2008). This is especially true for real situations that typically are instances where (*i*) the decision problem is ill-structured (unlike artificial or well-structured problems), (*ii*) there is a time stress (as opposed to ample time), (*iii*) the environment is dynamic (as opposed to static or laboratory environments), (*iv*) the goals are uncertain or competing (as opposed to clear or well-known goals), (*v*) the decision maker has to keep an eye on the chosen actions for possible re-assessment (unlike one-shot actions), and (*vi*) there is an involvement of high-stakes in terms of life and/or

[4] These biases are: (*i*) the representativeness bias, (*ii*) the availability bias, and (*iii*) the adjustment and anchoring bias.

property (as opposed to situations where a wrong decision would not have devastating consequences) (Zsambok, 1997).

This new approach to decision-making is termed as *Naturalistic Decision Making*. Basic to an NDM-based decision-making approach are three important capabilities:

1. *Learning*: An agent should possess knowledge about important things in the environment for a problem that it is to be decided upon.
2. *Situation awareness*: An agent should be capable of noting the changing environment to characterize a new situation or recognize it as the one it has already seen in the past.
3. *Decision-making*: An agent should be able to make plans based on intentions formed due to current recognition of a situation and re-assess its plans for any modification as a result of changes in the environment or as a result of failing expectancies.

The present work considers these capabilities in a particular context so that the resulting models can easily be worked out and at the same time, they should have enough generality that they can be applied or used in other contexts. The context that is considered here is the offshore emergency environment. As an example of how the models proposed here would form the agent's overall behavior when an event happens, consider an example in which an agent is monitoring an airplane's flight operations during a typical flight. At some moment, an alarm for proximity warning starts sounding. The agent also receives a verbal message about the situation that the airplane is getting close to the ground. If the agent knows what to do in this situation. The agent tries to get the airplane altitude or position with other instruments, and if possible, visually. With all this information, the agent needs to build "an intention" for doing a course of action. In Object-Oriented-2APL (OO2APL[5]) (Dastani & Testerink, 2014), which is a BDI agent platform, this intention is generated in the form of a trigger that calls a plan associated with the trigger. Figure 1.2 depicts how the information gathering, to situation-awareness, and finally to decision-making stages should proceed. The only difference in Figure 1.2's depiction and the way standard BDI logic works is in the estimation of the probability that the alarm is not a false alarm. This inclusion of probability is based on the approach presented in this thesis as explained in the following chapters.

The capability in (1), in the preceding paragraph, asks to model learning, which is taken to be *spatial learning* because this type of learning is directly involved in emergencies in that the workers are given training for egress in the event of an emergency by exploiting dedicated escape routes. The capability in (2), is taken here

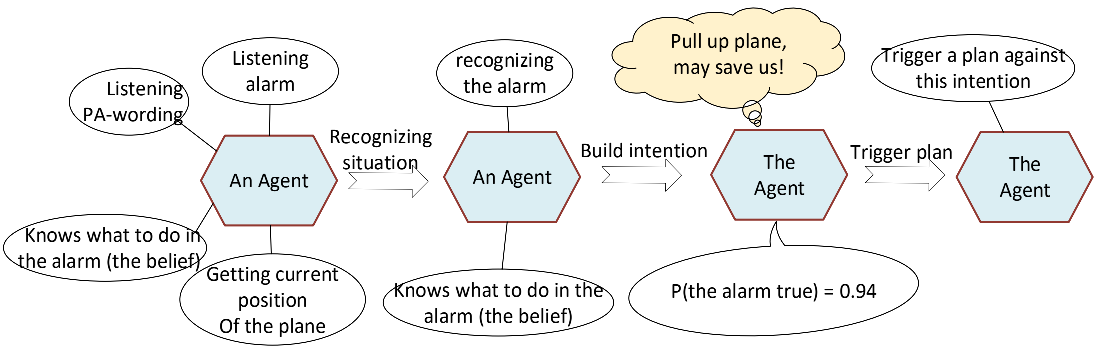


Figure 1.2. Stages of a BDI agent from perceiving to situation recognition to decision-making. Note that the probability that the situation is really what the alarm means is based on proposed agent model.

[5] OO2APL is a design pattern that can be used with an object-oriented programming language such as Java. This enables a programmer to develop a multiagent system comprising BDI-agents.

as SA related to emergencies where important factors to be noticed are platform alarms, public address announcements, and typical hazards like fire and smoke.

Finally, modeling decision-making, as mentioned in (3), means to develop the method(s) by means of which effective decision-making could be performed during emergencies, where the agent will not have ample time to weigh an option against every other possible option. In this respect, an NDM approach called *Recognition-Primed Decision-making* (G. Klein, 1998, 2004, 2008) model based agent model is proposed.

1.5 Spatial learning, situation recognition, and decision-making: an overview

This section reviews concepts in previous studies relevant to modeling the capabilities mentioned in Section 1.4.

1.5.1 Spatial learning: An overview of how people learn routes

In the event of an emergency, people should prefer evacuation through escape routes because they are designed to facilitate quick evacuation. Also, workers at offshore installations, children in schools, and employees in industries receive regular evacuation drills. The first aspect of human behavior that is discussed in this thesis is *route learning* due to its importance in emergency evacuation training. Though route learning is a ubiquitous experience, it involves features of places that stimulate efficiency, which can be seen in the resulting shorter evacuation time. Since evacuation time is a crucial aspect of during evacuation, a great deal of literature aims to unearth factors and features significant in route learning.

A few aspects on the subject of route learning are discussed here, but interested readers should consult the works by (Golledge, 1977, 1990, 1991, 1999a; Lynch, 1960; Passini, 1977; Tolman, 1948; Tuan, 1977). Also, chapters 2 & 3 explain in detail about the past work in the field of route/environment learning and relevant modeling approaches.

Route learning in a new environment is a classical problem that falls under a broader category of *Environmental Knowing* (Moore & Golledge, 1976) where people collect different features as cues to build their mental representation of the environment (Golledge, 1977; Lynch, 1960). This sort of environmental knowing based on learning individual routes is termed as route-based environmental learning. People learn and become aware of features in the environment, but this spatial learning develops through stages in that people begin with selective and fragmentary information about places, and over time they add individual information until a holistic mental image is prepared (MacEachren, 1992). This mental representation is often called a cognitive map, and it is this representation that a person uses to find routes from one place to another in the environment (Tolman, 1948).

The features (of places) may be called *landmarks* that are normally used for two purposes. The first is to orient in the environment (along a route), and the second purpose is to prime upcoming decision-points (e.g., where to take a turn, when to start looking for a door, and so on). These features are the real stimulants for learning a route that enhance a person's view about the environment. They can be real or imagined, confined to a particular point (say a post office), or extend in space, as the mountains of Sierra Nevada. Sometimes, the features of a landmark may be formed due to a feeling, for example, a forbidden house that is always drowned in a cloak of darkness. At times, these are mere subjective matters, of which others remain ignorant (Golledge, 1999a). One thing is for sure that people, somehow, discover the presence of these features and they use them to differentiate the places (Tuan, 1977).

Several types of spatial information are accrued as part of the features of a landmark. Humans learn identities of places, location, shape, color, size, magnitude, and the time of their existence (Golledge, 1990). This spatial information is what makes a feature salient so that a landmark could be remembered and could serve a purpose in navigating a route. Nonetheless, this saliency of features is not an invariant attribute. Due to individual differences and differences in the saliency of features, it is reasonable to assume that no two wayfinding tasks are equal (Allen, 1999). Studies suggest that landmarks with different saliency levels (of features) will have distinguished tendencies to stimulate one's ability to remember and recognize (McKinlay, 2016; Yurkiewicz & Tsao, 2012) a decision-point along a route.

In the context of built structures, such as buildings, airports, railway stations, hotels, and offshore installations, route learning is an important area of investigation because when an emergency situation occurs, some personnel in the environment may not be able to respond as quickly as is needed for evacuation. Muster drills are, therefore, part of a training curriculum for safety practices. However, new workers are more likely to forget a part of an escape route because of relatively little exposure. Similarly, workers who are exposed to more than one location to perform their jobs suffer the same risk of forgetting parts of an escape route at one of the workplaces because of extra burden on their memories for remembering spatial information belonging to different places (Chowdhury, 2016). To overcome this problem, escape routes are equipped with exit signs that are installed as navigation aids. The SOLAS Chapter II-2 regulation 13 (IMO, 2009) requires all exit signs to be made up of photo-luminescent material for better visibility in a blackout or low visibility conditions. Despite this, many incidents have been reported where the designated signage system fails to fulfill the needs in real emergencies. This is especially true when a designated escape route has been compromised and people will have to rely on their memories about landmarks to evacuate through other available routes (BBC, 2013; Weinspach, Gundlach, Klingelhofer, Ries, & Schneider, 1997). Moreover, workplaces such as engine rooms, are cluttered with a variety of machinery and tools, which render the workers at risk of being distracted or unable to see an exit sign when a real emergency brings panic (Mackintosh, 1973). Therefore, the present thesis urges the importance of adding exit signs at locations that have salient environmental features as discussed before.

1.5.2 Situation Awareness

Situation awareness is considered as an important prerequisite for decision-making in in safety-critical systems. The main theme of SA is to consider an operator's understanding of the system status separate from the actual system status (Woods, 1988). Traditionally, SA models have been of descriptive nature and based on human psychology (Endsley, 1988). The first serious attempt towards building a formal theory of *situation semantic* could be found in the works (Barwise & Perry, 1980; Barwise, 1989; Barwise & Perry, 1983). Devlin (1991) expanded Barwise and Perry's formalism to define *situation awareness* as being a phenomenon that refers to the information flow from a situation to a subject such that the subject can reason about the situation. Kokar et al. (2009) developed a situation theory ontology based on Devlin, Barwise, and Perry's formalism. However, these attempts do not incorporate the concept of uncertainty[6] that is a common experience when interpreting a situation. Naderpour, Lu and Zhang (2014) propose a cognitive decision support system that uses a dynamic Bayesian network (DBN) to represent situations and a fuzzy rule system that simulates operator's behavior by having rules of the form "if the probability of an event, say increased vapor pressure, is high and the event is catastrophic then the associated risk of the situation is not acceptable". The use of dynamic BN ensures that situations exist in time, and one situation could transform into another with the passage of time. However, the automatic generation of a DBN based on simple rules is not possible. The attributes or random variables in a Bayesian network (BN) are of propositional nature and the network is fixed having the same number of fixed variables for its lifetime. Therefore, the reasoning can be made based on the same fixed number of variables (Russell & Norvig, 1995, pp. 589-593). Modelling SA requires, as a first step, representing the situation and then reasoning about it. This latter part is necessary for modeling awareness. The present work proposes a computational model of SA based on Endsley's (Endsley, 1988, 1995, 2000) human SA model. The proposed model can be used to realize SA as a cognitive mechanism for agent minds so that awareness about an emergency can later be used in deciding a reasonable course of action.

[6] in perception, and in reasoning about the perceived cues

1.5.3 Decision-making: an overview of the RPDM model

As opposed to conventional theories of decision-making, such as the Classical Decision Making, or Behavioral Decision Making, the NDM approach is recent and more promising for situations that require decisions to come about under different mental stressors, such as time constraint, missing information, changing circumstances, and vague goals. Examples of such situations abound. Klein (1998), in his book *Sources of Power*, narrated many real stories from people working in the field. For example, the case of a 45 years old man suffering from trigeminal neuralgia. The man, on one day after waking up in the morning, finds severe pain on the right side of his face. He consulted some doctors who suggested different remedies but none was able to diagnose what the actual cause of the pain was. Finally, an expert neurologist saw him and in just five minutes diagnosed him as the classic case of trigeminal neuralgia. An avid reader can find, from the same account, other examples related with firefighting situations, marketing policy decisions, and many more. In many real-life situations, a naturalistic decision maker makes many critical decisions in a very short time (see Klein, 1998, pp. 2-3). This type of decision-making approach is based on an intuition that comes through experience (Klein, 2015) and it opposes the comparative evaluation of options that selects one (Klein, 1998, p. 20) out of many available choices. Lipshitz, et al., (2001) identified five major contributions of the NDM approach, viz., the recognition-primed decision-making model, dealing with uncertainty, team decision-making, decision errors, and research methodology. A widely studied model of NDM is *recognition-primed decision-making* (RPDM). RPDM is qualitative and needs a computational or quantitative equal for simulating situations (such as a fire situation on an offshore oil & gas platform) that suffer from immense time pressure. The aim of proposing a quantitative version of RPDM is to develop a cognitive agent model that can be used for simulating and experimenting with behaviors typical of operators and fire commanders. Certainly, such agent behaviors have potential benefits in improving virtual environment fidelity, which in turn gives better training opportunities for personnel working in the relevant industries, such as offshore oil and gas industries, aviation and nuclear plants.

A general model of pilot behavior during midair encounters was developed using the RPDM approach in (Hu, et al., 2018). The authors used BN to model different components in a midair encounter situation. The authors proposed a Bayesian recognition algorithm to model the situation recognition process. The algorithm provides a probabilistic similarity criterion, which is used in deciding the plan against the recognized situation. Cannellas & Feigh (2016) use Fast-and-Frugal heuristic (FFH) program and propose a simple and general mathematical form of NDM. The authors divide the whole process of decision-making into components: task type, utility functions, incomplete information, estimates of missing value, and cue weights. However, they did not explain how the model could be validated via a real-life or even a laboratory-based case study. A model of RPDM for multi-agent rescue simulation is presented in (Nowroozi, et al., 2012). The authors exploit Unified Modelling Language (UML) to present various components of the model. The general computational RPD (C-RPD) model was used to model a firefighter agent in the Rescue Agent Simulation environment (RoboCup Rescue Agent Simulation League, 2011), which is a platform of the rescue agent simulation league providing a benchmark for evaluating rescue operations in emergency situations. C-RPD algorithm was tested in national and international tournaments and it outperformed the other participating algorithms because it put out all the fires effectively. Other RPDM based models include an RPDM implementation integrated with the Belief-Desire-Intention agent model (Norling, 2004). In all cases where an RPDM is modeled in quantitative terms, there is still a need to validate if the model or the underlying theory, i.e., RPDM approach (Klein, 1998), is committed to serving as a potential candidate of human decision-making approach when classical decision logic is not a choice. The present work proposes to model RPDM and test the model results with results from real people. The model can be verified according to the theoretical work in (Klein, 1998), and it may validate RPDM as an approach that explains human decision-making under mental stressors, provided the results agree with the empirical findings.

1.6 Thesis Organization

The thesis is written in manuscript format. Four research articles have evolved during this study. Table 1.1 presents these papers to elaborate on the connection with the overall objectives of the thesis.

Table 1.1. Papers and their connection to the overall research objectives of the thesis.

Article titles	Research objectives	Associated Tasks
Chapter 2: A Generalized Stochastic Petri Net model of route learning for emergency egress situations.	• To develop an agent model that simulates how much of a route will be remembered after being exposed to a route for the first time. The model should simulate human-like behavior of remembering a route when a person, on the average, sees the same route. • To develop agents with different skills of remembering parts of a route.	• To understand the phenomenon of spatial learning, especially route learning. • Identify important literature. • Identify important escape routes in the virtual environment. • Identify landmarks along the selected escape route. • To classify landmarks based on saliency. • To understand how saliency of landmarks play a role in human's remembering and forgetting of a landmark when a new route is traversed. • Develop a Generalized Stochastic Petri Net model of route learning. • Estimate rates that can be used with different classes of landmarks in the proposed model. • Produce simulations • Validate the model based on empirical observations.

Article titles	Research objectives	Associated Tasks
Chapter 3: Human-Like sequential learning of escape routes for virtual reality agents	To develop agents with different route knowledge in an environment.	• To understand the impact of successive training on human competence. • To develop an agent model that simulates how much of a route will be remembered after being exposed to a route over *n*-successive exposures. • To develop a method whereby the model proposed in Chapter 2 can be used iteratively to represent the effect of successive training.
Chapter 4: Situation Awareness Modeling for Emergency Management on Offshore Platforms	To develop agents with different experiences of recognizing fire and evacuation emergencies.	• To understand the phenomenon of SA. • To understand different formal ways of modeling SA. • To develop an agent model that is capable of using cues or environmental features, e.g., alarms, hazards like fire & smoke, and announcements, to recognize a situation. • The agent should be able to learn different situations and classify one that is currently being observed with one or some of those in the situation-KB or a repertoire of situations. • This repertoire of situations will act as *experience* for the agent model proposed in Chapter 5.

Article titles	Research objectives	Associated Tasks
Chapter 5: On the realization of the Recognition-Primed Decision Model for artificial agents	To develop agents with different ability to make decisions as to what needs to be done in an emergency situation in a similar way that people make decisions.	• To understand how people make decisions in evolving emergencies. • To understand the factors people give importance in recognizing a situation and coming to a decision. • To understand how ontologies can be used to model situations. • To develop an agent model based on the concepts in RPDM. • The agent uses route data, and the repertoire of situations, learned in Chapters 2, 3, & 4 to employ the proposed realization of RPDM.

1.7 Contribution & Novelty

1. Very few attempts have been made where a route learning algorithm uses landmarks. Landmark recognition has been treated as a function of saliency and, to the knowledge of the author, no attempt has so far been made to incorporate the phenomenon of forgetting in a route learning mechanism for agents. This work, seemingly the first time, models forgetting and remembering of a navigation command as a stochastic process, where the probability of forgetting or remembering depends on the saliency of the nearby landmarks.

2. Some routes are difficult to remember, and some are easy to learn. Although there have been some references in the literature that mention this fact, no systematic classification of routes (especially the escape routes) is done based on environmental features such as landmark saliency. This work establishes a connection between known environmental features, such as cluttered spaces, and mirrored layouts, with the level of difficulty one may experience in remembering a landmark, and hence the associated navigation command.

3. Until the time of writing this thesis, the author has been unable to find out evidence that shows that a route learning algorithm, on the average, learns an escape route the way people learn. That is, no reports of a validation exercise where an algorithm is tested against real people's results have so far been obtained.

4. Most of the work in situation assessment deals with matching the environmental cues with those stored before to look for typical situations. A very few attempts are made where a situation assessment methodology uses reasoning to model SA so that all three levels of the Endsley's human SA model are satisfied.

5. Quantification of the components of RPDM has been taken up in only a few earlier works. It is difficult to see the exact implementation details in order to figure out why most of the results in those works were very promising. For example, it might be possible that the comparisons made between a proposed quantitative model of RPDM with real data contains an inherent discrepancy. An expert's opinion, which is considered as 'a result', collected while the expert is not present in that emergency situation, rather his opinion came through his experience, is certainly something in disagreement with the NDM approach where people make decisions under the stress due to emergency.

6. Each of the previous studies reported here has focused on a subset of the features of the RPDM approach. This work attempts to increase the size of that subset. That is, by modelling the constructs in the RPDM approach, such as mental simulation, which have not been given the attention they deserve.

Quantitative RPDM models need to validate the real people decision-making approach in emergencies. An attempt that compares the simulated results with empirical findings (i.e., data from real people) has not yet been reported in the literature. This work is unique in this sense too, that the simulated results are compared with virtually real data from real people performing the same tasks. The simulated results are explained in Chapter 5. There are similarities between the simulated and real data, but there are marked differences too. Chapter 5 has explained those differences but the main advantage that could be gained from the differences is that the reported agent may be considered as having different traits.

1.8 Co-authorship statement

The idea behind this work was originally proposed by Dr. Brian Veitch from the Faculty of Engineering and Applied Science. Dr. Veitch was interested to see how an agent can be developed that exploits the faculties of NDM philosophy. Applications of such an agent model in training simulators should have positive influence on participants training for offshore emergency situations. Dr. Faisal Khan from the Faculty of Engineering and Applied Science has contributed to this work by directing the author to the required areas of knowledge pertinent to intelligent agent modeling, stochastic processes, and other advanced modelling approaches such as Petri nets, Markov nets, and ontological representation of knowledge. This work is produced after a constant and continuous feedback from Dr. Khan and Dr. Veitch on related material that the author discovered, and that the author produced in terms of mathematical or formal models and computer programs in a variety of languages such as Python, Java, C++ and ML programming languages.

The author was responsible for composing this thesis. He conducted the literature review, developed the agent model for spatial learning module, situation awareness module, and finally the decision-support module that combines the spatial learning and situation awareness modules to form the proposed agent model for decision-making in emergencies. The author produced theoretical agent models and implemented them using corresponding technology. The author also used an experiment performed by Ms. Jennifer Smith in 2015 as a testbed for validating the agent model. In this respect, the author employed knowledge elicitation techniques to generate new knowledge pertinent to this study for validation — replay videos that recorded individual participants' activities during different trials of Smith's (2015) experiment were watched and values for related variables were extracted. Conclusions were drawn on the basis of which recommendations are presented.

References

Abū Ḥāmid Muḥammad ibn Muḥammad Al-Ghazālī. (1998). *The niche of lights: A parallel English-Arabic text (translated by David Buchman)* (1st ed.; D. Buchman, Trans.). Provo, UT, USA: Brigham Young University.

Alanen, L. (2014). The Second Meditation and the nature of the human mind. In D. Cunning (Ed.), *The Cambridge Companion to Descartes' Meditations* (pp. 88–106). https://doi.org/10.1017/CCO9781139088220.005

Allen, G. L. (1999). Spatial abilities, cognitive maps and wayfinding: bases for individual differences in spatial cognition and behavior. In R. G. Golledge (Ed.), *Wayfinding behavior* (pp. 46–80). The John Hopkins University Press.

Aristotle. (2001). *Aristotle's On the soul and on memory and recollection [Aristotle's De Anima]* (J. Sachs, Trans.). Ann Arbor, Michigan, USA: Green Lion Press.

BBC, A. N. (2013, October). *Nairobi siege: How the attack happened*. Retrieved from https://www.bbc.com/news/world-africa-24189116

Borowiec, S., & Lien, T. (2016, March). *AlphaGo beats human Go champ in milestone for artificial intelligence*. Retrieved from https://www.latimes.com/world/asia/la-fg-korea-alphago-20160312-story.html

Bratman, M. (1987). *Intention, plans, and practical reason*. Cambridge, MA: Harward University Press.

Chowdhury, S. (2016). *Optimization and Business Improvement: Studies in Upstream Oil and Gas Industry*. New Jersey: Wiley.

Dastani, M., & Testerink, B. (2014). From Multi-Agent Programming to Object Oriented Design Patterns. In F. Dalpiaz, J. Dix, & M. B. van Riemsdijk (Eds.), *Engineering Multi-Agent Systems. EMAS 2014. Lecture Notes in Computer Science, Volume 8758.* (pp. 204–226). https://doi.org/10.1007/978-3-319-14484-9_11

Descartes, R. (2015). *The passions of the soul and other late philosophical writings* (M. Moriarty, Trans.). Oxford, UK: Oxford University Press.

Dyson, G. (2012). *Turing's Cathedral*. New York, NY: Pantheon Books.

Endsley, M. (1988). Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *32*. https://doi.org/10.1177/154193128803200221

Endsley, M. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32–64.

Endsley, M. (2000). Theoretical underpinnings of situation awareness: A critical review. In M. R. Endsley & D. J. Garland (Eds.), *Situation Awareness Analysis and Measurement* (pp. 3–32). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Golledge, R. G. (1977). Multidimensional analysis and environmental behavior and design. In I. Altman & J. F. Wohlwill (Eds.), *Human behavior and environment: advances in theory and research*. Heidelberg: Springer Berlin Heidelberg.

Golledge, R. G. (1990). The Conceptual and Empirical Basis of a General Theory of Spatial Knowledge. In *Spatial Choices and Processes* (pp. 147–168). https://doi.org/10.1016/B978-0-444-88195-3.50014-3

Golledge, R. G. (1991). Cognition of physical and built environment. In T. Garling & G. W. Evans (Eds.), *Environment, Cognition, and Action: An Integrated Approach*. New York, NY: Oxford University Press.

Golledge, R. G. (1999). Human wayfinding and cognitive maps. In R. G. Golledge (Ed.), *Wayfinding behavior* (p. 9). Baltimore: The Johns Hopkins University Press.

Harman, G. (1976). Practical Reasoning. *The Review of Metaphysics*, *29*(3), 431–463. Retrieved from https://www.jstor.org/stable/20126812

Hassler, S. (2016). Marvin Minsky and the pursuit of machine understanding - Making machines-and people-think [Spectral Lines]. *IEEE Spectrum*, *53*(3), 7–7. https://doi.org/10.1109/MSPEC.2016.7420381

Hayes-Roth, B. (1995). An architecture for adaptive intelligent systems. *Artificial Intelligence*, *72*(1–2), 329–365. https://doi.org/10.1016/0004-3702(94)00004-K

Heath, B. L., & Hill, R. R. (2008). The early history of agent-based modeling. *IIE Annual Conference.Proceedings*, 971–976. Retrieved from https://search.proquest.com/docview/192465049?accountid=12378

IMO. (2009). *SOLAS: Consolidated Edition* (5th ed.). London, UK: International

Maritime Organization.

Kahneman, D., Slovic, P., & Tversky, A. (Eds.). (1982). *Judgment under Uncertainty: Heuristics and Biases*. Cambridge, United Kingdom: Cambridge University Press.

Kay, A. (1984). Computer software. *Scientific American*, *251*(3), 53–59.

Khan, S. A. (1989). Whether man is capable of free thinking. In *Freedom of Thought in Islam*. Karachi, Pakistan: Royal Book Company.

Klein, D., Marx, J., & Fischcach, K. (2018). Agent-Based Modeling in Social Science, History, and Philosophy. An Introduction. *Historical Social Research*, *43*(1).

Klein, G. (1998). *Sources of Power*. Cambridge, MA: MIT Press.

Klein, G. (2004). *The Power of Intuition*. Doubleday.

Klein, G. (2008). Naturalistic Decision Making. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(3), 456–460. https://doi.org/10.1518/001872008X288385

Lucas, J. R. (1961). Minds, machines and Gödel. *Philosophy*, *36*(137), 112–127. Retrieved from http://www.jstor.org/stable/3749270

Lynch, K. (1960). *The Image of the City*. Cambridge, MA: MIT Press.

MacEachren, A. M. (1992). Application of environmental learning theory to spatial knowledge acquisition from maps. *Annals of the Association of American Geographers*, *82*(2), 245–274.

Mackintosh, N. (1973). Stimulus selection: Learning to ignore stimuli that predict no change in reinforcement. In R. A. Hinde & J. S. Hinder (Eds.), *Constraints on Learning* (pp. 75–96). London: Academic Press.

Maes, P. (1995). Artificial life meets entertainment: lifelike autonomous agents. *Communications of the ACM*, *38*(11), 108–114. https://doi.org/10.1145/219717.219808

Man, L. (2015). An Agent-based Approach to Automated Merge 4D Arrival Trajectories in Busy Terminal Maneuvering Area. *Procedia Engineering*, *99*, 233–243. https://doi.org/10.1016/j.proeng.2014.12.531

McCarthy, J. (1979). *Ascribing mental qualities to machines (Memo No. 326)*. Palo Alto, CA.

McKinlay, R. (2016). Technology: Use or lose our navigation skills. *Nature*, *531*(7596), 573–575.

Minsky, M. (1988). *The society of mind*. New York, NY: Simon & Schuster.

Minsky, M. (1991). Conscious machines. *Machinery of Consciousness. Proceedings of the National Research Council of Canada, 75th Anniversary Symposium on Science in Society, June 1991.* NRC.

Moore, G. T., & Golledge, R. G. (Eds.). (1976). *Environmental knowing: Theories, research, and methods*. Stroudsburg, Pennsylvania: Dowden, Hutchinson & Ross, Inc.

Nwana, H. S. (1996). Software agents: an overview. *The Knowledge Engineering Review*, *11*(3), 205–244. https://doi.org/10.1017/S026988890000789X

Parsons, S., & Wooldridge, M. (2002). Game Theory and Decision Theory in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, *5*(3), 243–254. https://doi.org/10.1023/A:1015575522401

Passini, R. (1977). *Wayfinding : a study of spatial problem solving with implications for physical design*. Pennsylvania State University, Pennsylvania, USA.

Penrose, R. (1989). *The emperor's new mind: Concerning computers, minds, and the laws of Physics*. Oxford University Press.

Penrose, R. (1991). The emperor's new mind. *RSA Journal*, *139*(5420), 506–514. Retrieved from http://www.jstor.org/stable/41378098

Rao, A. S., & Georgeff, M. P. (1995). BDI Agents: From theory to practice. In V. Lesser & L. Gasser (Eds.), *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* (pp. 312–319). Retrieved from 0262621029

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Ross, S. A. (1973). The economic theory of agency: the principal's problem. *American Economic Review*, *63*(2), 134–139.

Russell, S. J., & Subramanian, D. (1995). Provably Bounded-Optimal Agents. *Journal of Artificial Intelligence Research*, *2*, 575–609. https://doi.org/10.1613/jair.133

Selfridge, O. G. (1988). Pandemonium: a paradigm for learning. In *Neurocomputing: foundations of research* (pp. 115–122). Cambridge, MA: MIT Press.

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, *60*(1), 51–92. https://doi.org/10.1016/0004-3702(93)90034-9

Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, MA: Cambridge University Press.

Smith, D. C., Cypher, A., & Spohrer, J. (1994). KidSim: programming agents without a programming language. *Communications of the ACM*, *37*(7), 54–67. https://doi.org/10.1145/176789.176795

Smith, J. (2015). *The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios [Master thesis]*. Memorial University of Newfoundland, St. John's, NL, Canada. Retrieved from: https://research.library.mun.ca/8401/1/thesis.pdf

Tolman, E. C. (1948). Cognitive maps in rats and men. *The Psychological Review*, *55*(4).

Tuan, Y.-F. (1977). *Space and place: The perspective of experience*. Minneapolis: University of Minnesota Press.

Turing, A. M. (1947). *Lecture to the L.M.S. [London Mathematical Society] Feb. 20, 1947*. London, UK: This record is held by Cambridge University: King's College Archive Centre.

Turing, A. M. (1950). I.—Computing machinery and intelligence. *Mind*, *LIX*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

Tversky, A., & Kahneman, D. (1974). Judgment under Uncertainty: Heuristics and Biases. *Science*, *185*(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press.

Weinspach, P. M., Gundlach, J., Klingelhofer, H. G., Ries, R., & Schneider, U. (1997). Analysis of the Fire on April 11th, 1996; Recommendations and Consequences for Dusseldorf Rhein-Ruhr-Airport. *Staatskanzlei Nordrhein-Wstfalen, Mannesmannufer*, *1*.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. West Sussex, United Kingdom: John Wiley & Sons Ltd.

Wooldridge, M., & Jennings, N. R. (1995). Agent theories, architectures and languages: a survey. In M. J. Wooldridge & N. R. Jennings (Eds.), *Intelligent Agents*. https://doi.org/10.1007/3-540-58855-8

Yurkiewicz, I. R., & Tsao, J. W. (2012). Book Review: Why people get lost: the psychology and neuroscience of spatial cognition by Paul A. Dudchenko. *Journal of the Neurological Sciences*, *313*(1–2), 197–198.

Zsambok, C. E. (1997). Naturalistic Decision Making: Where are we now? In C. E. Zsambok & G. Klein (Eds.), *Naturalistic Decision Making* (pp. 3–16). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Chapter 2

A Generalized Stochastic Petri Net model of route learning for emergency egress situations[‡‡]

Co-authorship statement. A version of this chapter has appeared as an article in the journal titled *Engineering Applications of Artificial Intelligence* published by Elsevier. The author, Syed Nasir Danial, has developed and implemented the Petri-net based model and extracted the empirical data using re-play video files for validation of the model. The co-authors Dr. Faisal Khan and Dr. Brian Veitch supervised this study. All authors read and approved the final draft.

Abstract. Route learning is an essential activity for a person visiting a new environment. The element of forgetting a location (called decision point) along a route, where a change in direction is needed, is of immense importance especially during emergency evacuation scenarios. It is this element that has not been given the attention it deserves in developing a route learning algorithm. This work proposes a model of route learning in a new environment based on landmarks using generalized stochastic Petri nets because landmarks based route learning has been observed as a method natural to humans. The model takes information about landmarks along a route and associated navigation commands and then chooses whether to save this information as part of the learned route or not. The selection is made by exploiting stochastic transitions for which the firing rates are dependent on the type of landmark encountered at a decision point. The final output is a route having some decision points missing; resembling the situation that humans encounter after they visit a route in a new environment. The model results closely match empirical results obtained with human subjects.

[‡‡] Danial, S. N., Khan, F., & Veitch, B. (2018). A Generalized Stochastic Petri Net model of route learning for emergency egress situations. *Engineering Applications of Artificial Intelligence*, *72*, 170–182. https://doi.org/10.1016/j.engappai.2018.03.024

2.1 Introduction

Representing human-like intelligent behavior is an active research area in artificial intelligence (AI). Today, technological advances seem to support the idea that some mental modalities may be modeled as AI constructs. Speech recognition on ordinary cell phones, and a recent defeat of Lee Sedol, the world champion of Go[8] by AlphaGo (Borowiec & Lien, 2016) — a computer program — are to mention but a few. This work considers learning as a mental modality and considers only one type, viz., the *route learning* in a new environment, to construct a model based on empirical understanding of the human route learning process. The purpose is to have a model that can be used by a software agent so that the agent can produce human-like behavior, such as forgetting a portion of escape route, in a training simulator for emergency egress.

How people learn routes in a new environment is a classical problem. Route learning falls under a broader subject area of *Environmental Knowing* where people collect different landmarks as cues to build their own mental representation of the environment (Golledge, 1977). This mental representation is often called a cognitive map and it is this representation that a person uses to find routes from one place to another in the environment (Tolman, 1948).

Learning a route becomes of prime importance when one considers emergency evacuation situations. People need training to egress through designated routes in a

[8] Go is an ancient Chinese abstract strategy adversarial two player board game that aims to occupy more space on the board than the opponent. Due to shear complexity of the game, the one who is better in intuition, creativity and strategic planning will most likely win.

facility, such as an offshore petroleum platform. In this regard, a high-fidelity virtual environment (VE) is considered a suitable training environment compared to traditional classroom type training using video tapes or presentation-based methodology. A serious limitation of VEs is the general unavailability of reasonably intelligent agents to support various training tasks. For example, if a human participant watches an agent performing a typical task, like moving to a muster station in the case of a fire alarm and corresponding public announcement call for evacuation, it is hard to show the human participant cases where the agent goes astray, because the agent is typically given access to a complete map of the environment and a related path finding algorithm, such as A* (Buckland, 2004; Hart, Nilsson, & Raphael, 1968), and so can use this map to perfectly retrieve the desired route information. This enables the agent to always find a correct path for the desired destination instead of making it possible to expose the agent to the dangers of taking a wrong route. One way to get around this problem is to model remembering and forgetting of landmarks, because it is observed that missing a landmark results in forgetting needed turn along a route to a destination (Beusmans, Aginsky, Harris, & Rensink, 1995). Although remembering and forgetting can be seen as functions of knowledge retrieval mechanisms, such as similarity matching and frequency biases (Reason, 1990 pp. 13, 97-98, 125-126), they can also be modelled as functions of an information gathering mechanism, where an agent does not remember a landmark due to lack of practice or retention time-out. The behavior of these naïve agents is important in the assessment of difficulties of learning a route due to physical properties of the environment. If given more than one exposure to a route, people tend to adapt to these difficulties because of practice (Kyritsis, Gulliver, & Morar, 2014).

The form of route learning the present work involves is like ant colony optimization (ACO) algorithms (Dorigo & Stutzle, 2004) in the sense that ACOs are inspired by pheromone trails and use the trails as landmarks to guide the search of finding the best route — just as people use landmarks to remember certain moves such as *move left*, *move right*, *go straight* along their route. We refer to such moves as *navigation commands* (NC) in this work. This paper presents a Generalized Stochastic Petri Net (GSPN) model of route learning (GSPNRL) based on remembering landmarks along the route. This means that the model allows storing some landmarks, while at the same time producing the effect of forgetting by not storing some of the landmarks. The GSPN model of route learning is explained by describing parts of the model with the help of algorithms presented in Section 2.4. However, the main contribution of this study is the GSPN model of route learning.

Section 2.2 discusses related work and explains different concepts in the human route learning process, along with some artificial intelligence aspects. Section 2.3 covers topics in the human route-learning process and presents some definitions related to the types of Petri nets used in this work. In Section 2.4, the proposed model is presented with elements of training and learning. Section 2.5 describes an experiment whose data are used to test the validity of the model. The model results are compared with the empirical results. The mathematical verification of the model is also presented in this section. Results and future directions are discussed in Section 2.6.

2.2 Related work

The subject of *Environmental Knowing* has an important role in situation awareness (Endsley, 1995). Lehtonen, Sahlberg, Rovamo, and Summala (2017) show how learning about an environment increases situation awareness and thereby decreases possible accidents of child bicyclists. Beusmans et al. (1995) determine how a route learning process is involved in one's continues effort to stay aware of the surrounding environment during driving on a road. The authors performed an experiment in a VE where sixteen participants were used as paid volunteers. The participants were given verbal navigation commands, such as take the next left or right, without pinpointing the landmarks and asked to remember the route, which was a 1770m long complex road map. The authors found that some participants developed the skill of navigating using landmarks and some of them went further as they developed a mental image of the environment. The latter showed less situation awareness between intersections on roads and the former showed situation awareness as their ubiquitous property irrespective of their location on route.

Gale, Golledge, Pellegrino, and Doherty (1990) performed an experiment with children of ages 9-12 to investigate spatial knowledge acquisition. Their general result about the mode of learning supports active exploration through field trips in the real environment. However, the use of video tapes also proved to be fairly effective in terms of representing fundamental components in spatial learning. The authors found that children learned more at intersections where they needed to make decisions about their move, rather than in between the intersections. These intersections are termed decision points. They also suggest that during a route learning task, knowledge about the features of the surroundings starts being stored concurrently as a background process. Another important finding was that successful navigation does not require extensive knowledge about the route, rather route navigation seems to be parsimonious, i.e., the modeling of route learning may be simpler than other types of learning tasks. Plank, Snider, Kaestner, Halgren, and Poizner (2014) used a large immersive VE to investigate human memory about remembering positions of 39 distinct objects in the VE. The experiment started by making the subjects explore the positions on day 1. On day 2, the positions of some of the objects were changed and the subjects needed to recognize that. The subjects correctly identified 87% of times that the objects were moved or not. The authors also suggest that these findings could help understand neurocognitive stages related to an early first-pass allocentric space processing, followed by integration of the objects' locations in the spatial cognitive map. An allocentric spatial representation expresses location of an object in an environment with reference to other objects, provided the environment with all the objects takes an arbitrary orientation that defines left/right and up/down positions (Grush, 2000). On the contrary, the egocentric spatial representation considers the self as the reference point.

Studies of rats show that the brain creates multiple cognitive maps, each representing a different segment of the environment (Derdikman & Moser, 2010). The study (Eilam, 2014) details exploration of an unfamiliar environment using home-base and looping behavior in mice. The author gives a detailed account of the path integration, retracing, and wall-following mechanisms, and describes analogies between humans and other animals in biobehavioral mechanisms. Eilam (2014) also suggests that the three important phases in spatial learning, viz., the path integration phase, the place recognition (or landmark recognition) phase, and the reorientation phase that works while using representations of a surface layout (Wang & Spelke, 2002), may be explained in terms of looping being a way to do path integration, home-base behavior being an expression for place recognition, and wall-following being moving with reference to a surface layout.

The use of AI techniques in matters related to emergency situations is an important area of investigation. Ramchurn et al. (2016) develop a disaster response system called *Human Agent Collectives-Emergency Response* (HAC-ER) system. The HAC-ER serves as a mediator between humans and agents and provides a platform where humans and agents can develop a social relationship to address a number of evolving phenomena in an emergency situation, such changing demands. The authors develop a novel way to team up agents and humans to act more effectively in emergency situations. Sud, Andersen, Curtis, Lin, and Manocha (2007) propose a multi-agent navigation graph for real-time path planning for a dynamic VE. The agents use this graph as a global data structure to compute, in parallel, the maximal clearance paths without using a separate path planning data structure for individual agents. Kang, Kim, and Kim (2010) deploy a Region of Interest (ROI) in a VE to enable the system to detect abnormal shortest routes selected by different users. The ROI with the highest level is selected and a discretized path graph (DPG) is constructed using the data sampled in the selected ROI. The VE's existing navigation-graph is then integrated with the DPG using Delaunay triangulation. Nonetheless, in real emergencies, several risks related to human factors are in play. This means it is possible that some trained personnel become overwhelmed by mental stress and make mistakes (Reason, 1990). Musharraf, Khan, Veitch, MacKnnon, and Imtiaz (2013) assess human reliability during emergencies on offshore petroleum platforms. They use four major factors that influence stress, which in turn deteriorates human performance. Norazahar, Khan, Veitch, and MacKinnon (2016) present a method based on Bayesian networks to identify critical human and organizational factors in escape and evacuation systems.

2.3 Background concepts

2.3.1 Human route learning process

Route learning is defined as a phenomenon in which a navigator recognizes an origin and a destination location, and identifies route segments, turn angles, and the order in which they appear, in order to make a complete route (Golledge, 1999a). In other words, route learning in humans is characterized by associating specific moves or turn angles with a particular entity. The entity is either situated or it has some special relationships with a location in the environment. It turns out that the entity has striking features that make it easy to remember without an explicit reference to something else. This self-appealing characteristic is due to its features in relation with the nearby objects such that the entity stands out from its immediate environment. Such an entity is commonly referred to as a *landmark*, and is reported in literature with different names, such as *occurrence*, *reference point*, and *decision-point* (Golledge, 1991). Various kinds of attributes that may be associated with a landmark include structure or shape, size or color, or some other functional characteristic such as a school, a hospital, or even a fear associated with a particular house in a neighborhood (Golledge, 1999a). Lynch (1960) in his book *The Image of the City* argued that among the other features of an environment, such as nodes, paths, boundaries and districts (in case of a city), landmarks are the most dominant and well-known. There are four important attributes associated with a landmark. The first is its identity, which can be given by attaching a name or a label. The identity can be a place specific cue or a class specific cue. A place specific cue is identified by a unique place location such as The Grand Mosque of Makkah in Saudi Arabia, or the British Museum on Great Russell Street, Bloomsbury, London WC1B 3DG, United Kingdom. A class specific cue is identified by a general label, such as a food court. The second attribute is location that can be specified either by using a precise metric system or by a less precise means by employing words like "near", "far from", "in front of". The third attribute is a measure of magnitude that determines how distinctive the entity is before it could be considered as a landmark. The magnitude may include size of the entity, its volume, shape or even color. The permanence of the landmark on the temporal scale is the final attribute that plays a key role for developing spatial knowledge structures (Golledge, 1993).

To humans, and also with other animals, there is a natural tendency to remember a navigation command (such as *move left* or *go straight*) one has made once traversing near a landmark. It has been observed that practicing a route traversal again and again strengthen the binding between landmarks and associated navigation commands. This tendency is the major contributor, in humans, to learn routes. The present study uses place specific names to identify landmarks along a route. The landmarks considered here are invariant in space and time. Each landmark is selected empirically, at or near a decision-point, due to its salience in the VE.

2.3.2 Petri Nets

Petri (1966) developed a network that became known as a Petri net. Since then, the original Petri net has seen much advancement. Colored Petri-Nets (CPN) (Jensen, 1981, 1996), Stochastic Petri-Nets (SPN), and Generalized Stochastic Petri-Nets (Ajmone Marsan, 1990; Ajmone Marsan, Conte, & Balbo, 1984; Bause & Kritzinger, 1996; Trivedi, 2002) are among a few widely-used variants in a multitude of disciplines. Petri nets are an important class of formal design and modelling techniques. Systems that exhibit concurrency or parallelism are the best examples where the strength of Petri nets can be witnessed. The concurrency might be involved in all or some of the events occurring in the system under certain constraints, such as precedence or frequency of the occurrence of these events (Peterson, 1977). It is important to note that the standard Petri nets, also called Place-Transition nets, do not involve any concept of time, and therefore, can only be used to understand qualitative properties of a system. However, timed Petri nets are used in many disciplines from biological to engineering sciences to model and analyze the quantitative aspects of a system under investigation (Febbraro, Giglio, & Sacco, 2016; L. Li & Yokota, 2009; Maciel, Trivedi, Matias, & Kim, 2011; Murata, 1989).

2.3.2.1 PLACE-TRANSITION NETS

The graphic representation of a Place-Transition (PN) net forms a bipartite graph containing two main components: places and transitions, connected by directed arcs. A place is drawn using circles and is used to represent conditions or variables. A transition is drawn using rectangles. They represent any activity or function call that may alter the conditions or variables on places. Each place has tokens, which represent a value or data of the associated variable or condition. Tokens in PNs are represented by small black dots inside a place, or by a number if they are too many. The number of tokens on each place defines the state of the PN and is usually referred to by the PN marking. A PN marking is described by a vector, M, where the ith component of M represents the number of tokens at the ith place. In case of CPNs, there can be many types of tokens in a single net, whereas in the PNs, only one type of token is defined. Thus, PNs contain type-free tokens. A token can move from one place to another through an enabled transition only. If a transition is not enabled, the tokens from its input place cannot move through. A transition is enabled if all of its input places contain at least one token. An enabled transition may fire, and when it fires it deletes tokens from each of its input places and creates them on each of its output places depending on the multiplicity of the involved input and output arcs (Ajmone Marsan et al., 1984).

2.3.2.2 COLORED PETRI NETS

The notion of color or *type* for tokens in a Petri net was first introduced by Jensen (1981). The result is Colored Petri net. A CPN is an extension of PN, where a *color* is used to represent datatype or type (such as an integer, a string, or a composite type). A token in CPN is defined to be of a specific color. A CPN is defined in (Bause & Kritzinger, 1996 pp. 147-148) as a 6-tuple $CPN = \{P, T, C, I^-, I^+, M_0\}$, where:

(a) P is a finite and non-empty set of *places*,

(b) T is a finite and non-empty set of *transitions*,

(c) $P \cap T = \varnothing$,

(d) C is a color function defined from $P \cup T$ into finite and non-empty sets,

(e) I^{-}, I^{+} are, respectively, the backward and forward incidence functions defined on P×T such that $I^{-}(p,t), I^{+}(p,t) \in [C(t) \rightarrow C(p)_{MS}], \forall (p,t) \in P \times T$,

(f) M_0 is a function defined on P describing the initial marking such that $M_0(p) \in C(p)_{MS}, \forall p \in P$.

CPNs offer a convenient way to model complex concurrent problems. The concept of color or datatype allows that typed places can take typed tokens that represent certain real-world processes. Also, different types of tokens, representing different processes, would get the benefit of using the same net resulting in low net size and complexity (Gorton, 1993).

2.3.2.3 STOCHASTIC PETRI NETS

The continuous-time Stochastic Petri-Net is an extension of PN. Formally, it is defined over $PN = \{P, T, I^{-}, I^{+}, M_0\}$, as $SPN = (PN, \Lambda)$, where

(a) $P = \{p_1, p_2, \ldots, p_n\}$ is a finite and non-empty set of places,

(b) $T = \{t_1, t_2, \ldots, t_m\}$ is a finite and non-empty set of transitions,

(c) $P \cap T = \varnothing$,

(d) I^{-}, I^{+} are, respectively, the backward and forward incidence functions in $P \times T \rightarrow \mathbb{Z}$,

(e) M_0: $P \rightarrow \mathbb{Z}$, is the initial marking, where a marking represents the distribution of tokens over the places,

(f) $\Lambda = \{\lambda_1, \lambda_2, \ldots \lambda_m\}$, where λ_i is, possibly, the marking dependent rate of transition t_i. This means that the firing time of each transition follows an exponential distribution of the random variable involved.

2.3.2.4 Generalized Stochastic Petri nets

Bause and Kritzinger (1996) define a GSPN as a 4-tuple $GSPN = (PN, T_1, T_2, W)$, where,

(a) $PN= (P, T, I^-, I^+, M_0)$ is the underlaying Place-Transition net,

(b) $T_1 \subseteq T$ is the set of timed transitions, $T_1 \neq \emptyset$,

(c) $T_2 \subseteq T$ is the set of immediate transitions, $T_1 \cap T_2 = \emptyset$, and $T = T_1 \cup T_2$,

(d) $W=(w_1, w_2, \ldots, w_{|T|})$ is an array whose entry $w_i \in \mathbb{R}^+$

i. is a (possibly marking dependent) rate of a negative exponential distribution specifying the firing delay, when transition t_i is a timed transition, i.e., $t_i \in T_1$ or

ii. is a (possibly marking dependent) firing weight, when transition t_i is an immediate transition, i.e., $t_i \in T_2$.

A GSPN model or its colored variant in which colored tokens are used, as with the case of a CPN, can be converted into a Markovian model for performance evaluation purposes. The net is unfolded first. The unfolding process includes generating one copy of the original net per color by replicating the places, transitions, and arcs. The underlying Markovian model is constructed based on the unfolded net, and its solution is used to define performance measures on the colored net (Balbo, Chiola, & Bruell, 1992) such as estimating the steady state probability distribution as reported in Section 2.5.5.

2.3.3 Continuous Time Markov Chain

A stochastic process $\{X_n: n = 0, 1, \ldots\}$ with a finite or countably infinite state space $\mathbb{S}$ is said to be a Markov chain, if for all $i, j, i_0, \ldots, i_{n-1} \in \mathbb{S}$, and $n = 0, 1, 2, \ldots$, the probability $P(X_{n+1}=j \mid X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0=i_0)=P(X_{n+1}=j|X_n=i)$.

The above property is called Markovian property according to which, given the state of a Markov chain at present, X_n, its future state, X_{n+1}, is independent of the past states $X_{n-1}, X_{n-2}, \ldots, X_1, X_0$. If the time spent between transitions is a continuous random variable such that when the process X_n or $X(n)$ enters in state i, it remains there for some period of (exponentially distributed) time and then moves to the next state, the process $X(n)$ is said to be a Continuous Time Markov Chain (CTMC) (Ghahramani, 2005; Trivedi, 2002). The sample path of a typical CTMC may look like the one depicted in Figure 2.1, where each horizontal line represents the duration of time the

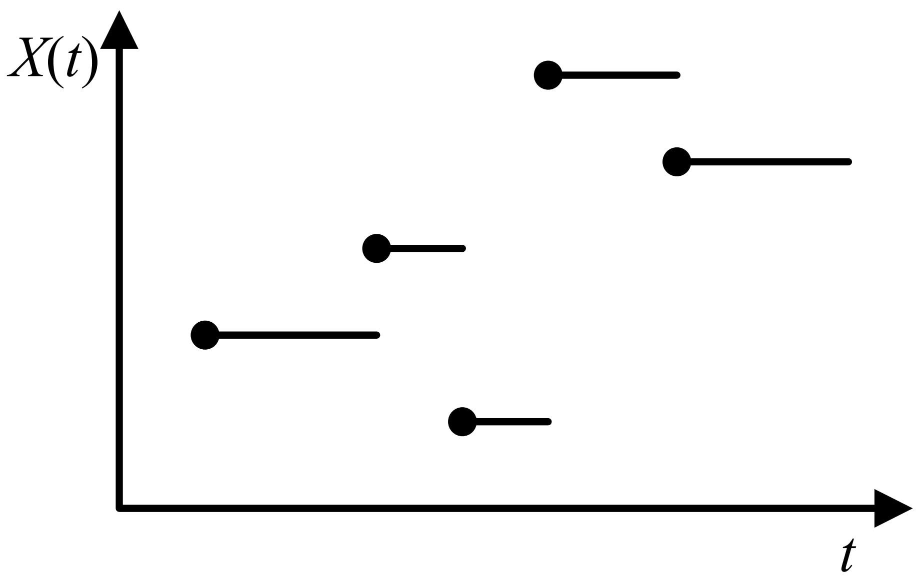


Figure 2.1. The sample path of a typical CTMC. Source: (Adopted from Ajmone Marsan (1990).)

process is in a particular state. The time the process spends in a state is called the sojourn time of that state. The sojourn times follow the negative exponential distribution, because this is the only distribution with the memoryless property as required by the Markovian property of CTMC. The density function of sojourn time is given by

$$f_X(x) = \mu e^{-\mu x} u(x), \tag{2.1}$$

where u is the unit step function, and μ is the rate of the pdf.

2.3.4 CTMC model

A CTMC model can be constructed by specifying a state space, $\mathbb{S}$ and a transition rate matrix, Q (also called the infinitesimal generator). The state space, $\mathbb{S}=\{s_1, s_2, \ldots, s_n\}$, is a finite collection of states the CTMC model can visit (Mo, 2013). The matrix Q is a $|\mathbb{S}|\times|\mathbb{S}|$ matrix, and it has the off-diagonal elements as nonnegative rationals. The diagonal entries follow $-(\sum_{i\neq j} \lambda_{j,i})$ that consequently make the sum of each row equals to zero (Aziz, 2000).

2.3.4.1 Solving CTMC model

The probability that a CTMC model will stay in a state j after t units of time provided that it is in state i presently is denoted by $p_{ij}(t)$ and is represented as follows:

$$p_{ij}(t) = P[X_{t+s} = j | X_s = i]. \tag{2.2}$$

Mo (2013) argues that an irreducible, finite CTMC has a unique steady state probability distribution vector, π. In case the CTMC is irreducible and infinite then there will at most be one steady state vector. A CTMC ($\mathbb{S}$, Q) can be solved by solving the *balance equation* of ($\mathbb{S}$, Q). The balance equation of the CTMC ($\mathbb{S}$, Q) is given by:

$$\pi Q = 0. \tag{2.3}$$

Since, π is a steady state vector, it satisfies:

$$\sum_j \pi_j = 1. \tag{2.4}$$

Solving Eqs. (2.3) and (2.4) yields the steady state probability distribution of the CTMC model ($\mathbb{S}$, Q). As an example, consider a simple problem of an electric kettle that can be either in good running state or in a faulty state. The state space is $\mathbb{S}$=$\{runnable, faulty\}$. If the transition rate, q_{rf}, from runnable to faulty state is λ and from faulty to runnable state (say after getting repaired) is μ, then, the left hand side of Eq. (2.3) can be written as:

$$\pi Q = (\pi_r, \pi_f)\begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

which can be solved by comparing to zero to give the steady state distribution vector as $\pi = \left(\pi_r, \pi_f\right) = \left(\frac{\mu}{\lambda+\mu}, \frac{\lambda}{\lambda+\mu}\right)$.

2.4 Methodology

A number of experiments on route learning can be seen as a phenomenon that occurs on two axes. The first is the training part. This may employ any method or methods of giving spatial knowledge to a learner, for example, through verbal communication or maps, video touring, or practice in a virtual environment. The second part is to

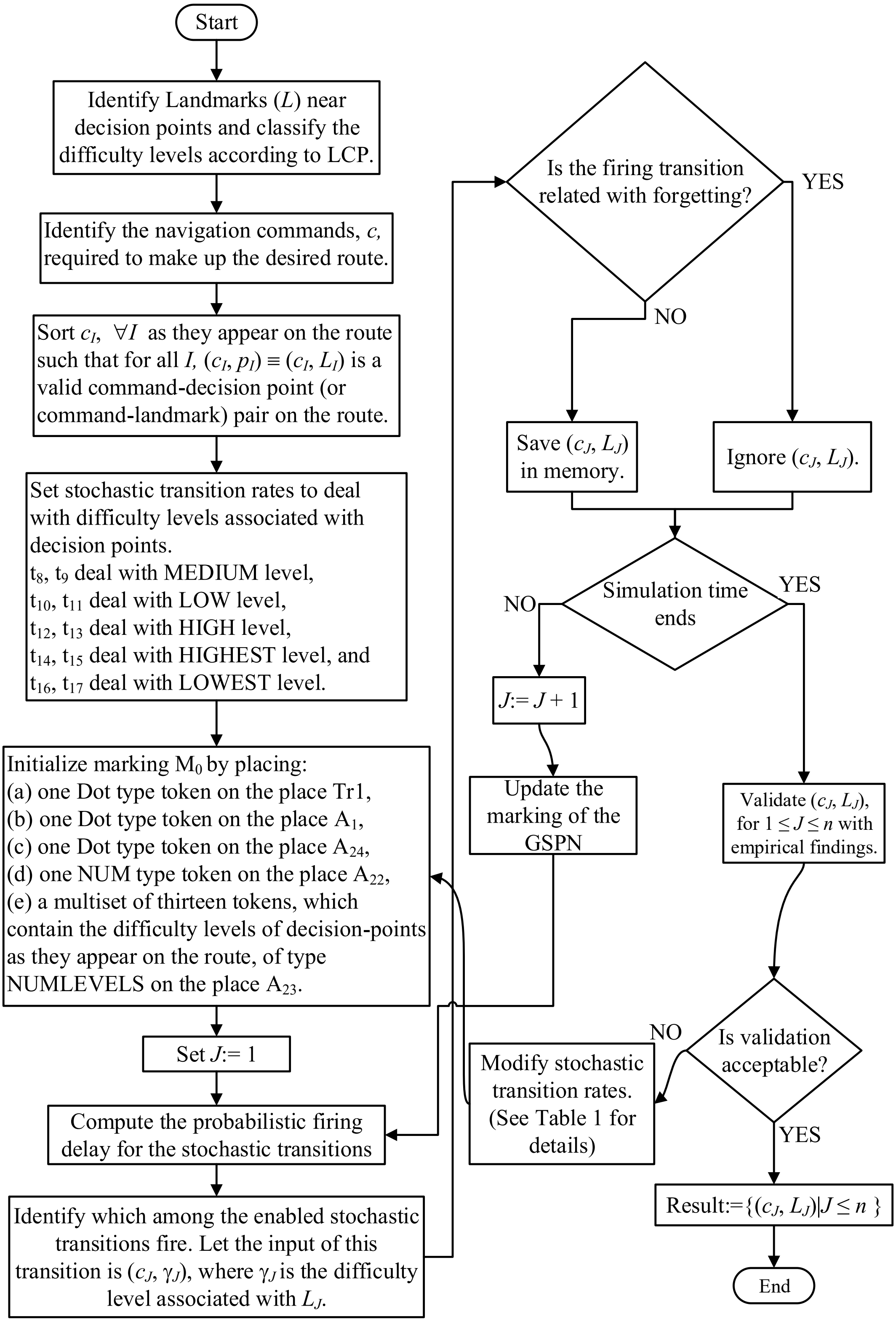


Figure 2.2. The general methodology for modeling landmarks-based route learning.

assess the extent to which the spatial knowledge is retained after the training is complete. Allen (1999) says that a successful landmark-based navigation requires recognition of landmarks and then remembering the associated navigation commands. The level of difficulty in remembering landmarks and associated navigation commands is modeled in the present work. Allen (1999) further says that, due to individual differences and other environmental factors, it is reasonable to suppose that no two wayfinding tasks are alike. We have developed a modeling approach, which is explained in Figure 2.2, to develop a model of route learning based on landmarks.

The approach is followed by modeling three main components (see Figure 2.3) of route learning activity separately: (*i*) the trainer model (N1), whose purpose is to pass on navigation commands in succession to N2, (*ii*) the action generation and learning model (N2), which generates actions in the environment according to the received navigation command and then attempts to remember the actions and landmarks (N2 receives inputs from N1 and N3), and (*iii*) the route model (N3) that sends a sequence of perceived difficulty levels associated with landmarks or decision points to N2. This route information is provided, on the place A_{23}, as part of the initial marking, M_0, where a marking in a Petri net refers to the distribution of tokens on the places at any

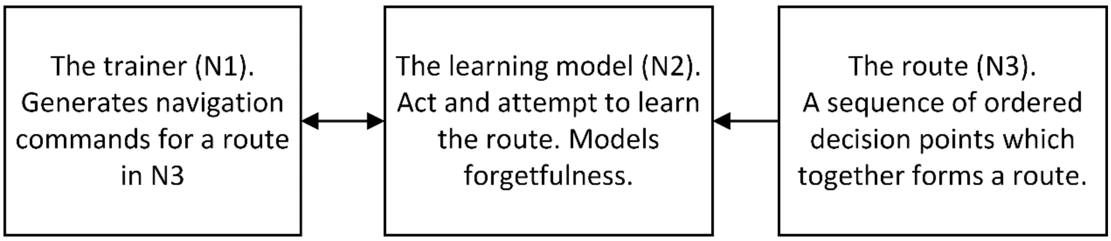


Figure 2.3. A block diagram of the proposed model

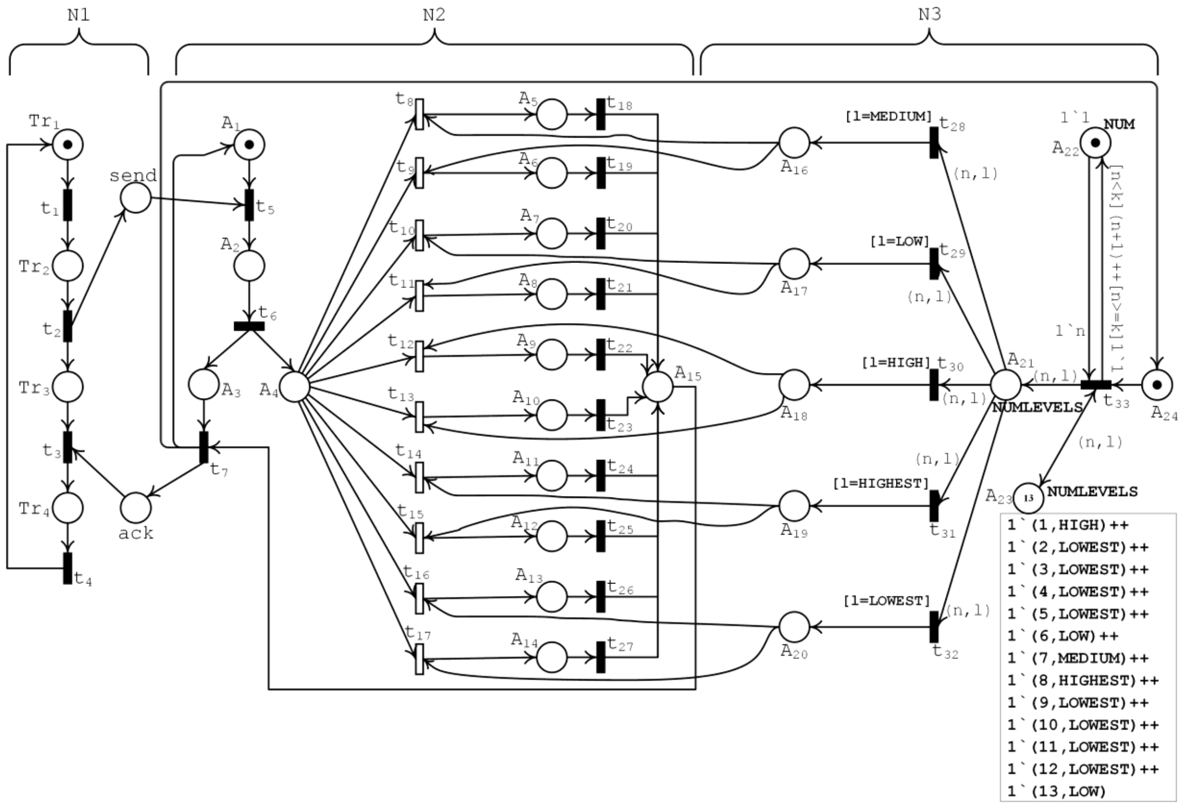


Figure 2.4. GSPN model of landmark based route learning. Transitions t_8 - t_{17} are the stochastic transitions. All other transitions are immediate transitions. Transitions, t_8, t_{10}, t_{12}, t_{14}, and t_{16}, are executed when the decision is *not to store a landmark and its associated navigation command*, whereas, the transitions, t_9, t_{11}, t_{13}, t_{15}, and t_{17}, are used for *saving* the information about landmarks and associated navigation commands. The operator ++ is used to construct a multiset of tokens. The operator ` is a binary operator in Snoopy. It takes a non-negative integer as a left operand that specifies the number of copies of the element provided as the right operand. The place A_{23} has thirteen tokens. Notice that the current distribution of tokens as shown here constitutes the initial marking M_0.

Algorithm 2.1: demonstrating the working of transitions in N1

while *true* **do**
; //i.e., true until there are commands for N2
perform t_1; //means to prepare the NCs
perform t_2; //sends the NC via channel ‘send’
perform t_3; //listen to acknowledgement from N2 on ‘ack’
perform t_4; // interpret the acknowledgement and act accordingly

Algorithm 2.2: demonstrating the working of transitions in N3

while A_{22}, A_{23}, *and* A_{24} *contain at least a single token* **do**
perform t_{33}; //observe landmarks
perform t_j, $28 \leq j \leq 32$; /* where j is chosen randomly such that the input token on t_j satisfies its guard expression. The transition t_j delivers information regarding observed difficulty level to t_{j+k-20} and t_{j+k-19}, where k=0, 1, 2, 3, 4, for successive increasing values of j. This step contributes in enabling t_{j+k-20} and t_{j+k-19} simultaneously.

moment of time, of the proposed GSPN model of route learning (Figure 2.4). Marking of a Petri net represents the state of the system being modelled. N1 and N2 are connected by two channels: the places `send` and `ack`. The former sends navigation commands from N1 to N2, and the latter receives acknowledgements from N2. N1 involves the places Tr_1 to Tr_4, `send`, and `ack`, and transitions t_1 to t_4. N1 forms a model to represent a trainer who has knowledge of a specific route and sends the navigation commands to the learner, N2, via the channel send and then waits on the learner to process the command before sending the next command. This process continues until there are no more commands left with the trainer, as explained in Algorithm 2.1. A token on N1’s place shows a state the trainer is in. For example, a token on Tr_1 means that the trainer has a command for the learner to follow.

The net N3 in the GSPN model (see Figure 2.4) is responsible for collecting the route information in terms of how difficult a landmark might be to remember by a person who visits it the first time. Here, landmarks are divided in five classes: (*i*) lowest

Algorithm 2.3: Landmark Classification Procedure (LCP)

Input: A route consisting of decision points, $p_1, p_2, \ldots, p_n$.
Output: γ, the level of difficulty to remember a landmark

for *integer* i:=1 to n **do**
  if p_i is crowded by many objects **then**
    $\gamma \leftarrow$HIGH
  else if *near* p_i *there are two similar landmarks leading to different locations* **then**
    $\gamma \leftarrow$MEDIUM
  else if *there is an easier route originating in close proximity to* p_i **then**
    $\gamma \leftarrow$HIGHEST /*here, the remaining part of the route is deliberately ignored in favor of re-routing.*/
  else if p_i *has a clear landmark nearby* **then**
    $\gamma \leftarrow$LOW
  else
    $\gamma \leftarrow$LOWEST /*no landmarks needed to move forward, the route here is so easy that the entire scene is remembered.*/
return γ

Algorithm 2.4: demonstrating the working of transitions in N2

while *true* **do**
  perform t_5; //understands the NCs from N1
  perform t_6; //run the NCs and begin learning surroundings
  perform t_7; // acknowledge N1 for executing the NC
  perform any one of t_8-t_{17}; //remember or forget NC with landmarks
  perform any one of t_{18}-t_{27}; /* depends on which of A_5-A_{14} gets token from the transitions t_{18}-t_{27}. The purpose is to enable transition t_7 so that the acknowledgement of executing the NCs can be sent to N1.*/

difficulty, (*ii*) low difficulty, (*iii*) medium difficulty, (*iv*) high difficulty, and (*v*) highest difficulty, as per chances that they serve as cues to recall required changes in one's heading. The transitions t_{28}, t_{29}, t_{30}, t_{31}, and t_{32} are enabled when the guard expressions on them are true. For example, t_{28} is enabled only when a medium difficulty level landmark is passed in the transition t_{33}. The net N3 works according to Algorithm 2.2. The GSPN model is 1-bounded. Therefore, if any of the transitions in the set t_{28}, t_{29}, t_{30}, t_{31}, t_{32} is enabled, all others in this set will be disabled. It should be noted that only N3 uses colored tokens; N1 and N2 use uncolored tokens. The datatype `NUM` is for numeric values starting from zero, and `NUMLEVELS` is

Table 2.1. The meaning or purpose of the transitions used in the proposed GSPN model. The index sets are I_1={8, 10, 12, 14, 16} and I_2={9, 11, 13, 15, 17}.

Transitions	Meaning/context
t_1	prepare or collect the navigation command.
t_2	send the command to N2 via channel `send`.
t_3	collect the acknowledgment of N2 via the channel `ack`.
t_4	define the next step after last acknowledgment.
t_5	read the navigation command.
t_6	interpret the command and standby on acknowledgment.
$t_i \in I_1$	read the landmark difficulty level, process it according to its value with the navigation command and do not save this information in memory.
$t_i \in I_2$	read the landmark difficulty level, process it according to its value with the navigation command and save this information in memory.
$t_{18\text{-}27}$	prepare for acknowledgment.
t_{33}	read landmark difficulty levels in order.
$t_{28\text{-}32}$	send the difficulty level according to matched guard expression
t_7	send the acknowledgment to N1 via channel `ack`.

derived from `NUM` and the enumerated type `D = {LOW, LOWEST, MEDIUM, HIGH, HIGHEST}`, which represents difficulty levels to remember landmarks. A variable of type `NUMLEVELS` is a tuple (pair) containing a value of type `NUM` as the first element, and an element of `D` as the second element. The purpose of integers in the first place is to bring an order in the tokens on the place A_{23} according to an input route. The values of the set `D` are used to show difficulty level associated with landmarks.

The classification of landmarks proposed in the preceding paragraph is justified because experiments with humans suggest that recognizing and remembering landmarks involve a host of factors, such as to avoid losing sight of familiar landmarks (Yurkiewicz & Tsao, 2012), being different or unique in color, layout, shape, or some other attribute associated with a particular location (Golledge, 1977). Most importantly, repeated or mirrored layouts, and crowded passages are the factors that are found to have adverse effects on human navigational abilities (McKinlay, 2016). On the other hand, the use of WiFi, wall-mounted antennas, and sensors, for local navigation gadgets in indoor spaces, as suggested by McKinlay (2016), may be unsuitable in emergency evacuation situations because of the emerging hazards, and highly time-critical situation. A method to classify landmarks as to how difficult they might be to remember is proposed here in Algorithm 2.3.

The first purpose of N2 is to generate the required actions to follow the input from N1. The second purpose is to either enter into a state of forgetting or the state of remembering the landmark and associated navigation command. Algorithm 2.4 explains the working of net N2. Note that Algorithms 2.1, 2.2, and 2.4 are described to explain the basic execution cycle in the GSPN model in terms of simple programming constructs, rather than directly implementing in program on a platform without the support of the concept behind stochastic Petri-nets. A computer program for implementing a transition, say t_9, would need to materialize the intention behind the transition, for example, fetching the landmark related information (such as, position, color, size, etc.), the navigation command, and then saving this data into a repository, which may be a simple data structure or a table in a relational database management system. Since forgetting and remembering are more dependent on time compared to other activities, such as fetching a navigation command from N1 (see Table 2.1), the transitions that deal with forgetting and remembering are modelled by stochastic timed transitions. For all other activities, immediate transitions are used, which correspond to vanishing states because the sojourn time of markings that enable the immediate transitions is not exponentially distributed. When enabled, the immediate transitions take zero time to fire. The stochastic transitions, (t_i, t_{i+1}), $i \in \{8, 10, 12, 14, 16\}$, are in conflict. If A_{16} has a token, it can only contribute in

Table 2.2. The stochastic transition rates of the GSPN model in Figure 2.4. The rates Q_1 and Q_2 are taken at random from the range α, whereas Q_3 and Q_4 are randomly selected from near the lower-end and the upper-end values in the range α, respectively.

Transitions (t_i)	Rate range (α)	Q_1	Q_2	Q_3	Q_4	γ
t_8	(0.2, 0.4]	0.39	0.39	0.25	0.395	MEDIUM
t_9	$1-\alpha_{i-1}$	0.61	0.61	0.75	0.605	MEDIUM
t_{10}	(0.05, 0.2]	0.11	0.12	0.08	0.20	LOW
t_{11}	$1-\alpha_{i-1}$	0.89	0.88	0.92	0.80	LOW
t_{12}	(0.4, 0.6]	0.49	0.46	0.41	0.55	HIGH
t_{13}	$1-\alpha_{i-1}$	0.50	0.53	0.59	0.45	HIGH
t_{14}	(0.6, 1.0]	0.79	0.73	0.65	0.85	HIGHEST
t_{15}	$1-\alpha_{i-1}$	0.20	0.26	0.35	0.15	HIGHEST
t_{16}	(0.0, 0.05]	0.04	0.005	0.001	0.05	LOWEST
t_{17}	$1-\alpha_{i-1}$	0.96	0.99	0.999	0.95	LOWEST

enabling t_8 and t_9. Similarly, A_{17} can only enable t_{10}, t_{11}, and so on. Finally, a token on A_{20} may only enable t_{16} and t_{17}. Once a pair of transitions are in conflict, anyone may fire depending on exponential firing delay. Thus, firing of t_8, which deals with the medium difficulty landmarks, means that the model will not retain the navigation command and observed landmark, whereas, t_9, which also deals with the medium difficulty landmarks, calls for saving the navigation command and the observed landmark. In the same way, the transitions t_{10}-t_{11}, t_{12}-t_{13}, t_{14}-t_{15}, and t_{16}-t_{17} deal, respectively, with low difficulty, high difficulty, highest difficulty, and lowest difficulty levels of forgetting/remembering a navigation command and associated landmarks. The first in each pair models forgetting and the second remembering.

2.4.1 Model description

The model presented in Figure 2.4 is a landmark-based route learning model for agents. The model has thirty places, twenty-three immediate, and ten stochastic transitions. The temporal nature of the model is specified by the use of probabilistic firing delays in stochastic transitions. The stochastic transitions use negative exponential distributions with rates, α_t, assigned randomly from the ranges defined in Table 2.2. The rate ranges are selected so that: (*i*) at the lowest difficulty possible, the rate of forgetting is at the minimum, (*ii*) at the highest difficulty possible, the rate of remembering is at the minimum. The boundaries of the rate ranges are defined arbitrarily to produce the results close to empirical values. The arcs used in the model are all standard arcs. An arc without an explicit arc-expression means that the default expression, i.e., `1`dot` is used.

As with any mathematical modeling exercise, there is a certain level of abstraction involved here. All places except A_{21}, A_{22}, and A_{23} are of `Dot` type. For any variant of colored Petri nets, `Dot` is a simple type that does not associate any value with it. Thus the tokens used with the places Tr_1-Tr_4, `send`, `ack`, A_1-A_{20}, and A_{24} do not carry any value. A navigation command is assumed at Tr_1, for instance, whenever there is a token (of type `Dot`) present at Tr_1. For simplicity, values of navigation commands, such as *move left*, *move right* are not used, rather only the flow of the commands is considered. Similarly, when there is a token (of type `Dot`) at `ack` it would mean that the previous navigation command has been used and now is the time to pass on the next command. A token on a place primarily represents the state the agent is in. If, for example, in some application there is an agent based only on N1 then that agent can only send navigation commands to another agent that is supposed to act accordingly. This forms a basic trainer-learner approach, which simulates the situations when the trainer agent is separate from the learner agent. On the other hand, an agent that knows the route information, say by visiting the environment long time ago, may act as a trainer as well as a learner at the same time. In this case, the agent would have a portion of its mind representing N1 and another portion acting as N2 and N3. This reflects situations when an individual agent traversing the route would tell itself about where to go when it observes a particular landmark.

The tokens used with the place `A22` are of integer type. These are primarily used for counting the cycles and sending the right route information to the net from the place `A23`. The tokens used with the places `A21` and `A23` are of type `NUMLEVELS`, which is a compound type as mentioned in this section before. The identities of the landmarks (see Section 2.5.1) have not been made part of the declaration of `NUMLEVELS` in order to reduce the complexity of the model.

2.5 Simulating Smith's experiment

J. Smith (2015) performed an experiment to assess the utility of VE training on learning and competence of human participants during emergency egress situations on offshore petroleum platform. The experiment involved 36 participants that formed

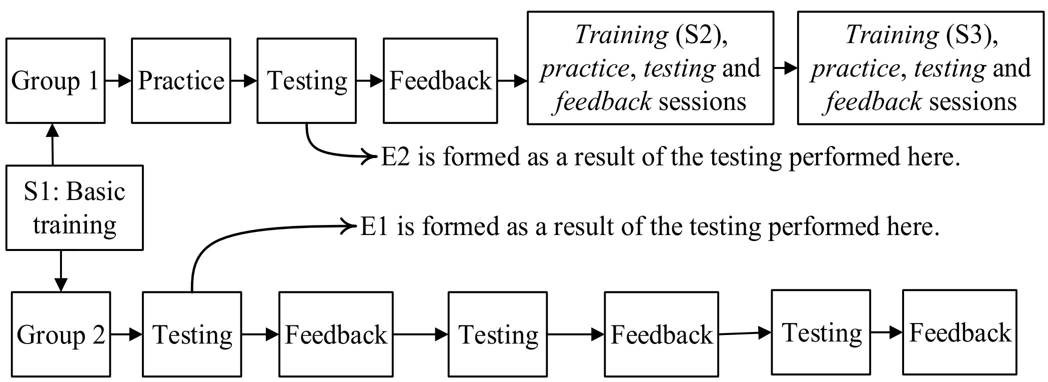
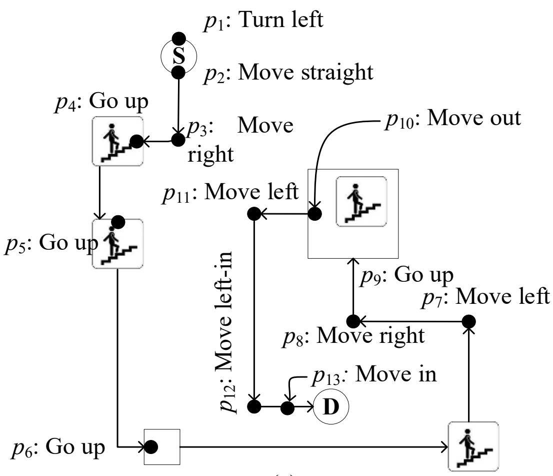


Figure 2.5. Training exposure to participants *Source: Adopted from* (J. Smith, 2015).

two groups: Group 1 containing 17 and Group 2 containing 19 participants. The VE used in this study was All-hands Virtual Emergency Response Trainer (AVERT). AVERT is a simulator of an offshore oil and gas facility that comprises several decks. It is used to train participants for better response in emergency situations, such as fires and explosions. Figure 2.5 shows the difference in training the participants of both groups received. Group 1 was given repeated exposure to training and Group 2 was trained once. The practice sessions were interactive. The testing sessions included interactive tasks that were performed in AVERT, followed by a quiz. These testing sessions were recorded as replay video files so they can be watched later on the simulator. The training content in the experiment targeted six learning objectives: (1) establish spatial awareness of the environment, (2) routes and mapping, (3) alarm recognition, (4) continually assess situation and avoid hazards on route, (5) register at temporary safe refuge, and (6) general safe practices (J. Smith, 2015 p. 59-60). The present study considers only the first two of these learning objectives because they are directly related with route learning activity, and also because this keeps the complexity of the model low. The participants were instructed to behave as if it was their first day of training for a job on an offshore platform. An important part of the training was how to correctly respond to an emergency. The learning materials on the basic training included AVERT Platform Orientation with a platform video tour, and lecture style instructions on Keeping a Safe Workplace and Responding to Emergencies. A 30-minute platform exploration was also performed by each participant. They also watched five route videos highlighting important primary and secondary escape routes. Two of the routes were from a cabin in the accommodation

Table 2.3. Difficulty level, γ, associated with each decision points in R1.

Decision point (p_i)	γ_{R1}
p_1	HIGH
p_2	LOWEST
p_3	LOWEST
p_4	LOWEST
p_5	LOWEST
p_6	LOW
p_7	MEDIUM
p_8	HIGHEST
p_9	LOWEST
p_{10}	LOWEST
p_{11}	LOWEST
p_{12}	LOWEST
p_{13}	LOW



(a)

Decision points	Nearby landmarks (landmarks' identities)
Near p_1	are a number of objects, here and there, leading to different locations
p_2	Some pipes
p_3	A steel railing
p_4	Equipment stairs # 1
p_5	Stairwell ending at an arch
p_6	Small stairwell
p_7	Two doors on opposite ends of a corridor
p_8	A fire exit door with a warning sign
p_9	Stairwell with a door on just left.
p_{10}	A fire exit door
p_{11}	A blue colored pillar
p_{12}	Lifeboat station
p_{13}	Mess hall door

(b)

Figure 2.6. (a) The route, R1, from worksite (S) in the engine room to the primary muster station (D). Decision points, p_i, $1 \leq i \leq 13$, with required navigation commands are shown. The route is scaled for better readability. (b) Description of landmarks at the decision points.

block of the platform, and three routes were from a worksite in the engine room. The route videos played a key role in the participants' learning of the designated escape routes. The routes required a participant to remember his/her cabin, worksite (in the engine room), and primary (mess hall) and secondary (lifeboat) muster stations. After the training, the participants were tested to see how they egressed to their muster point, and competence was assessed. Both groups received the basic training.

2.5.1 The primary escape route

The primary escape route from the worksite in the engine room to the muster station (the mess hall) is depicted in Figure 2.6. This route is named here as R1. There were two other escape routes from the worksite, but only a small fraction of the participants followed those routes during testing scenarios. R1 is composed of a list of ordered pairs having a navigation command as the first component and a decision point or a landmark as the second. This information is collected by carefully watching the route videos that were used for training and collecting the salient features of the route. It turns out that there are 13 decision points, $p_1, p_2, \ldots, p_{13}$ that make up R1.

Algorithm 2.3 is used to determine the difficulty levels, γ_{R1}, associated with the landmarks near the decision points in R1, and the results are reported in Table 2.3. These values are used as input to the place $\mathtt{A}_{23}$ of the GSPN model. The choice of exactly how many landmarks should be used in condition#1 on line#2 in Algorithm 2.3 so as to make it `true` is critical and application dependent. In Smith's case study, there were around twelve possible routes at p_1 and the location lies within a big hall that had more than forty objects (including four stairwells, two service generators, many pillars and railings, ten canisters, and other machinery relevant to an engine room) cluttered around. This extraneous information, which was provided in as little time as around 37 seconds (in training and testing scenarios), suggested the difficulty level for p_1 to be `HIGH`. For condition# 2 on line# 4 there were exactly two similar landmarks at the opposite ends of a long corridor near p_7. The corridor had symmetric appearance on both ends. An easier detour was found near p_8. This detour uses the central stairwell to reach a muster station. This condition is tested in Algorithm 2.3 on line#6. The decision point p_6 had a clear landmark (condition#4 on line #8), which was an arch near a stairwell. Similarly, p_{13} also satisfied condition#4 on line#8 because of the presence of a red colored mess hall door situated opposite to starboard side lifeboat station. The difficulty levels associated with p_6 and p_{13} were set to `LOW`. The remaining decision points fell into the category of `LOWEST` difficulty level.

2.5.2 Datasets for model validation

2.5.2.1 EMPIRICAL DATASET (E1)

The empirical dataset E1 comprises the average number of wrong decisions (% values) per decision point obtained when the participants of Group 2 were tested for a scenario that asked to muster at the primary muster station from the worksite at the decision point p_1. The results are shown in Figure 2.7. During the testing scenario, 47% of the participants failed to produce the correct path from p_1. The point p_6 had a clear landmark, which was an arch near a stairwell. The 12 participants who reached p_6 correctly followed this decision point except one who made a mistake here. The decision point p_7 was a door at the end of a stairwell, opening out into a corridor having a similar look on both sides. The doors at the opposite ends of the corridor were similar. These doors were the potential landmarks because they were different than the other doors in terms of their orientation and signage, but they were two, and were identical in color and layout. Their signages were also exactly the same. The right and the left side portions of the corridor, as viewed from p_7, were perfect examples of a mirrored layout, which resulted in around 26% of those who arrived here selecting the wrong direction. As for p_8, there was a clear landmark, which was a single fire exit door. However, 71% of those who reached p_8 made a detour from a location just after p_7, and thereby left the route R1. We consider it a deliberate action because of the possibility of an easy alternative to the destined muster station, which was the platform's mess hall. We think that the participants considered the remaining part of R1, or at the very least just the landmark near p_8, as difficult compared to their chosen alternative route, which ran through the central stairwell. Therefore, p_8 witnessed 71% failure in choosing the right direction. The point p_{13} was reached by all the 19 participants except one who reverted back to some other location. The rest of the decision points were successfully followed by all the participants who reached there. The average percentage of wrong decisions, $e(p_i)$, per decision point, p_i, committed by the participants was estimated by counting how many participants made a wrong choice at an ith decision point provided the participant arrived at the decision point, and dividing the result with the total number of participants.

2.5.2.2 Empirical dataset (E2)

The participants of Group 1, after receiving the basic training, were given five practice sessions before the first testing scenario of this study. This testing scenario (TE) was related to egressing from worksite to mess hall (called TE2) in the event of an emergency. The participants were assessed on their familiarity with the escape routes.

As with the E1, only the primary escape route is considered here. The other testing and practice scenarios as depicted in Figure 2.5 were developed to assess the learning objectives (3)-(6) (see Section 2.5) and are, therefore, not considered here due to scope of the present study. The $e(p_i)$s were computed by watching participants' performance in replay videos. These results are depicted in Figure 2.7.

2.5.3 Simulation and results

The proposed route learning model is developed in the software environment Snoopy (Heiner, Herajy, Liu, Rohr, & Schwarick, 2012). The simulation interval was set to a start and an end point at 0 and 100, respectively. The initial marking was set to M_0, which is shown as a distribution of tokens over the entire set of places in Figure 2.4. Usually, stochastic simulations require a significant number of simulation runs (Marwan, Rohr, & Heiner, 2012). We use repeated stochastic simulations by setting the number of runs at 500 for samples S1 and S3, and at 5000 for samples S2, S4, S5,

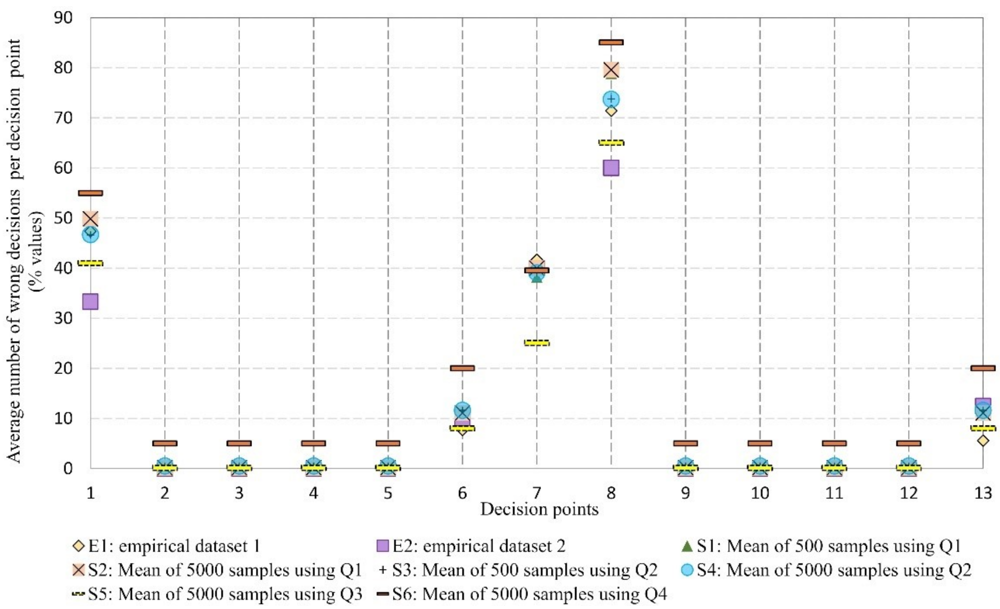


Figure 2.7. Empirical and simulation results.

and S6. Each sample is then averaged. As an example, consider the case of S2. The simulation output generates 5000 series each simulating the decision points (followed or forgotten) in R1. The average of these series is a more reliable estimate of learning compared to individual series.

As mentioned earlier, the outputs are simulated routes corresponding to the route R1, with some missing decision points showing the phenomenon of forgetting. Figure 2.7 reports six samples obtained by setting up the model for route R1. The average percent wrong decisions (errors) per decision point, $e(p_i)$, $1 \leq i \leq 13$, obtained in empirical datasets E1 and E2 are shown for comparison with the simulated results. Sample 1 (S1) is the average percentage of wrong decisions per decision point for 500 simulations, which can be considered as the average performance of 500 agents. The deviations from the observed data is clearly visible. Sample 2 (S2) is the average obtained by running 5000 simulations. S1 and S2 are obtained when the rates for the transitions t_8-t_{17} are set according to the values of the set Q_1 (see Table 2.2). The samples S3 and S4 are computed for 500 and 5000 simulations with a different set of values, Q_2, for the transition rates. The transition rates, Q_1 and $Q2$, are randomly chosen, whereas, Q_3 and Q_4 are taken close to the minimum and the maximum values in the transition rates described in Table 2.2. The paired two-tailed *t*-tests for S1, S2, S3, and S4, with the empirical series E1 accept, with 95% confidence, the null hypothesis that these samples have, statistically, the same mean as that of the E1. It is clearly seen in Figure 2.7 that S5 and S6, which were obtained when the transition rates were set to Q_3 and Q_4, respectively, make lower and upper bounds of the GSPN model for the failure to remember navigation command at a decision point. This means that depending on what values of rates are chosen for stochastic transition in the model, the model output cannot go beyond these limits unless the ranges defined in Table 2.2, for Q_3 and Q_4, are modified. Now, comparing the simulated results with the series E2, it is clearly seen that the model output deviates significantly where the difficulty level is `HIGH` or above, i.e., at points p_1 and p_8. This is because the participants were not naïve because they had been through to a practice session (before testing) that had improved their spatial skills. To address sequential learning, the proposed model may be used in a repeated manner each time producing an epoch of learned route, and in each iteration the rates of the stochastic transitions may be modified in some (say linear) manner to produce better results.

2.5.4 Analysis of the proposed model

Marcie model checker (Heiner, Rohr, & Schwarick, 2013) is used to analyze various mathematical properties of the GSPN model in Figure 2.4. The model has a finite reachability graph, containing 312 states and 481 edges with zero absorbing state, and is *live* and *reversible*. It is 1-bounded, and therefore, safe (Bause & Kritzinger, 1996).

2.5.5 Performance evaluation

The CTMC associated with the GSPN model is derived below. If $\mathcal{L}$ denotes this CTMC, then the steady state distribution (Ajmone Marsan, 1990) representing that $\mathcal{L}$ in state i at time t is given by

$$\pi(t) = \Pr\{\mathcal{L}(t) = i\}, \tag{2.5}$$

where, $\pi(t) = (\pi_1(t), \pi_2(t), \dots,)$.

Table 2.4. States in the CTMC of the GSPN model in Figure 2.4, obtained for the sample S4. States containing the places A_5–A_{14} are shown for brevity. The actual markings involved unfolding (of the GSPN) due to the colored subnet N3. For clarity only A_{22} was replaced by its corresponding uncolored place in the markings below. The rest of the places are shown as they appeared in Figure 2.4.

States (M_j)	Markings	$\pi_j \times 10^{-4}$
M_0	$Tr_2, A_1, A_{24}, A_{22-1}, A_{23}$	5.94
M_{292}	$Tr_3, A_5, A_3, A_{22-8}, A_{23}$	37.16
M_{306}	$Tr_3, A_6, A_3, A_{22-8}, A_{23}$	57.95
M_{307}	$Tr_3, A_7, A_3, A_{22-1}, A_{23}$	11.04
M_{308}	$Tr_3, A_7, A_3, A_{22-7}, A_{23}$	11.04
M_{309}	$Tr_3, A_8, A_3, A_{22-1}, A_{23}$	84.07
M_{310}	$Tr_3, A_8, A_3, A_{22-7}, A_{23}$	84.07
M_{311}	$Tr_3, A_9, A_3, A_{22-2}, A_{23}$	44.42
M_{117}	$Tr_3, A_3, A_{10}, A_{22-2}, A_{23}$	50.70
M_{274}	$Tr_3, A_{11}, A_3, A_{22-9}, A_{23}$	70.11
M_{275}	$Tr_3, A_{12}, A_3, A_{22-9}, A_{23}$	25.01
M_{276}	$Tr_3, A_{13}, A_3, A_{22-13}, A_{23}$	0.48
M_{277}	$Tr_3, A_{13}, A_3, A_{22-12}, A_{23}$	0.48
M_{278}	$Tr_3, A_{13}, A_3, A_{22-11}, A_{23}$	0.48
M_{279}	$Tr_3, A_{13}, A_3, A_{22-10}, A_{23}$	0.48
M_{280}	$Tr_3, A_{13}, A_3, A_{22-6}, A_{23}$	0.48
M_{281}	$Tr_3, A_{13}, A_3, A_{22-5}, A_{23}$	0.48
M_{282}	$Tr_3, A_{13}, A_3, A_{22-4}, A_{23}$	0.48
M_{283}	$Tr_3, A_{13}, A_3, A_{22-3}, A_{23}$	0.48
M_{284}	$Tr_3, A_{14}, A_3, A_{22-13}, A_{23}$	94.64
M_{285}	$Tr_3, A_{14}, A_3, A_{22-12}, A_{23}$	94.64
M_{286}	$Tr_3, A_{14}, A_3, A_{22-11}, A_{23}$	94.64
M_{287}	$Tr_3, A_{14}, A_3, A_{22-10}, A_{23}$	94.64
M_{288}	$Tr_3, A_{14}, A_3, A_{22-6}, A_{23}$	94.64
M_{289}	$Tr_3, A_{14}, A_3, A_{22-5}, A_{23}$	94.64
M_{290}	$Tr_3, A_{14}, A_3, A_{22-4}, A_{23}$	94.64
M_{291}	$Tr_3, A_{14}, A_3, A_{22-3}, A_{23}$	94.64

The steady state probability distribution for $\mathcal{L}$ exists and is reported in Table 2.4. The steady state distribution vector $\pi = (\pi_1, \pi_2, \pi_3, \ldots,)$ is obtained by solving the following system of linear equations (Mo, 2013):

$$q_i \pi_i = \sum_{j \neq i} q_{ji} \pi_j \,, \tag{2.6}$$

where, $q_i = \sum_{j \neq i} q_{ij}$, and $Q = (q_{ij})$ is the so-called infinitesimal generator, i.e., the matrix of transitions rates (see Table 2.2 for the values used here). Since π is a steady state distribution vector, it satisfies:

$$\sum_{\forall j} \pi_j = 1\,, \tag{2.7}$$

Now, let $G(.)$ be a performance function that determines the average value of learned navigation commands and landmarks (or decision points). Using the vector π the expected value of G can be estimated as (Ajmone Marsan, 1990; Mo, 2013):

$$E[G(X)] = \sum\nolimits_{j \in \mathbb{S}} G(j).\pi_j, \tag{2.8}$$

where, $\mathbb{S}$ is the state-space.

For the case study of Section 2.5, let X represent the state that involves remembering the navigation commands and associated landmarks, viz., the states involving the places A_6, A_8, A_{10}, A_{12} and A_{14}, then $G(X)$ represents that the state X is being reached or the process is in state X (in terms of the GSPN model, that the marking M_X is reached and the desired place has n = 1 token). Thus, for a state m where learning is not involved, $G(m)$ is considered as 0, otherwise 1. It can be seen, using Table 2.4, that

$$E[G(X)] = \sum\nolimits_{j \in \mathcal{A}} 1.\pi_j = 1059, \tag{2.9}$$

where, $\mathcal{A}$={M_{306}, M_{309}, M_{310}, M_{117}, M_{275}, M_{284} - M_{291}}. The above equation shows the average fraction of time the model stays in states responsible for learning. It follows that if Y corresponds to the states where the model does not remember navigation commands and landmarks,

$$E[G(Y)] = \sum_{j \in \mathcal{B}} 1.\pi_j = 178, \tag{2.10}$$

where, $\mathcal{B}$={M_{292}, M_{307}, M_{308}, M_{311}, M_{274}, M_{276} - M_{283}}. Thus, the total fraction of time the system is in learning state is 85.6%. The sums in Equations (2.9) and (2.10) are for the case of S4. The average learning time for the empirical data reported previously, is estimated using $\sum_{1 \le i \le n}(100 - e(p_i))/n$, which is found to be 86.6%. That is, the average percentage of learning route R1 is 86.6%, which is very close to the value obtained for S4. Equations (2.9) and (2.10) are also used to estimate the values corresponding to the samples S5 and S6. The percentages of time that the model spends in learning are found to be 88.6% for S3 and 80.0% for S4. This means that the lower the rates of stochastic transitions, t_8, t_{10}, t_{12}, t_{14}, and t_{16}, the greater is the average learning.

2.6 Conclusion

The proposed GSPN model for route learning in a new environment is validated by using Smith's experiment for the case of escape routes on an offshore petroleum platform (J. Smith, 2015). The case study clearly demonstrates estimation of the difficulty level associated with a variety of decision points. The results of paired t-tests confirm (with 95% confidence) that the simulations generated similar results as obtained with human participants. Equation (2.9) uses steady state probabilities to determine how long the model stays in those states where the transitions (see Figure

2.4 and Table 2.4) for saving the navigation commands and respective landmarks are executed. This estimates how much learning is done when compared to the results of Equation (2.10), which determines the proportion of time the model stays in states responsible for *not saving* the landmarks and associated navigation commands.

A software agent, based on the GSPN model, would provide an interesting venue for trainees to see the effects of forgetting a location in a route during egress in an emergency evacuation scenario in a VE (Dooley, 2017; Hayden, 2015). The model may be used, as an application of machine learning, to produce *student models* in an intelligent virtual training system for emergency evacuations (Gilmore & Self, 1988). Other applications of this model may include computer games in which a software agent uses the GSPN model to learn routes based on its continuous exposure to the game world. Different agents can show different ability to learn routes by using different sets of firing rates of the transitions t_8-t_{17} and thereby will demonstrate different strength in opposing or favoring the human player. The quantification of landmarks and their impact on learning, as presented here, is an important result in the sense of better situation awareness. The case studies suggest that an *i*th landmark along a route should be treated as if it is a temporary destination before the final destination point. Thus, every other landmark that is situated before the *i*th landmark should be of reasonable difficulty so that it can be easily remembered by people so there is no need to detour, as this may delay the evacuation process. Mirrored layouts should be avoided, and the emergency egress routes should have clear signage at every decision-point so that people do not need to remember other objects in the environment for finding the right direction. It turns out that escape routes are not used during normal operating hours, especially those that use ladders to egress. Workers should be trained regularly so that their navigation skills remain dependent on designated signage system rather than relying on ordinary or natural environmental cues. The GSPN model should be a good choice for software or intelligent agents in a virtual environment where access to the environment's geometry would make it easy to recognize and classify landmarks to be used as input to the model. Algorithm 2.3 defines, in a broader sense, what needs to be done for landmark classification. A consensus on landmark classification based on exact numbers of nearby objects, or shape, color or size of an object is still an open problem and the authors have found only a few articles discussing this problem in a broader sense.

As mentioned before, the model takes only a single exposure of training. Future work, therefore, should enhance the capability to add multiple exposures of training so that after a certain number of iterations the model output is free from any missing knowledge. This may involve a systematic and iterative adjustment in the firing rates of the stochastic transitions. Still another dimension of work is to add support for learning multiple routes and then integrate them all to produce survey knowledge of the environment. Because of the presence of multiple routes between the same source and destination points, this strand of work may exploit naturalistic decision making (G. Klein, 2004) to model how an agent selects routes from a set of learned choices in a moment of high stress.

Acknowledgments

The authors acknowledge with gratitude the support of the NSERC-Husky Energy Industrial Research Chair in Safety at Sea, and the Canada Research Chair Program.

References

Ajmone Marsan, M. (1990). *Stochastic Petri Nets*. Springer.

Ajmone Marsan, M., Conte, G., & Balbo, G. (1984). A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Trans. Comput. Syst.*, *2*(2), 93–122. https://doi.org/10.1145/190.191

Allen, G. L. (1999). Spatial abilities, cognitive maps and wayfinding: bases for individual differences in spatial cognition and behavior. In R. G. Golledge (Ed.), *Wayfinding behavior* (pp. 46–80). The John Hopkins University Press.

Aziz, A. (2000). Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, *1*(1), 162–170.

Balbo, G., Chiola, G., & Bruell, S. C. (1992). An example of modeling and evaluation of a concurrent program using colored stochastic Petri nets: Lamport's fast mutual exclusion algorithm. *IEEE Transactions on Parallel and Distributed Systems*, *3*(2), 221–240.

Bause, F., & Kritzinger, P. S. (1996). *Stochastic Petri Nets: An Introduction to the Theory*. Verlag Viewweg.

Beusmans, J. M., Aginsky, V., Harris, C. L., & Rensink, R. A. (1995). Analyzing situation awareness during wayfinding in a driving simulator. In D. J. Garland & M. R. Endsley (Eds.), *Proceedings of the International Conference on Experimental Analysis and Measurement of Situation Awareness* (pp. 245–251). Daytona Beach, Florida: Embry-Riddle Aeronautical University Press.

Borowiec, S., & Lien, T. (2016, March). *AlphaGo beats human Go champ in milestone for artificial intelligence*. Retrieved from https://www.latimes.com/world/asia/la-fg-korea-alphago-20160312-story.html

Buckland, M. (2004). *Programming Game AI by Example*. Jones & Bartlett Learning.

Derdikman, D., & Moser, E. I. (2010). A manifold of spatial maps in the brain. *Trends in Cognitive Science*, *14*(12).

Dooley, B. J. (2017). *Why AI with augmented and virtual reality will be the next big thing*. Upside, 4th April.

Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. MIT Press.

Eilam, D. (2014). Of mice and men: building blocks in cognitive mapping. *Neuroscience and Biobehavioral Reviews*, *47*, 393–409.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32–64.

Febbraro, A. Di, Giglio, D., & Sacco, N. (2016). A deterministic and stochastic Petri net model for traffic-responsive signaling control in urban areas. *IEEE Transactions on Intelligent Transportation Systems*, *17*(2), 510–524.

Gale, N., Golledge, R. G., Pellegrino, J. W., & Doherty, S. (1990). The acquisition and integration of route knowledge in an unfamiliar neighborhood. *Journal of Environmental Psychology*, *10*, 3–25.

Ghahramani, S. (2005). *Fundamentals of Probability with Stochastic Processes* (3rd ed.). Pearson Prentice Hall.

Gilmore, D., & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self (Ed.), *Artificial intelligence and human learning: intelligent computer-aided instruction*.

Golledge, R. G. (1977). Multidimensional analysis and environmental behavior and design. In I. Altman & J. F. Wohlwill (Eds.), *Human behavior and environment: advances in theory and research*. Heidelberg: Springer Berlin Heidelberg.

Golledge, R. G. (1991). Cognition of physical and built environment. In T. Garling & G. W. Evans (Eds.), *Environment, Cognition, and Action: An Integrated Approach*. New York, NY: Oxford University Press.

Golledge, R. G. (1993). Geographical perspectives on spatial cognition. In T. Garling & R. G. Golledge (Eds.), *Behavior and Environment: Psychological and Geographical Approaches* (pp. 16–46). Amsterdam: Elsevier Science Publishers B. V.

Golledge, R. G. (1999). Human wayfinding and cognitive maps. In R. G. Golledge (Ed.), *Wayfinding behavior* (p. 9). Baltimore: The Johns Hopkins University Press.

Gorton, I. (1993). Parallel program design using high-level Petri nets. *Concurency and Computation: Practice and Experience*, *5*(2), 87–104.

Grush, R. (2000). Self, world and space: the meaning and mechanisms of ego- and allocentric spatial representation. *Brain and Mind*, *1*, 59–92.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on System Science and Cybernetics*, *4*(2), 100–107.

Hayden, S. (2015). *NASA is creating a virtual reality mission to MARS, "The Mars 2030 experience."* Retrieved from https://www.roadtovr.com/nasa-creating-virtual-reality-mission-mars-mars-2030-experience/

Heiner, M., Herajy, M., Liu, F., Rohr, C., & Schwarick, M. (2012). Snoopy -- a unifying Petri net tool. *PETRI NETS 2012, June 2012, Lecture Notes in Computer Science*, *7347*, 398–407. Springer-Verlag.

Heiner, M., Rohr, C., & Schwarick, M. (2013). *MARCIE - Model checking And Reachability analysis done effiCIEntly*. 389–399. Springer-Verlag.

Jensen, K. (1981). Coloured Petri nets and the invariant-method. *Theoretical Computer Science*, *14*, 317–336. https://doi.org/https://doi.org/10.1016/0304-3975(81)90049-9

Jensen, K. (1996). *Colored Petri Nets* (3rd ed.). Springer.

Kang, S.-J., Kim, Y., & Kim, C.-H. (2010). Live path: adaptive agent navigation in the interactive virtual world. *The Visual Computer*, *26*, 467–476.

Klein, G. (2004). *The Power of Intuition*. Doubleday.

Kyritsis, M., Gulliver, S. R., & Morar, S. (2014). Cognitive and environmental factors influencing the process of spatial knowledge acquisition within virtual reality environments. *International Journal of Artificial Life Research*, *4*(1).

Lehtonen, E., Sahlberg, H., Rovamo, E., & Summala, H. (2017). Learning game for training child bicyclists' situation awareness. *Accident Analysis and Prevention*, *105*, 72–83. https://doi.org/http://dx.doi.org/10.1016/j.aap.2016.07.036

Li, L., & Yokota, H. (2009). Application of Petri nets in bone remodeling. *Gene Regulation and System Biology*, *3*, 105–114.

Lynch, K. (1960). *The Image of the City*. Cambridge, MA: MIT Press.

Maciel, P. R. M., Trivedi, K. S., Matias, R., & Kim, D. S. (2011). Dependability Modelling. In *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions* (pp. 53–97). IGI Global.

Marwan, W., Rohr, C., & Heiner, M. (2012). Petri Nets in Snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks. In J. van Helden, A. Toussaint, & D. Thieffry (Eds.), *Bacterial Molecular Networks. Methods in Molecular Biology (Methods and Protocols)* (Vol. 804). Springer.

McKinlay, R. (2016). Technology: Use or lose our navigation skills. *Nature*, *531*(7596), 573–575.

Mo, J. (2013). Performance modeling of communication networks with Markov chains. In J. Walrand (Ed.), *Syntheis Lectures on Communication Networks # 5*. Morgan & Claypool Publishers.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, *77*(4), 541–580.

Musharraf, M., Khan, F., Veitch, B., McKonnin, S., & Imtiaz, S. (2013). Human reliability assessment during offshore emergency conditions. *Safety Science*, *59*, 19–27.

Norazahar, N., Khan, F., Veitch, B., & MacKinnon, S. (2016). Prioritizing safety critical human and organizational factors of EER systems of offshore installations in a harsh environment. *Safety Science*.

Peterson, J. L. (1977). Petri nets. *ACM Computing Surveys (CSUR)*, *9*(3), 223–252.

Petri, C. A. (1966). *Kommunikation mit Automaten. Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2; (English translation)* (Vol. 1). Vol. 1. New York.

Plank, M., Snider, J., Kaestner, E., Halgren, E., & Poizner, H. (2014). Neurocognitive stages of spatial cognitive mapping measured during free exploration of a large-scale virtual environment. *Journal of Neurophysiology*, *113*, 740–753.

Ramchurn, S. D., Huynh, T. D., Ikuno, Y., Flann, J., Wu, F., Moreau, L., … Roberts, S. J. (2016). A disaster response system based on human-agent collectives. *Journal of Artificial Intelligence Research*, *57*, 661–708.

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Smith, J. (2015). *The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios [Master thesis]*. Memorial University of Newfoundland, St. John's, NL, Canada. Retrieved from: https://research.library.mun.ca/8401/1/thesis.pdf

Sud, A., Andersen, E., Curtis, S., Lin, M., & Manocha, D. (2007). Real-time path planning for virtual agents in dynamic environments. *Proceedings of IEEE Virtual Reality*, 91–98. https://doi.org/10.1109/VR.2007.352468

Tolman, E. C. (1948). Cognitive maps in rats and men. *The Psychological Review*, *55*(4).

Trivedi, K. S. (2002). *Probability and Statistics with Reliability, Queuing, and Computer Science Applications* (2nd ed.). Wiley Interscience Publication.

Wang, R. F., & Spelke, E. S. (2002). Human spatial representation: insights from animals. *Trends in Cognitive Sciences*, *6*, 376–382.

Yurkiewicz, I. R., & Tsao, J. W. (2012). Book Review: Why people get lost: the psychology and neuroscience of spatial cognition by Paul A. Dudchenko. Journal of the Neurological Sciences, 313(1–2), 197–198.

Chapter 3

Human-like sequential learning of escape routes for virtual reality agents***

Co-authorship statement. A version of this chapter has appeared as an article in the journal titled *Fire Technology published by Springer-Verlag*. The lead author, Syed Nasir Danial, has developed and implement the model and extracted the empirical data using re-play video files for validation of the model. The co-author Jennifer Smith performed the experiment and verified the data extracted from the experiment. Co-authors Dr. Faisal Khan and Dr. Brian Veitch supervised the study. All authors read and approved the final draft.

Abstract. The Piper Alpha disaster (1988) witnessed 167 casualties. The offshore safety guidelines developed afterward highlighted the need for effective and regular training to overcome the problems in evacuation procedures. Today, virtual environments are effective training platforms due to high-end audio/visual and interactive capabilities. These virtual environments exploit agents with human-like steering capabilities, but with limited or no capacity to learn routes. This work proposes a sequential route learning methodology for agents that resembles the way people learn routes. The methodology developed here exploits a generalized stochastic Petri-net based route learning model iteratively. The simulated results are compared with the route learning strategies of human participants. The data on human participants were collected by the authors from an earlier study in a virtual environment. The main contribution lies in modeling people's route learning behavior over the course of successive exposures. It is found that the proposed methodology models human-like sequential route learning if there are no easy detours from the original escape route. Although the model does not accurately capture individual learning strategies for all decision nodes, it can be used as a model of compliant, rule-following training guides for a virtual environment.

*** Danial, S. N., Smith, J., Khan, F., & Veitch, B. (2019). Human-Like Sequential Learning of Escape Routes for Virtual Reality Agents. *Fire Technology*, *55*(3), 1057–1083. https://doi.org/10.1007/s10694-019-00819-7.

3.1 Introduction

Effective wayfinding and mustering during an emergency are critical for managing the safe evacuation of an offshore petroleum platform. Ordering the full evacuation of a platform is a time-sensitive decision and requires all personnel onboard to be accounted for at the temporary safe refuge area. Conventional muster drills provide offshore personnel with repeated weekly exposure to their escape routes and muster procedures (e.g., knowing what to do in the event of an emergency). These drills are performed in benign conditions and as a result, may not provide crews with sufficient exposure to the offshore platform to develop survey knowledge (a mental map-like representation of the platform). The training is especially important for crew members who are working in an unfamiliar, or new work environment.

Virtual environments (VEs) provide a means to practice escape routes virtually, which enables people to learn the platform before they step foot on the real platform. Certainly, VEs provide easy access to training where a required number of training exposures can be given to participants at the expense of fidelity of the real environment. VE training can help prepare people to navigate the platform safely in the event of an emergency. VEs also provide a virtual lab setting to investigate human behavior while participants learn emergency escape routes. Artificially intelligent (AI) agents can be used in the VE to improve the fidelity of VE training. These agents can be developed with a range of behaviors, from people who are fully compliant with safety protocols, to people who have poor situation awareness and panic in emergency situations. Agents can be programmed to exhibit these behaviors. In general, agents have limited capabilities to express human cognitive behavior, such as the strategy a person would take to learn their escape route. This paper proposes a method that can be used to develop a sequential learning strategy of escape routes for agents. The approach is validated on human performance data collected over a series of training sessions in a VE.

Sequential route learning is a natural way to learn a route over some exposures to the route. In each exposure, information about locations along a route that requires a person to change their course is remembered due to the presence of exit signs, environmental cues, or landmarks (Allen, 1999). Such locations are commonly referred to in the literature as decision-points. The signage considered here does not only comprise the conventional exit symbols, but environmental cues are also coupled to make the location more salient. For example, a fire exit door is red in color as well as has an exit sign near the top of the door. The coupling of environmental cues with exit signs supports recent findings (Galea, Xie, & Lawrence, 2014), where the authors found that only 38% of the sampled participants were able to see static exit signs. Adding environmental cues to exit signs makes a more salient landmark near a decision-point, but not all decision-points are easy to remember. There may be many reasons why a particular landmark is forgotten after being observed (e.g., lack of attention, (Reason, 1990)). Intersecting escape routes often become a problem as they require careful decision-making if a person is to stay on the shortest unobstructed route during a real emergency (Koutamanis, 1995). The information gathered at each decision-point is known as a navigational command (e.g., turn left, turn right, or continue going straight on the route). For this paper, the agent stores two pieces of information at the decision-points: (1) recognizing the landmark along the route, and (2) recalling the navigational command at the landmark. The landmarks along a route are the basis to remember the navigational commands (Waller & Lippa, 2007) that direct the agent when to stay on the route and where to make a turn.

The problem of wayfinding during an emergency has a wide scope. Airports, railway stations, industrial buildings, and public places are important locations where there are a fair number of people, at all the times, who do not know the environment very well, or who have only limited memories of the most important escape routes. However, as the VE used in this study simulates the environment of an offshore petroleum platform, the discussion here is focused on this particular context. The time to reach the designated muster station from a current location is an important variable to determine the effectiveness of an evacuation training exercise (Kristiansen, 2005), and therefore, it has significance in real emergencies. The Piper Alpha disaster witnessed 167 casualties due in part to insufficient evacuation procedures (Cullen, 1993). The SOLAS Chapter II-2 Regulation 13 (IMO, 2009 p. 167) has specific guidelines about the use of signs in an escape route on offshore platforms. The regulation requires all signs to be photoluminescent material, or marked by light, or both. The OSHA fact sheet describes the operational features of all escape routes and advocates at least two routes for safe and rapid arrival at muster points in case of an emergency (OSHA, 2003). Nevertheless, on real platforms, there are locations where people must use environmental cues to orient towards the exit direction. These include engine gear rooms and open spaces.

A colored Generalized Stochastic Petri-Net (GSPN) based Route Learning (GSPNRL) model that uses landmarks is proposed in (Danial, Khan, & Veitch, 2018) to represent how humans forget some landmarks, while remembering others when these are observed during the first visit to an unfamiliar, or new environment. The present work extends the scope of the GSPNRL model. We propose a methodology whereby the model can be used in an iterative way, such that each iteration produces a trace of a single route learning exercise. Successive iterations of the model should reveal a learning curve that converges. This means that the total number of landmarks missed from learning in the model after the first iteration should decrease in the successive iterations. The literature review shows that, although, some route learning and generation models (Caduff & Timpf, 2005; Goldschmidt, Manoonpong, & Dasgupta, 2017; Götze & Boye, 2016) consider landmarks as cues that direct an agent nearer to its goal, how should successive training exposures impact the learning of a route? Even if a sequential route learning model for agents is developed, how does its output show resemblance with human route learning curve obtained over multiple training exposures? The proposed methodology aims to answer these questions by generating simulated responses from agents and then comparing these responses with real people's route learning responses collected in a virtual environment.

The approach developed in this study has potential to be used in intelligent agents to augment VE training. These intelligent agents could provide opportunities for participants to learn the consequences of forgetting a decision-point along a route. For example, during an emergency evacuation scenario, a tutoring agent can show trainees the consequences of forgetting navigational commands or a decision point along an egress route. Such scenarios are important to show the importance of escape route learning to novice personnel beginning work at an offshore installation. Simulating the route learning of agents with different levels of route knowledge is another application that can be used to represent the real-world variability of people in simulated emergency evacuation scenarios. The route learning approach proposed here can be used during the design phase of a ship or offshore platform or similar complex space because it provides a model to analyze which escape routes are easier or more difficult to learn.

Section 3.2 provides the theoretical background on wayfinding, reviews the use of VE training for wayfinding, and outlines the modeling of intelligent agents for wayfinding and evacuation behaviors in emergencies. Section 3.3 briefly explains the GSPNRL model (see details in Chapter 2, and Appendix A). In section 3.4, a case study for collecting human performance data for validation of the proposed approach is presented. Section 3.5 shows the simulation results and compares them with empirical results collected in Section 3.4. Section 3.6 shows sensitivity analysis of the GSPNRL model. Section 3.7 discusses the results and concludes with possible future directions.

3.2 Literature review

Passini (1977) defines wayfinding as a process that leads a person from a source to a destination location. During this process, people learn to recognize the origin and the destination locations, and to remember certain environmental features that help them to remember which way to take (Golledge, 1999a). These environmental features are called landmarks, and they play an important role in adjusting the direction of travel towards the desired destination (Gale et al., 1990; Lee, Shusterman, & Spelke, 2006; Lynch, 1960; Nys, Gyselinck, Orriols, & Hickmann, 2015). Many experiments involving human participants have shown the importance of landmarks during a course of learning a new route. As an example, the experiment reported in (Gale et al., 1990) shows that children of ages 9-12 learned more about the environmental features of the route on intersections. Emo (2014) investigated the effect of spatial configuration, such as the structure of the space and the integration of paths, on individual decision-making and visual attention during wayfinding. Sources that cover various aspects of route and survey knowledge acquisition — the concepts like home base behavior, and path integration — can be found in (Eilam, 2014; Golledge, 1999b; Vandenberg, 2016). The study (Sharma et al., 2017) attempts to quantify the influence of landmarks on wayfinding by using electroencephalograph (EEG) signal analysis. EEG signals comprises signals coming from human brain with respect to various states the human body undergoes, such as resting position, and a stressed mental state (Kumar & Bhuvaneswari, 2012). The authors conducted a wayfinding experiment in a VE with two groups. Participants in the first group were given navigational tasks in an environment with landmarks. In the second group, landmarks were removed. The EEG data were collected using a 64-channel device. The behavioral analysis revealed that the participants of the first group took less time to complete the navigational tasks and committed fewer errors due to the presence of landmarks. The EEG data analysis showed that the left-hemispheric activation was more visible in those participants who used landmarks compared to those who did not. The questions of which landmarks have more influence and how they contribute to learning a route when more than one training exposure is given have not been addressed.

The Fire Safety Engineering Group at the University of Greenwich (2017) developed buildingEXODUS as software to simulate evacuation from built environments using adaptive agents. The evacuation model simulates people-people interactions, people-fire interactions, and people-structure (such as exit signage) interactions. Although these agents are adaptive and the interaction with the signage system is stochastic (Galea, 2003), i.e., many simulations are needed to come up with a viable solution, and successive simulations produce different results, they lack consideration of the failure of signage system or insufficiencies in the signage system. Also, buildingEXODUS does not take into account environmental landmarks. It relies only on a signage system.

Landmarks are used when it is difficult to use signage, for example, due to signage noise (as at airports or railway stations) or due to difficulty in seeing the signs in complex structures such as engine rooms on offshore platforms, and also due to a phenomenon called *learned irrelevance* (Mackintosh, 1973), which is a kind of an impairment in attentional set-shifting that results in an inability of a person to use or learn information that was considered unimportant in the past (Gruszka, Hampshire, & Owen, 2010). Many incidents have been reported where the designated signage system failed to fulfill the needs of people in real emergencies. Such include the Nairobi Westgate shopping mall terrorist attack (BBC, 2013) in which the escape routes were compromised and redirection of people away from compromised exit routes consumed valuable egress time. In such situations, people use their instinct of using landmarks to navigate towards a safe location. Similarly, during the evacuation of Dusseldorf Airport due to a fire incident, many people could not see the emergency exit signs and that delayed the evacuation process (Weinspach et al., 1997).

Two problems are of particular importance: the case of compromised escape routes, and the case when people are unable to see or recognize the exit signs for some reason. Dynamic signage systems have been proposed to address the first problem. These are systems that can guide people towards a safe escape route, just as ordinary exit signs do, but can also redirect evacuees away from an escape route that has been compromised (Xie, Filippidis, Galea, Blackshields, & Lawrence, 2012). The second problem may have different solutions including: (1) the use of photoluminescent material for exit symbols (IMO, 2009), and (2) the consideration of size and placement of signage. For agents in VEs, the computation of visibility catchment area (VCA) should also accommodate these solutions (Filippidis, Galea, Lawrence, & Gwynne, 2001) so that an agent could detect a sign in the same way a human counterpart would see that sign. A VCA for an agent can be defined either from the agent's perspective or from the exit sign's viewpoint. In the former approach, the VCA is defined as an area around the agent so that when some object falls into this area the agent is able to see that object. In the later approach, which is considered computationally efficient, VCA is defined as an area that surrounds an exit sign such that when the agent comes into that area the navigation sign is expected to be visible unless another object obstructs the sign. The present work proposes to use other environmental cues (the landmarks), in addition to using the recommended signage, as part of evacuation training curriculum for cases where most of the people are part of the workforce, and therefore can be trained. Route learning based on landmarks, and not based only on the available signage, is an important solution regarding selecting an alternate route in cases when a primary escape route is blocked, or when exit signs are difficult to observe. Further, there are practical limitations of using signage on every point in a large facility such as an oil and gas offshore platform, so wayfinding based on landmarks is of general importance.

Caduff and Timpf (2005) modeled how knowledge about landmarks in an environment can be integrated into wayfinding tasks so that an agent can generate a route based on the available landmarks. The authors considered the environment as a graph containing many routes from a source to a destination. Each node of a route within the environment has a landmark. The authors ignored the possibility of turnarounds. Landmark selection is made by assigning a low weight to the most salient landmark and high weight to the least salient one. A weighting function is used that expresses the weight of a node in a route as a linear combination of the distances (between the nodes and the landmarks), orientation (of the traveler concerning the landmarks), and salience of landmarks. Finally, route generation exploits a revised version of Dijkstra's shortest path algorithm (Dijkstra, 1959; Sedgewick & Wayne, 2011 p. 638) using node weights, and provides a route based on landmarks. However, as a part of a graph, the authors focus more on shortest path generation based on landmarks rather than answering more fundamental questions of human cognitive abilities (e.g., remembering or forgetting a landmark due to lack of training) and therefore, the effect of training on remembering landmarks remains unaddressed.

3.3 Overview of the GSPNRL model

The literature on human wayfinding behavior, as explained in the preceding section, suggests that wayfinding is centered around the concept of landmarks. In the absence of a landmark, or if a landmark is either not remembered or the associated heading is forgotten, a navigator explores the environment to try to discover a route to the destination. Communicating with others around is also a way to figure out a possible heading that may lead to the destination. Figure 3.1 explains the steps that are typical of a human wayfinding behavior. The GSPNRL model (Figure 3.2) follows the logic presented in the flowchart in Figure 3.1 except that the *explore* behavior used in agent programming is considered as something an agent can use from its steering capabilities (Buckland, 2004). The primary focus of the GSPNRL model is to capture a navigator's state of remembering or forgetting a landmark with associated direction

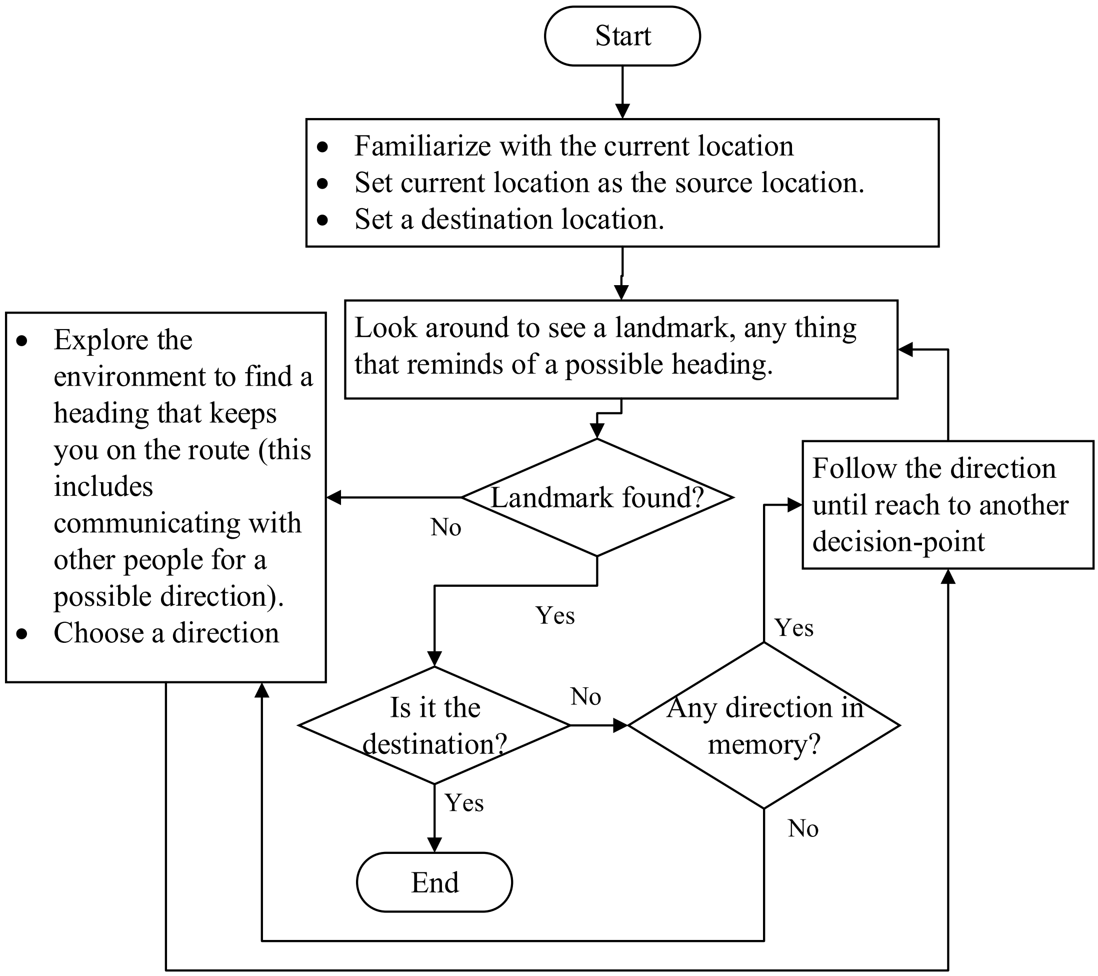


Figure 3.1. A typical human wayfinding behavior

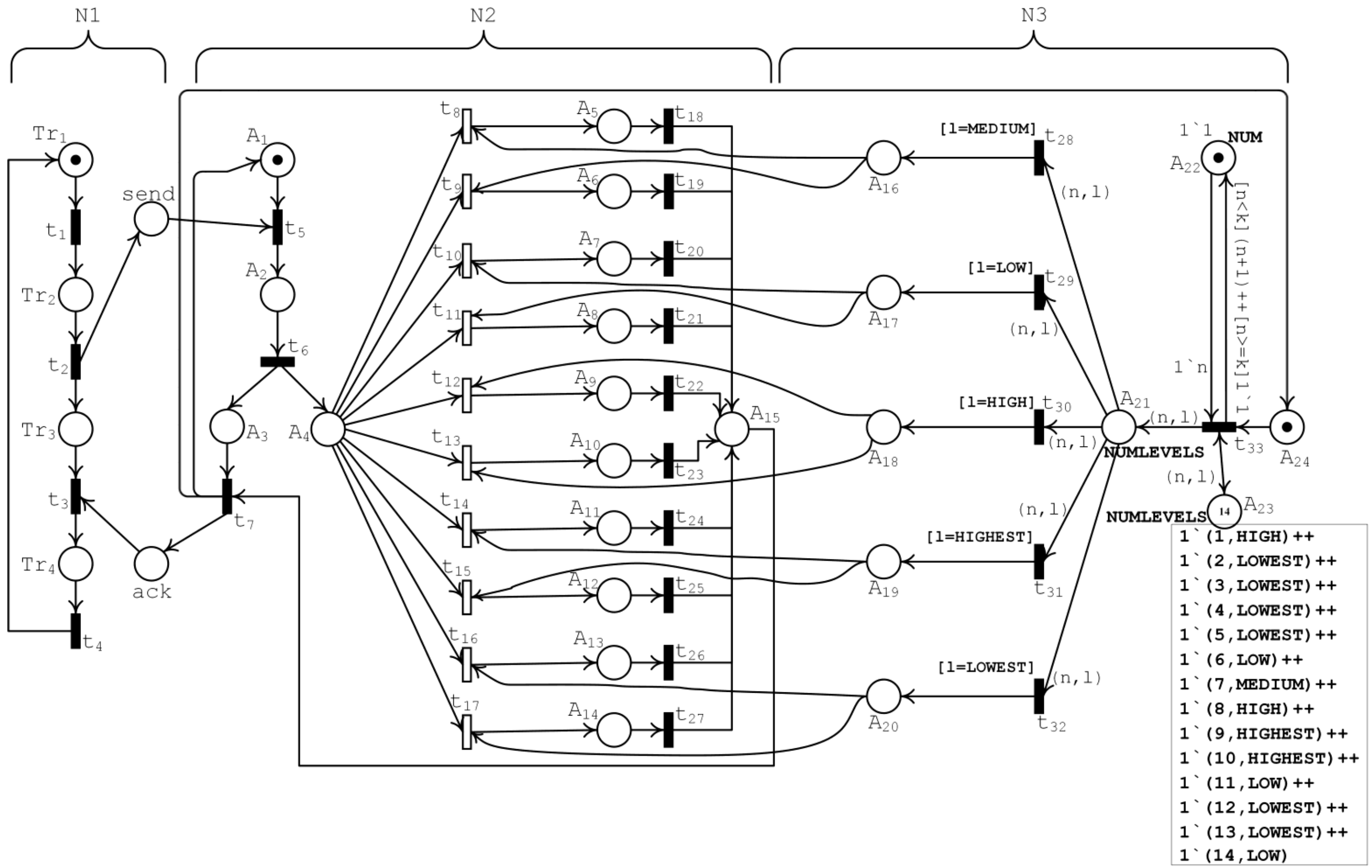


Figure 3.2. The GSPNRL model.

at all decision-points along a route. The model could be used as a route learning mechanism in intelligent agents so that the agents can exhibit human-like route learning behavior, at the same time exploiting other built-in features such as communicating with other agents about a direction.

The GSPNRL model is primarily a combination of three nets labeled as N1, N2, and N3 in Figure 3.2. The net N1 sends the navigation commands to N2, one after another. N1 could be a trainer (which may be another agent sending commands via a communication medium and then receiving back the acknowledgments), or it could be the same agent retrieving its memory, from a repository of experiences, and previously used navigation commands at a location observed through N3. The navigation commands include directions such as move left and move right.

The net N3 takes a list of difficulty levels associated with the landmarks or decision-points along the route in the order of occurrence as the agent traverses from the origin to the destination. The landmarks are divided into five classes based on their difficulty to remember: (*i*) lowest difficulty, (*ii*) low difficulty, (*iii*) medium difficulty, (*iv*) high difficulty, and (*v*) highest difficulty. The idea behind this classification is that there is less chance for a navigator to remember a navigation command used near a location that has a high difficulty. That is, if the location where a navigation command was used previously does not constitute a good landmark, it will be difficult to remember that navigation command. Thus, in future, the agent is likely to miss that navigation command and hence may not follow the route. A location that is difficult to remember would mean that its features are not salient (Götze & Boye, 2016) and, therefore, people would face difficulty to learn it. On the other hand, easy to remember locations, due to salient environmental features, increase the chances to remember the navigation commands (McKinlay, 2016). This behavior would mimic the situation of forgetting and remembering what has been seen a moment before. Since the GSPNRL model, as shown in Figure 3.2, does not explicitly consider multiple training sessions, an iterative approach is proposed in this work. The GSPNRL model requires minor adjustments in its inputs to accommodate each iteration. The execution of an iteration of the model is like a function call in a typical computer programming loop, and each iteration simulates a single training/learning session.

Figure 3.3 demonstrates two hypothetical cases: (A) a perfect learning case and (B) a forgetting navigational commands case. In the perfect learning case (Figure 3.3.A), the output will be formed by developing a 1:1 correspondence between all the navigation commands and the corresponding landmarks relevant to a route. It means that the agent knows each move and turn required from the origin to the destination of a route. In situations when the agent cannot retain some of the required navigation commands (Figure 3.3.B), the output will not form a 1:1 correspondence with the landmarks and the associated navigational commands. This behavior reflects the forgetfulness of navigation commands along a route when a person is not given sufficient training.

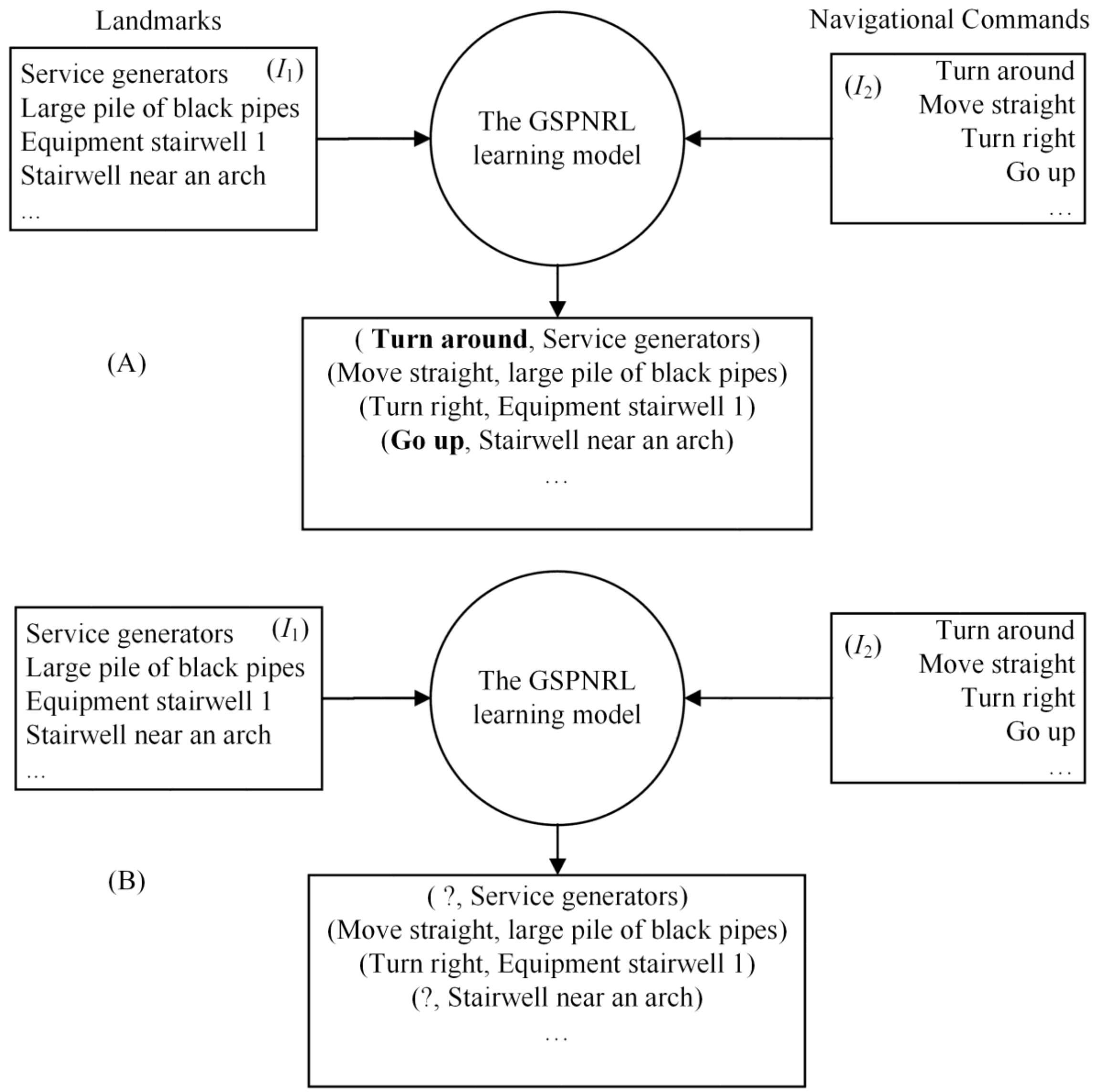


Figure 3.3. (A): A perfect learning case in which an agent using the model will retain everything that it was trained for. The input I_1 is the set of landmarks, and I_2 refers to the navigation commands required to reach the destination, (B): A realistic learning case in which an agent using the model does not retain everything that it was trained for. The question marks refer to the missing information in the output node. The learning model is proposed in Figure 3.2.

3.4 Collecting the human performance data in a route-learning scenario: a case study

3.4.1 AVERT simulator

The virtual environment used in this study is the All-hands Virtual Emergency Response Trainer (AVERT). AVERT is a simulator of an offshore facility comprised of several decks. It allows participants to train and develop better responses to emerging hazards. The training curriculum of the AVERT virtual environment includes basic offshore safety practices with the help of a learning management system.

3.4.2 Evacuation task in the AVERT simulator

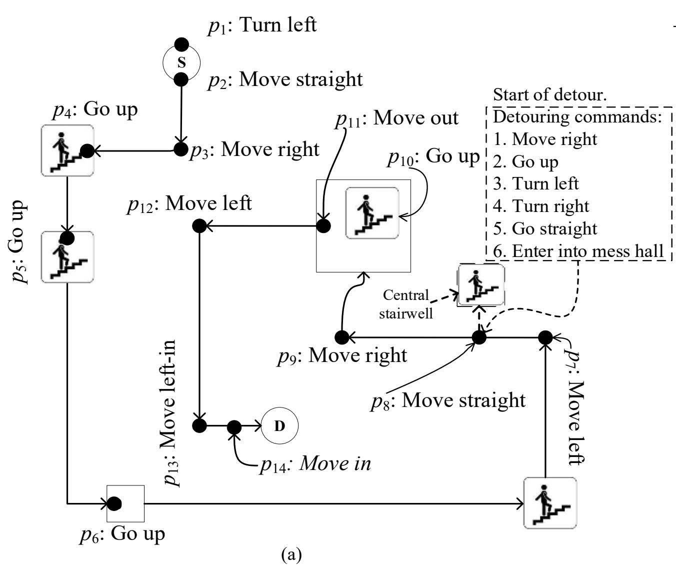
Training scenarios in AVERT were designed to teach participants the escape routes available to them from their worksite in the engine room of the platform. There were three escape routes from the engine room, and they are ranked based on highest to lowest priority (primary, secondary, and tertiary routes). All the routes have been marked with required exit signs according to SOLAS Chapter II-2 Regulation 13 (IMO, 2009 p. 167). For this work, the primary escape route is the only route used. During the training and testing scenarios, participants were required to follow one of the escape routes and muster at their designated muster station or lifeboat station. The participants' worksite was located in the engine room on the 3rd deck, and the muster stations (both the mess hall and lifeboat) were situated at A-deck.

A schematic of the primary escape route along with a description of landmarks is shown in Figure 3.4. The primary escape route is named R1. To go from the worksite in the engine room to any of the muster stations, one has to climb up three decks. R1 uses the equipment stairwell#1 that leads to the 2nd deck. From here, the route goes up to the upper-deck using another stairwell. From the upper deck, there are two ways to reach A-deck. The first, using the central stairwell, and the second, using a direct stairwell situated near a corner of a corridor in the upper-deck. R1 uses the direct stairwell to reach A-deck near the starboard side of the vessel just opposite to the lifeboat station on the same side, which is the secondary muster point. Opposite to the secondary muster station is the primary muster station, i.e., the mess hall.

3.4.3 Experimental results

J. Smith (2015) experimented to assess the efficacy of AVERT training for offshore emergency scenarios. Thirty-six people participated in the experiment, which divided the participants into two different training exposure groups: Group 1 participants were given repeated exposure to training and learning exercises (17 participants), and Group 2 participants received one initial exposure to the training (19 participants). After the training was completed, both groups were tested on a series of learning objectives. The training material for Smith's experiment targeted six learning objectives, which are: (1) establish spatial awareness of the environment, (2) routes and mapping, (3) alarm recognition, (4) continually assess situation and avoid hazards on route, (5) register at temporary safe refuge, and (6) general safe practices (J. Smith, 2015 pp. 59-60). The present work uses the scenarios that cover the first three learning objectives. Participant data from group 1 was used for the validation of the simulation results reported in the paper. All group 1 participants completed each training and testing scenario only once, and in each scenario, the decision-points or landmarks were encountered only once.

The participants were tested repeatedly over the course of three separate sessions (S1, S2, & S3). Each of the sessions involved test scenarios in responding to a range of activities. Only the work related to the first two sessions is used here (see Figure 3.5). Session one (S1) was designed to train and test the participants for environmental awareness. S1 contains a 30-minute video tour (called LE1) of the virtual platform, two training/learning scenarios (LE4, LE5) and two testing scenarios (TE2, TE4). Session two (S2) was designed to train and test the participants for emergency alarm recognition



p_i	Nearby landmarks
p_1	Crowded location
p_2	A pile of black pipes
p_3	A steel railing
p_4	Equipment stairs # 1
p_5	Stairwell ending at an arch
p_6	Small stairwell
p_7	Two doors on opposite ends of a corridor
p_8	An exit door opening in central stairwell
p_9	A fire exit door with a warning sign
p_{10}	Stairwell where there is a door on just left.
p_{11}	A fire exit door
p_{12}	A blue colored pillar
p_{13}	Lifeboat station
p_{14}	Mess hall door

(b)

Figure 3.4. (a) A schematic of the primary escape route (R1) in AVERT simulator from the worksite to the muster stations, (b) description of the landmarks near the decision-points (DP), pi, there are fourteen decision-points. The distances are not scaled but are presented for better readability.

during muster drills. This session contains two training scenarios (LA1, LA4) and two testing scenarios (TA2, TA4). Details of the training and testing scenarios and the participants' performance are given in (J. Smith, 2015).

Initially, participants received basic training and watched three route videos that highlighted the primary, secondary, and tertiary escape routes available from the worksite in the engine room to the muster stations. Each of the decision-points was explored in route videos. Typically, a participant watched each video two to three times. The participants then performed practice scenarios (LE4 & LE5) and testing scenarios (TE2 & TE4). Practice scenario LE4 required participants to practice how to egress from the worksite to the lifeboat station using the primary route. Practice scenario LE5 was designed for practicing the egress procedure using the secondary route. Other scenarios (LE2 and LE3) dealt with routes from cabins, which are not considered here, but these scenarios exposed a portion of the route that leads to the central stairwells.

In testing scenarios TE2 and TE4, the participants were asked to egress, respectively, at primary muster and lifeboat stations as quickly as practicable using any route. The same tasks were asked in the scenarios LA1, LA4, TA2 and TA4 with the addition of two alarms. In this training session, the general platform alarm (GPA) or the PAPA was sounded as a means to indicate which muster station to target. Both the alarms are

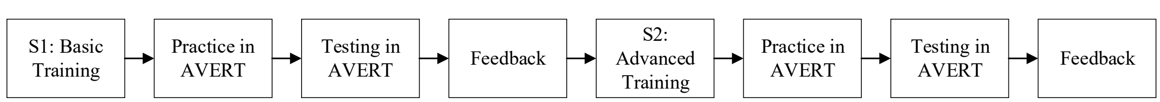


Figure 3.5. Training exposure to group-2 participants.

different audible signals. The `GPA` alarm requires the participants to go to the primary muster station, i.e., the mess hall. The `PAPA` alarm indicates a higher severity of an emergency and requires the participants to reach the lifeboat station at the starboard side of the vessel. Table 3.1 presents a sample of data collected from the participants for the scenarios LE4, TE2, TE4, and LA1. This data is binary, representing whether a participant has followed the navigation command needed to be followed at a decision-point or not. Figure 3.4 shows the navigation commands to be followed at each decision-point. Each participant was taught these commands through the route videos and the training sessions.

During the training and testing scenarios, the participants' route selection (primary, secondary or tertiary routes) and how much they were able to stay on the route were recorded. Since an overwhelming number of trials performed by each participant exploits R1 as the main route, the secondary and the tertiary route learning data were not of good statistical size and have been excluded from analysis. Therefore, the route learning observed and simulated here only uses R1 as the target route for spatial learning. For each training and testing scenario, the participants' data were compared to the fourteen decision-points (p_i) for the R1 escape route. It was observed that some of the participants used the central stairwell as a detour from decision point p_8 in R1 from the worksite scenarios (this will be discussed in further detail in the next section).

Table 3.1. A sample of data from four scenarios LE4, TE2, TE4, and LA1 with 10 participants tagged as CAG1, CSG1,…, MWG1. Each column represents the data for each decision-points p_1, …, p_{14}. A '1' shows that the participant has made the right decision at the respective decision-point and a '0' means that the participant did not follow the required action. Each entry contains four values, for example, for the participant CAG1, the values for p_1 are 0011 where the first 0 is for LE4, the second 0 is for TE2, the third value '1' is for TE4, and the last value is recorded from the scenario LA1.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}
CAG1	0011	1011	0011	0011	0011	0011	0001	0001	0001	0001	0001	0011	0011	0001
CSG1	0000	0000	0000	0000	1101	1101	1101	0000	0000	0000	0000	0000	0000	0101
DCG1	0001	0001	0001	0001	1101	1101	1000	0000	0000	0000	0000	0101	0101	0101
HSG1	1111	1111	1111	1111	1111	1111	0011	1111	1111	1111	1111	1111	1111	0101
JHG1	0001	0001	0001	0001	1000	0001	0001	0000	0000	0000	0000	0000	0000	0001
JMG1	0001	0001	0001	0001	1111	1111	1111	0101	0101	0101	0101	0101	0111	0111
KHG1	1101	1101	1101	1110	1101	1101	1101	0000	0000	0000	0000	0000	1000	1101
MBG1	0000	0000	0000	0000	1101	1101	1101	1100	1100	1100	1100	1100	1100	0101
MPPG1	1111	1111	1111	1111	1111	1111	0101	1111	1111	1111	1111	1111	1111	1001
MWG1	1111	1111	1111	1111	1111	1111	0111	1010	0010	0010	0010	1010	1010	0101

3.4.3.1 Aggregate sequential learning

The task performance measures how many decision-points are correctly followed because this demonstrates how well a route is learned. Route learning is estimated by calculating the percent average number of failures to follow the required navigation command at each decision point in the entire route for a given scenario. A high value of failures means less learning and a low value of failures means more learning. Figure 3.6 shows the average percentage of participants who failed to follow the decision-points p_1-p_6 in successive scenarios. The gradual learning is indicated by the trend in the decision errors as the participants performed successive sessions (as shown in Figure 3.6 from left to right). Each scenario, whether it is for practice or testing, teaches the participants about the decision-points on route R1. A high value at LA4 in Figure 3.6 shows that the participants forgot some decision-points at which they performed well in the previous scenarios. This is an example of typical human behavior. Figure 3.7 shows the average number of participants who did not follow the required navigation commands at decision-points p_7 to p_{14}. The navigation commands required to follow R1 are shown in Figure 3.4. Only eight cases out of eighty trials for ten participants over eight scenarios (LE4, LE5, TE2, TE4, LA1, LA4, TA2, and TA4) are found where a participant either did not reach any muster station or was mustered at a wrong muster station. The reason for splitting R1 into two parts as shown in Figures 3.6 and 3.7 is to demonstrate that almost half of the route up to p_6 comprises a portion where the participants showed significant learning due to successive training. The other half of the route is the segment from p_7 to p_{14} where, in most trials, the participants preferred a comparatively easy detour from location p_8. It means most of the participants did not go to p_9; instead, they used the central stairwell, by turning right from p_8 instead of going straight to reach the muster stations (see Figure 3.4). The participants had earlier exposure of using the central stairwell that also goes directly near the mess hall. Since p_9 is considered as the highest

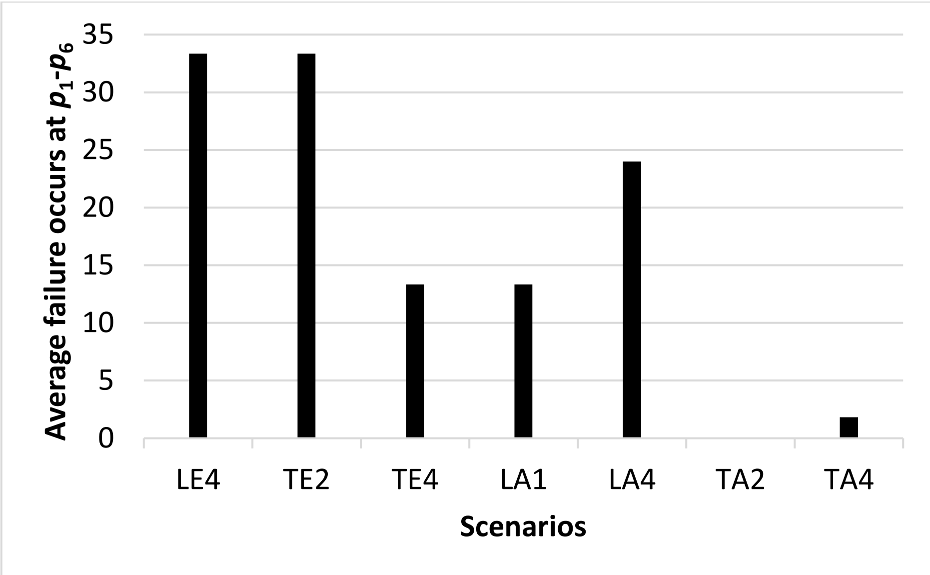


Figure 3.6. The average number of participants failed to follow the navigation commands corresponding to decision-points p_1-p_6. Percentage values are shown on the y-axis.

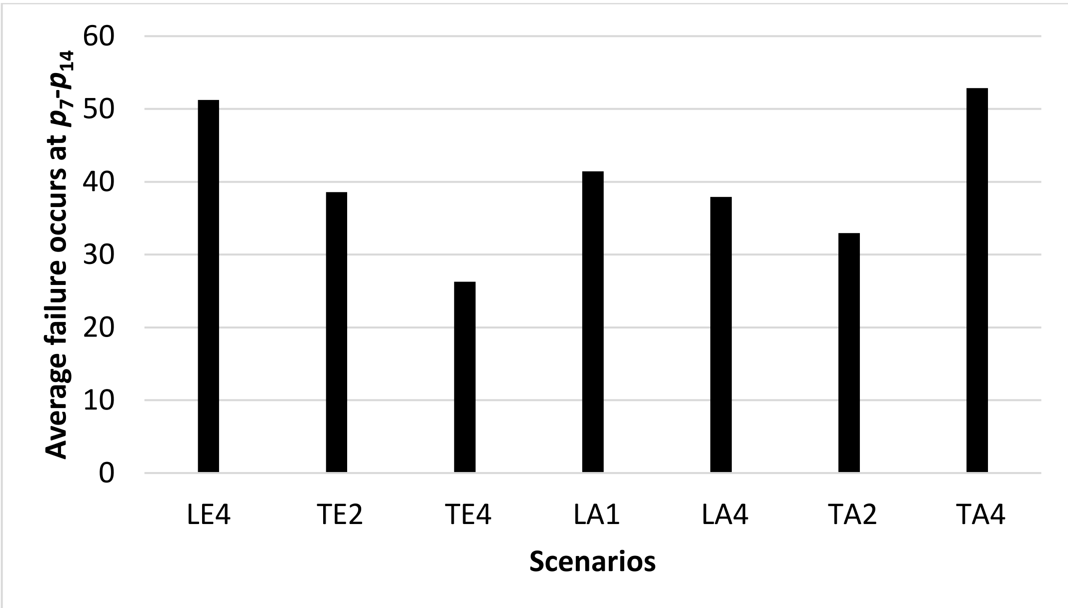


Figure 3.7. The average number of participants failed to follow the navigation commands corresponding to decision-points p_7-p_{14}. Percentage values are shown on the y-axis.

difficulty level decision point, around 40 percent of the participants (± 9%), took the detour and used the central stairwell from p_8.

3.5 Simulation results

The GSPNRL model is developed in the software environment Snoopy (Heiner et al., 2012). The start and end simulation times were set as 0 and 100, respectively. The repeated exposure to training sessions is simulated using the method proposed in Figure 3.8. A total of seven scenarios (LE4, TE2, TE4, LA1, LA4, TA2, and TA4) are simulated where each simulation uses 5000 runs. The rates of the stochastic transitions, t_8-t_{17}, are obtained randomly from the rate ranges defined in Table 3.2. The model inputs are (*a*) the landmarks (regarding difficulty levels denoted by γ)

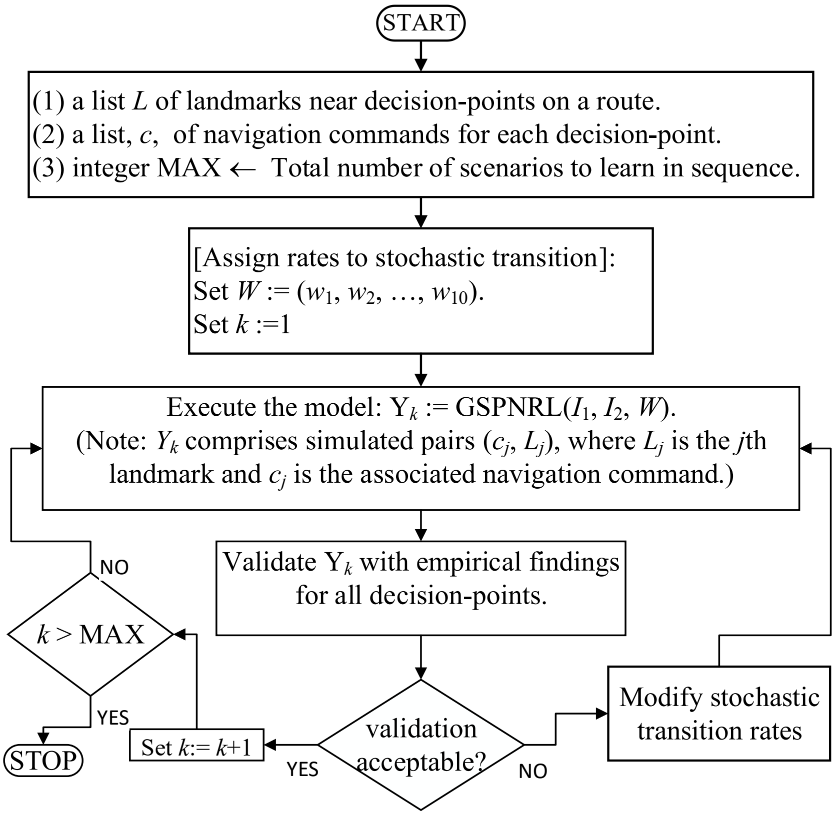


Figure 3.8. A method to apply the GSPNRL model for a sequential route learning task.

Table 3.2. Range of the stochastic transition rates. Rates were assigned to the GSPNRL model randomly from the range below.

Stochastic transitions (t_i)	Firing rate (α)	Difficulty levels (γ)
t_8	(0.2, 0.4]	MEDIUM
t_9	$1 - \alpha_{i-1}$	MEDIUM
t_{10}	(0.05, 0.2]	LOW
t_{11}	$1 - \alpha_{i-1}$	LOW
t_{12}	(0.4, 0.6]	HIGH
t_{13}	$1 - \alpha_{i-1}$	HIGH
t_{14}	(0.6, 1.0]	HIGHEST
t_{15}	$1 - \alpha_{i-1}$	HIGHEST
t_{16}	(0.0, 0.05]	LOWEST
t_{17}	$1 - \alpha_{i-1}$	LOWEST

along R1 near each decision point, and (*b*) the navigation commands required at each decision point (see Figure 3.4). The difficulty levels assigned to each landmark are based on the following criteria: (*a*) how crowded the space is near the landmark, (*b*) the symmetry in the environment, known as mirrored layouts (McKinlay, 2016), (*c*) the presence of easy detours nearby, and (*d*) any salient features, such as color, or shape, that allow the landmark to stand out from its surroundings. If, at some decision point, *p*, condition (a) holds, then its difficulty level γ is set to HIGH. If condition (b) is true, then γ is set to MEDIUM. The value HIGHEST will be assigned to γ if condition (c) is true. If condition (d) is found true, then γ will be set to LOW. In any other case, the value of γ will be set to LOWEST, which is typical of situations where participants can remember the entire scene, for example, recognizing the mess hall or lifeboat station.

The way of assigning difficulty levels to landmarks requires perceiving certain environmental characteristics, such as the presence of many objects, as mentioned in (a), or the presence of mirrored layout, as mentioned in (b). Table 3.3 shows the initial

Table 3.3. Initial difficulty levels assigned to each decision-points in R1.

Decision point (p_i)	γ_{R1}
p_1	HIGH
p_2	LOWEST
p_3	LOWEST
p_4	LOWEST
p_5	LOWEST
p_6	LOW
p_7	MEDIUM
p_8	HIGH
p_9	HIGHEST
p_{10}	HIGHEST
p_{11}	LOW
p_{12}	LOWEST
p_{13}	LOWEST
p_{14}	LOW

scores assigned to the landmarks near the decision-points p_1, p_2, …, p_{14} on R1. In Smith's (2015) case study, the decision point p_1 lies in a big hall where twelve different routes originated. There were more than forty objects of different kinds, such as stairwells, two service generators, many pillars, railings, and other machinery typical of an engine room. This suggested the difficulty level at p_1 to be HIGH. The corridor near p_7 is symmetric from both ends, and it was difficult to remember the correct direction. The difficulty level of p_7 is assigned as MEDIUM. The decision-points p_9, and p_{10} are assigned HIGHEST difficulties because of the presence of an easier detour near p_8 that uses a central stairwell to arrive at the mess hall. The point p_8 has been set to one step easier than p_9. The decision-points p_6, p_{11}, and p_{14} had clear, salient features, as shown in Figure 3.4, which made their difficulty levels equal to LOW. All the other decision-points are assigned the LOWEST difficulty level.

The agent model is used sequentially for simulating the successive scenarios (LE4, TE2, TE4, LA1, LA4, TA2, and TA4) in the order the participants carried out the scenarios with updated difficulty levels corresponding to the landmarks as described in Figure 3.8. Figure 3.9 shows the results from the agent modeling the participants' behavior in the scenario LE4. For each decision-point there is a separate box representing the minimum, the first quartile Q_1, the median, the third quartile Q_3 and the maximum number of times (in percentages) the agent remembers a landmark (decision-point) (Figure 3.9A); forgetting a landmark is shown in Figure 3.9(B). The decision-points with higher difficulty levels, such as p_9 and p_{10}, show higher variability than those with lower difficulty levels, such as p_2, p_6, p_{11}. It means that if many agents are used in a single scenario, they will differ in their ability to remember or forget a decision-point. Moreover, in all cases where a decision-point is

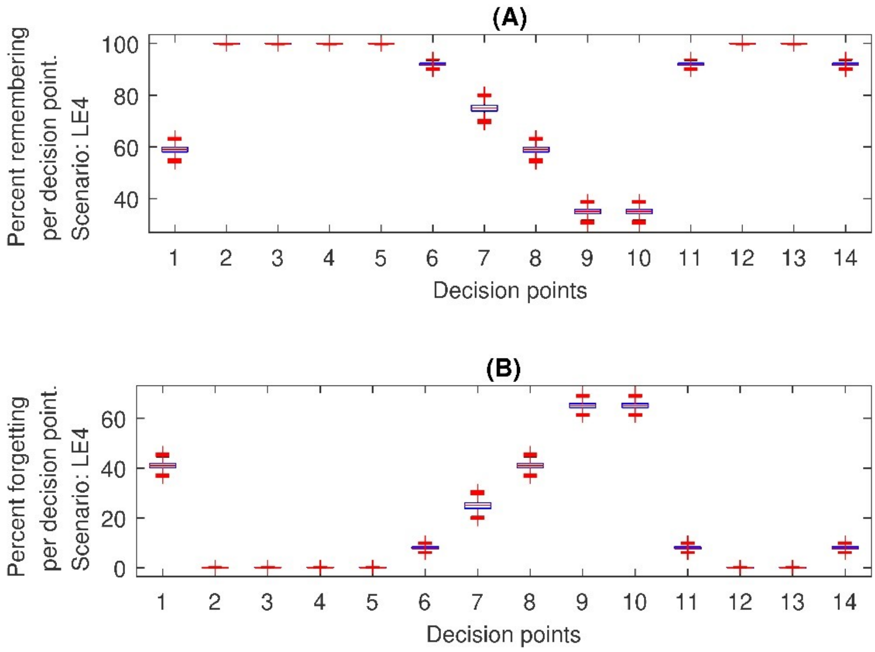


Figure 3.9. Simulated results for the scenario LE4. Represents an agent that models remembering/forgetting the decision-points and the associated navigation commands. (A) The boxplots shown here are based on the percentage of times a decision-point is remembered during 5000 runs of the simulation of LE4. (B) The boxplots shown here are based on the percentage of times a decision-point is forgotten during 5000 runs of the simulation of LE4.

remembered, the agent will use the navigation command associated with that decision-point. The navigation commands used in this work per decision-points are mentioned in Figure 3.4 along with a description of landmarks near a decision-point. Figures 3.10-3.12 represent simulated behaviors for the scenarios TE4, LA4, and TA4. As the participants' average behavior, i.e., the average number of times each decision-point is remembered in a scenario, is similar in LE4 and TE2, TE4 and LA1, and TA2 and TA4, the agent that simulates LE4 may be used in TE2. Similarly, the agents that simulate TE4 and TA4 can be used for the scenarios LA1 and TA2 respectively. The average failure to remember navigation commands associated with the decision-points per scenario is reported in Figures 3.13 and 3.14. The result of the first simulation, comprising 5000 runs, simulates the learning observed in the first scenario, i.e., LE4. After every one or two simulations, the difficulty levels associated

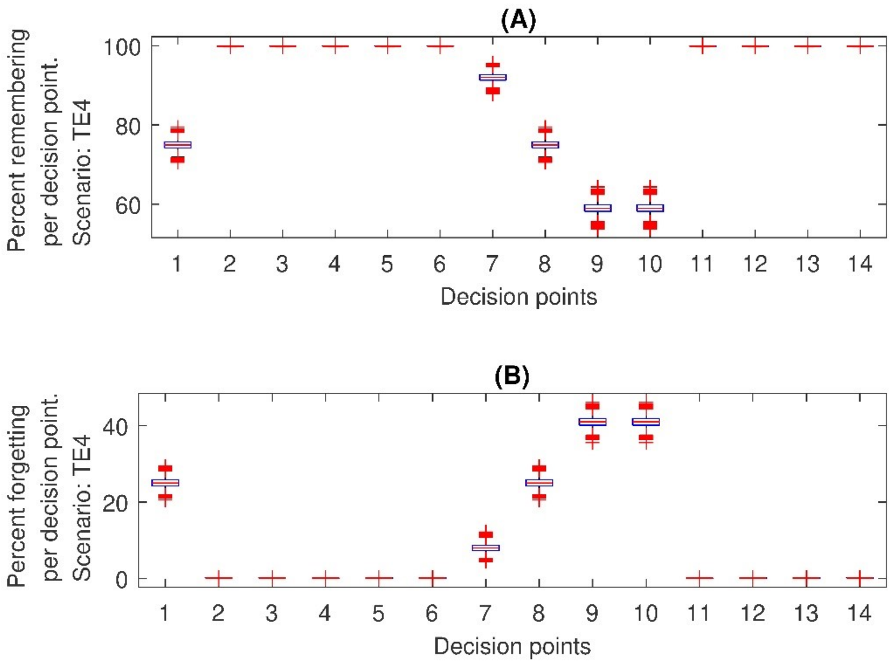


Figure 3.10. Simulated results for the scenario TE4. Represents an agent that models remembering/forgetting the decision-points and the associated navigation commands. (A) The boxplots shown here are based on the percentage of times a decision-point is remembered during 5000 runs of the simulation of TE4. (B) The boxplots shown here are based on the percentage of times a decision-point is forgotten during 5000 runs of the simulation of TE4.

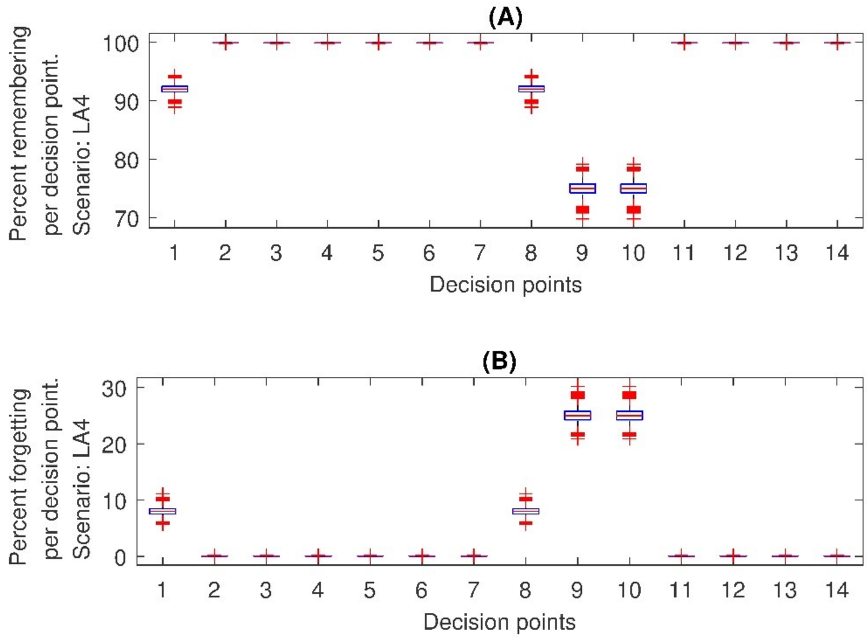


Figure 3.11. Simulated results for the scenario LA4. Represents an agent that models remembering/forgetting the decision-points and the associated navigation commands. (A) The boxplots shown here are based on the percentage of times a decision-point is remembered during 5000 runs of the simulation of LA4. (B) The boxplots shown here are based on the percentage of times a decision-point is forgotten during 5000 runs of the simulation of LA4.

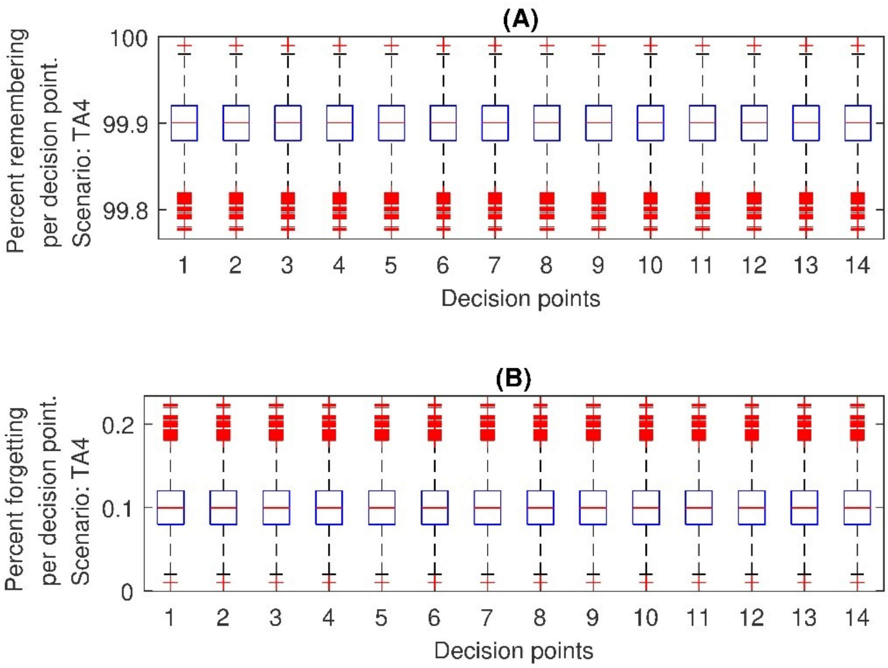


Figure 3.12. Simulated results for the scenario TA4. Represents an agent that models remembering/forgetting the decision-points and the associated navigation commands. (A) The boxplots shown here are based on the percentage of times a decision-point is remembered during 5000 runs of the simulation of TA4. (B) The boxplots shown here are based on the percentage of times a decision-point is forgotten during 5000 runs of the simulation of TA4.

with all the landmarks are decreased by one step (see the flowchart in Figure 3.8). Again, the choice of whether the difficulty levels should be decreased after each simulation, and to what extent, depends on the difficulty levels of the landmarks. For instance, the HIGHEST difficulty level should require more practice sessions than the MEDIUM difficulty level. However, to keep the simulation process simple, in Figures 3.13 and 3.14 we decrease the difficulty levels of every decision point after two simulations until LA1. We have done this because initially most of the decision-points carried higher values of the difficulty levels. After LA1, the difficulty levels were decreased after each simulation. The justification for this decrease is based on the assumption that the past exposure to these landmarks will make them more recognizable in repeated exposures and, as a result, easier to remember. This process is repeated until the difficulty levels associated with all the landmarks are reduced to a minimum. As an alternative to this approach, one may consider changing the rates

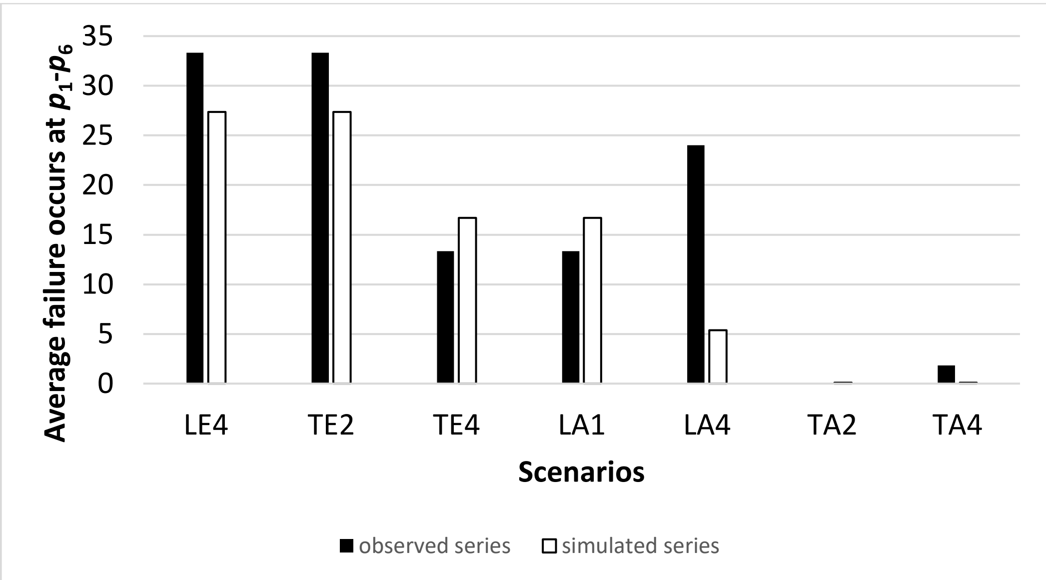


Figure 3.13. The average number of participants failed to follow the navigation commands corresponding to the decision-points p_1-p_6 are represented by black filled bars. Simulation results corresponding to each scenario are shown by unfilled rectangular bars. Percentage values are shown on the y-axis.

of the stochastic transitions in successive learning episodes of scenarios. A repeated application of the model on input sets (such as I1 and I2) produces the result where there is a 1:1 correspondence between each value of I1 and I2. The simulation results are shown by rectangular bars (unfilled) in Figures 3.13 and 3.14. The rectangular bars (colored in black) show the experimentally observed results obtained for human participants (also reported in Figures 3.6 and 3.7) for comparison purposes. There is a close relationship between the simulated and the observed values in Figure 3.13. The paired two-sample *t*-test between the simulated and observed series, where each series contains the average percentage of participants who failed to do as required in the scenarios LE4 to TA4, accepts the null hypothesis, with 95%

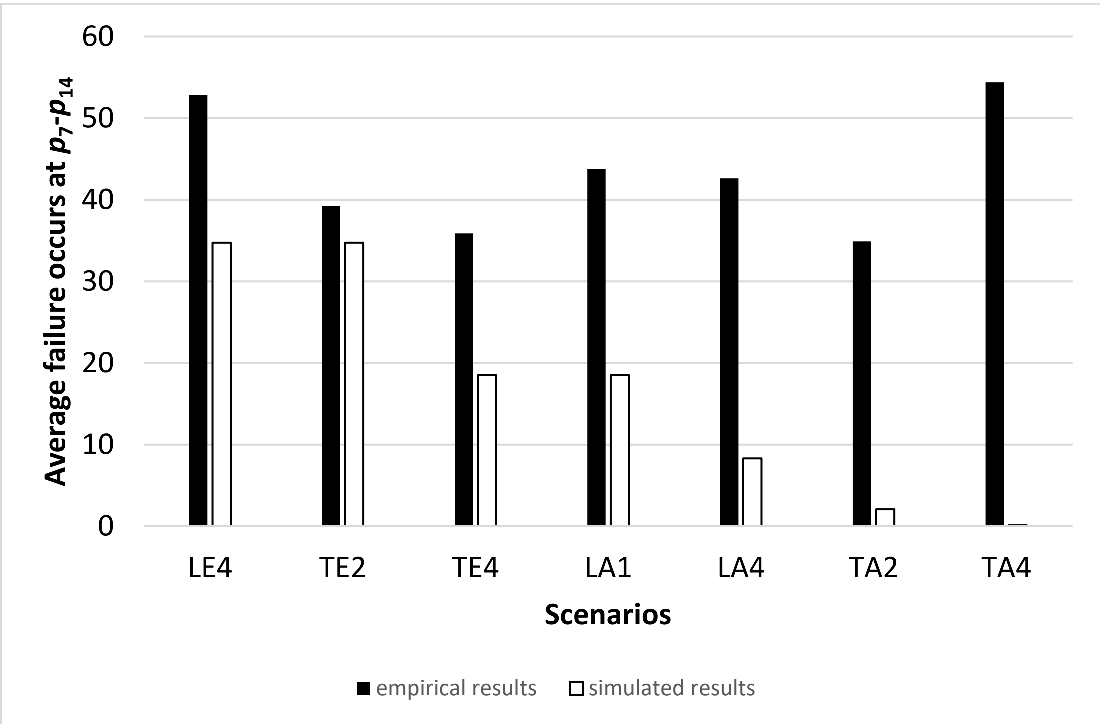


Figure 3.14. The average number of participants failed to follow the navigation commands corresponding to the decision-points p_7-p_{14} is depicted with black rectangular bars. The unfilled bars show the simulated results. Percentage values are shown on the y-axis.

confidence, that both series have the same mean. However, the result of a *t*-test on the series reported in Figure 3.14 rejects the null hypothesis of equal means, with 95% confidence.

3.6 Sensitivity Analysis

Parametric sensitivity analysis (Muppala & Trivedi, 1990) of the proposed model is carried out to observe how sensitive the remembering or forgetting behavior is to the rates of the stochastic transitions. The stochastic transitions t_8-t_{17} have separate output places, which are A_5-A_{14}. The derivatives of the steady-state probabilities concerning a fractional change in the parameter values are reported in Figure 3.15. The parameter values, that is, the values of the rates, are taken in the ranges defined

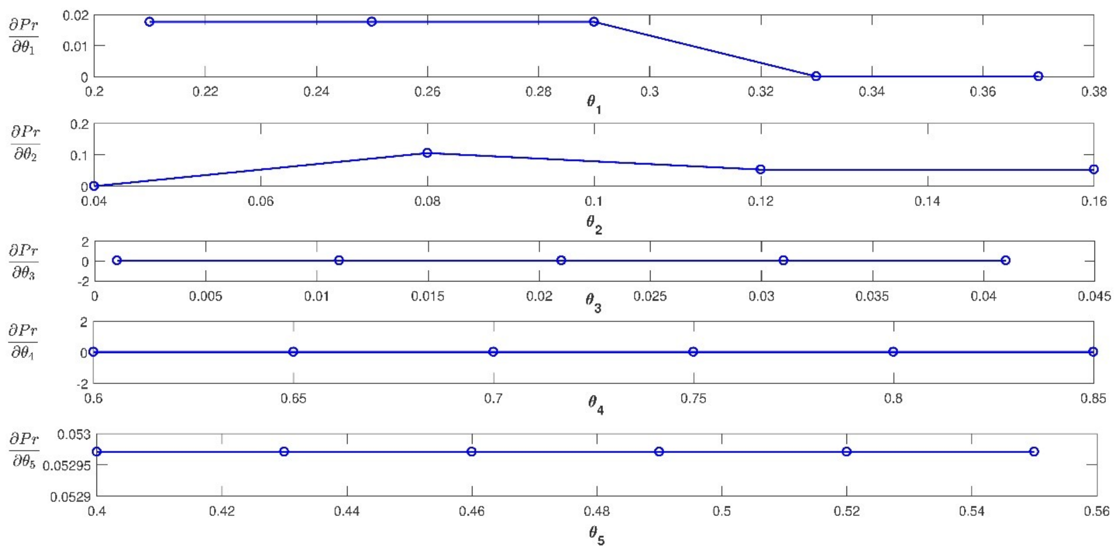


Figure 3.15. The derivatives of steady-state probabilities (that is, the probability that the output place of the transition t_n, the place A_{n-3}, has one token) concerning small changes in the parameter θ are shown. The parameters θ_1, θ_2, θ_3, θ_4, and θ_5 are the rates of the stochastic transition t_8, t_{10}, t_{12}, t_{14}, and t_{16} respectively.

in Table 3.2 so that the reachability graph of the GSPNRL model does not change. The parameters θ_1, θ_2, θ_3, θ_4, and θ_5 are the rates of the stochastic transition t_8, t_{10}, t_{12}, t_{14}, and t_{16} respectively. For θ_3, θ_4, and θ_5 the model is insensitive to any changes in the values, however, small changes in the probabilities are observed for higher values of θ_1 and θ_2. Overall, the GSPNRL model is not sensitive to changes in the rates of stochastic transitions for the range of values described in Table 3.2.

3.7 Discussion and conclusion

The simulation results for R1 up to p_6 show close resemblance with the experimental results of participants (see Figure 3.13). On the other hand, the *t*-test results for the simulated and experimental series reported in Figure 3.14 show that both series are different. The GSPNRL model was successful at modeling the forgetting commands and critical decision-points on R1 from p_1-p_6. After these nodes, the model is no longer effective at matching the human participants' variable performance. One possible reason for this simulated behavior is the use of linear decrement in the difficulty levels beyond p_6. The observed behavior in Figure 3.14 seems to fluctuate, reflecting that some participants kept repeating the mistakes they tried to learn. There may also be a psychological reason for the observed fluctuation beyond p_6, such as focus of attention. Generally, it is difficult to reproduce such an erratic behavior from a modeling perspective. The model could not accurately represent the deviations from the designated route made by participants. However, it does provide a model of someone who learned the route over time. This could be useful, intelligent agent behavior for in-simulation instructions and learning aids to help improve participants' performance. Ideally, when modeling real-world emergency evacuation situations, agents can come in several forms: those that comply with the mustering and wayfinding procedures, and those that have difficulty following them. Those that follow the procedures are useful to set examples for participants and to teach participants what to do properly in the event of an emergency. Those that have difficulty following safety procedures would be useful to implement in a simulation to add realism. Therefore, although the model does not accurately capture the observed individual behavior for all decision nodes, it does have the potential to be used as a model of compliant, rule-following training guides for a VE. Such VEs can be used in a variety of training settings, such as to train students or workers. Trainees who observe agents based on the proposed GSPNRL model could learn the dangers of making the wrong decisions. For example, an agent that makes a wrong choice at some decision point could later find it difficult to reach a muster location due to an evolving hazard.

A VE where intelligent agents, based on the proposed work, can collaborate with human participants would be an effective venue for team muster drills. Also, the participants can make a note of decision-points that require their keen attention to stay on the desired route. Non-interactive scenarios in which such agents perform certain tasks can be developed to teach the tasks to human trainees as part of basic training. Another type of application of the agent model is for assessing the evacuation qualities of newly designed spaces. For example, agents could be deployed in a large hotel, or cruise ship, in the design stage, as a way of evaluating the ease of use of escape routes in a variety of scenarios. Such an evaluation might help identify areas that are likely to be relatively problematic in an emergency evacuation. Having identified these areas, they might be redesigned, or aids to escape, such as signage, might be added.

So far, the difference between simulation results and the observed performance from participants in the latter half of route R1 (as shown in Figure 3.14) has been attributed to the model's performance. The model uses decreasing difficulty levels in successive scenarios because we want the agents to exhibit progress in route learning in successive training scenarios, just as we expect a cohort of trainees to learn the escape route by performing successive training sessions. However, the participants' poor performance in the latter half of R1 suggests that they did not acquire the expected competency. This may indicate deficiencies in the original training curriculum, especially regarding addressing the more difficult scenarios. The cohort performance from p_1-p_6 was satisfactory. This part of the route had only one decision point that was of HIGH difficulty level, and that was p_1, which came right in the beginning. The rest of the points from p_2-p_6 had less than or equal to LOW difficulty.

On the other hand, the point p_7 is of MEDIUM difficulty, and then the three consecutive points, p_8, p_9, and p_{10} are of greater than or equal to HIGH difficulty. This was the part of the route where the participants showed less than the desired competency with an increased rate of errors compared to that observed on p_1-p_6. The proposed approach suggests a reason why the participants did not learn to the expected competency and provides some insight into how the training curriculum might be adjusted to improve the participants' learning outcomes.

Further work is required to improve the model such that it better represents the sequential learning strategies of human participants, especially the variability among participants learning across the practice and testing sessions. In this respect, the use of dynamic rate selection for stochastic transitions in place of linear decrement in difficulty levels should be an interesting problem. The proposed work would benefit curriculum developers as it is an indicator of the level of effort required by different individuals to learn the route landmarks and navigation commands. Future work aims to further verify the GSPNRL model by comparing the agent results with human performance data from participants who received training for the same skills, but were taught using a markedly different pedagogical approach.

Acknowledgments

The authors acknowledge with gratitude the support of the NSERC-Husky Energy Industrial Research Chair in Safety at Sea, and the Canada Research Chair Program in Offshore Safety and Risk Engineering.

References

Allen, G. L. (1999). Spatial abilities, cognitive maps and wayfinding: bases for individual differences in spatial cognition and behavior. In R. G. Golledge (Ed.), *Wayfinding behavior* (pp. 46–80). The John Hopkins University Press.

Bause, F., & Kritzinger, P. S. (1996). *Stochastic Petri Nets: An Introduction to the Theory*. Verlag Viewweg.

BBC, A. N. (2013, October). *Nairobi siege: How the attack happened*. Retrieved from https://www.bbc.com/news/world-africa-24189116

Buckland, M. (2004). *Programming Game AI by Example*. Jones & Bartlett Learning.

Caduff, D., & Timpf, S. (2005). The Landmark Spider: Representing Landmark Knowledge for Wayfinding Tasks. *AAAI Spring Symposium - Technical Report*, 30–35.

Cullen, L. W. D. (1993). The public inquiry into the Piper Alpha disaster. *Drilling Contractor (United States)*, *49:4*.

Danial, S. N., Khan, F., & Veitch, B. (2018). A Generalized Stochastic Petri Net model of route learning for emergency egress situations. *Engineering Applications of Artificial Intelligence*, *72*, 170–182. https://doi.org/10.1016/j.engappai.2018.03.024

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. https://doi.org/10.1007/BF01386390

Eilam, D. (2014). Of mice and men: building blocks in cognitive mapping. *Neuroscience and Biobehavioral Reviews*, *47*, 393–409.

Emo Nax, B. (2014). *Real-world wayfinding experiments: Individual preferences, decisions and the space syntax approach at street corners*. Retrieved from http://discovery.ucl.ac.uk/1452725/

Filippidis, L., Galea, E. R., Lawrence, P., & Gwynne, S. (2001). Visibility Catchment Area of exits and signs. *InterFlam 2001: 9 Th International Fire Science & Engineering Conference*, 1529–1534. Retrieved from https://fseg.gre.ac.uk/fire/visibility_catchment_area.html

Gale, N., Golledge, R. G., Pellegrino, J. W., & Doherty, S. (1990). The acquisition and integration of route knowledge in an unfamiliar neighborhood. *Journal of Environmental Psychology*, *10*, 3–25.

Galea, E. R. (2003). Pedestrian and Evacuation Dynamics. *Proceedings of the 2nd International Conference on Pedestrian and Evacuation Dynamics, Greenwich, UK, 20-22 August 2003*. London, UK: CMS Press.

Galea, E. R., Xie, H., & Lawrence, P. (2014). Experimental and Survey Studies on the Effectiveness of Dynamic Signage Systems. *Fire Safety Science*, *11*, 1129–1143. https://doi.org/10.3801/IAFSS.FSS.11-1129

Goldschmidt, D., Manoonpong, P., & Dasgupta, S. (2017). A Neurocomputational Model of Goal-Directed Navigation in Insect-Inspired Artificial Agents. *Frontiers in Neurorobotics*, *11*, 20. https://doi.org/10.3389/fnbot.2017.00020

Golledge, R. G. (1999a). Human wayfinding and cognitive maps. In R. G. Golledge (Ed.), *Wayfinding behavior* (p. 9). Baltimore: The Johns Hopkins University Press.

Golledge, R. G. (1999b). Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes. In *Psycoloquy* (Vol. 10). Baltimore: The John Hopkins University Press.

Götze, J., & Boye, J. (2016). Learning landmark salience models from users' route instructions. *J. Locat. Based Serv.*, *10*(1), 47–63. https://doi.org/10.1080/17489725.2016.1172739

Gruszka, A., Hampshire, A., & Owen, A. M. (2010). Learned Irrelevance Revisited: Pathology-Based Individual Differences, Normal Variation and Neural Correlates. In A. Gruszka, G. Matthews, & B. Szymura (Eds.), *Handbook of Individual Differences in Cognition: Attention, Memory, and Executive Control* (pp. 127–144). https://doi.org/10.1007/978-1-4419-1210-7_8

Heiner, M., Herajy, M., Liu, F., Rohr, C., & Schwarick, M. (2012). Snoopy -- a unifying Petri net tool. *PETRI NETS 2012, June 2012, Lecture Notes in Computer Science*, *7347*, 398–407. Springer-Verlag.

IMO. (2009). *SOLAS: Consolidated Edition* (5th ed.). London, UK: International Maritime Organization.

Koutamanis, A. (1995). *Multilevel Analysis of Fire Escape Routes in a Virtual Environment*.

Kristiansen, S. (2005). *Maritime Transportation: Safety Management and Risk Analysis*. Amsterdam: Elsevier Butterworth-Heinemann.

Kumar, J. S., & Bhuvaneswari, P. (2012). Analysis of Electroencephalography (EEG) Signals and Its Categorization–A Study. *Procedia Engineering*, *38*, 2525–2536. https://doi.org/https://doi.org/10.1016/j.proeng.2012.06.298

Lee, S. A., Shusterman, A., & Spelke, E. S. (2006). Reorientation and Landmark-Guided Search by Young Children: Evidence for Two Systems. *Psychological Science*, *17:7*, 577–582.

Lynch, K. (1960). *The Image of the City*. Cambridge, MA: MIT Press.

Mackintosh, N. (1973). Stimulus selection: Learning to ignore stimuli that predict no change in reinforcement. In R. A. Hinde & J. S. Hinder (Eds.), *Constraints on Learning* (pp. 75–96). London: Academic Press.

McKinlay, R. (2016). Technology: Use or lose our navigation skills. *Nature*, *531*(7596), 573–575.

Muppala, J. K., & Trivedi, K. S. (1990). *GSPN Models: Sensitivity analysis and applications. Proceedings of the 28th Annual Southeast Regional Conference, Greenville, South Caroline. April 18-20, 1990*. Greenville, South Carolina: Association for Computing Machinery.

Nys, M., Gyselinck, V., Orriols, E., & Hickmann, M. (2015). Landmark and route knowledge in children's spatial representation of a virtual environment. *Frontiers in Psychology*, *5*, 1–15. https://doi.org/10.3389/fpsyg.2014.01522

OSHA. (2003). *Emergency-exit-routes-factsheet*. Retrieved from https://www.osha.gov/OshDoc/data_General_Facts/emergency-exit-routes-factsheet.pdf

Passini, R. (1977). *Wayfinding : a study of spatial problem solving with implications for physical design*. Pennsylvania State University, Pennsylvania, USA.

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Upper Saddle River, NJ, USA: Addison-Wesley.

Sharma, G., Kaushal, Y., Chandra, S., Singh, V., Mittal, A. P., & Dutt, V. (2017). Influence of Landmarks on Wayfinding and Brain Connectivity in Immersive Virtual Reality Environment. *Frontiers in Psychology*, *8*, 1220.

https://doi.org/10.3389/fpsyg.2017.01220

Smith, J. (2015). *The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios [Master thesis]*. Memorial University of Newfoundland, St. John's, NL, Canada. Retrieved from: https://research.library.mun.ca/8401/1/thesis.pdf

University of Greenwich, F. S. E. G. (2017). *EXODUS capabilities*. Retrieved from https://fseg.gre.ac.uk/exodus/exodus_products.html

Vandenberg, A. E. (2016). Human Wayfinding: Integration of Mind and Body. In R. H. Hunter, L. A. Anderson, & B. L. Belza (Eds.), *Community Wayfinding: Pathways to Understanding* (pp. 17–32). https://doi.org/10.1007/978-3-319-31072-5_2

Waller, D., & Lippa, Y. (2007). Landmarks as beacons and associative cues: Their role in route learning. *Memory & Cognition*, *35*(5), 910–924. https://doi.org/10.3758/BF03193465

Weinspach, P. M., Gundlach, J., Klingelhofer, H. G., Ries, R., & Schneider, U. (1997). Analysis of the Fire on April 11th, 1996; Recommendations and Consequences for Dusseldorf Rhein-Ruhr-Airport. *Staatskanzlei Nordrhein-Wstfalen, Mannesmannufer*, *1*.

Xie, H., Filippidis, L., Galea, E. R., Blackshields, D., & Lawrence, P. J. (2012). Experimental analysis of the effectiveness of emergency signage and its implementation in evacuation simulation. *Fire and Materials*, *36*(5–6), 367–382. https://doi.org/10.1002/fam.1095

Chapter 4

Situation Awareness Modeling for Emergency Management on Offshore Platforms‡‡‡

Co-authorship statement. A version of this chapter is accepted for publication in the journal *Human-centric Computing and Information Science* published *by Springer-Verlag*. The lead author, Syed Nasir Danial, has developed and implemented the model and extracted the empirical data using re-play video files for validation of the model. The co-author Jennifer Smith performed verification of the data extracted from the experiment. Co-authors Dr. Faisal Khan and Dr. Brian Veitch supervised the study. All authors read and approved the final draft.

Abstract. Situation awareness is the first and most important step in emergency management. It is a dynamic step involving evolving conditions and environments. It is an area of active research. This study presents a Markov Logic Network to model SA focusing on fire accidents and emergency evacuation. The model has been trained using empirical data obtained from case studies. The case studies involved human participants who were trained for responding to emergencies involving fire and smoke using a virtual environment. The simulated (queried) and empirical findings are reasonably consistent. The proposed model enables implementing an agent that exploits environmental cues and cognitive states to determine the type of emergency currently being faced. Considering each emergency type as a situation, the model can be used to develop a repertoire of situations for agents so that the repertoire can act as an agent's experience for later decision-making.

‡‡‡ Danial, S. N., Smith, J., Khan, F., & Veitch, B. (2019). Situation awareness modeling for emergency management on offshore platforms. *Human-centric Computing and Information Sciences* (Accepted for publication).

4.1 Introduction

The present work proposes a model based on Markov Logic Network (MLN) (Domingos & Richardson, 2007) for representing emergency situations involving smoke and fire on offshore petroleum platforms. The model is tested for two important situations, `FIRE` and `EVACUATE`. In the `FIRE` situation, fire is observed due to smoke at some place on the platform, and all workers need to muster to their primary muster station. In the `EVACUATE` situation, the fire is escalated so that some escape routes to the primary muster station are blocked and all personnel need to muster at the lifeboat or alternative muster station. The purpose of this work is to have a model that can be used by a software agent so that the agent can exhibit human-like situation awareness (SA). Such agents can subsequently be used, for example, in training simulators to enrich trainees' experience by showing them various scenarios in which the agent shows recognition of different situations (to makes various decisions). A participant can learn from the agent what information is important in a given scenario for correct SA.

Representing the emergency response of agents operating in a VE is a challenging and active research area. Emergencies on board can arise from several factors, among which accidents are on top (B. Khan, Khan, Veitch, & Yang, 2018). The Cullen Report (Cullen, 1993) following the *Piper Alpha* disaster has clear recommendations for operators to perform a risk assessment of ingress of smoke or gas into the accommodation areas. G. Klein (1998 p. 219) says that VE training is important for the crew in many respects, for example, because trainees get opportunities to learn from and about each other as a team, and also to learn about the cues that unfold in an evolving training scenario. Thus, a VE has an essential role as a training environment, and agents are important elements of VE fidelity (Luck & Aylett, 2000).

Situations are highly structured parts of the world that span a limited space and time, and people talk about them using language. They are composed of objects having properties such that the objects stand in relation with one another (Barwise & Perry, 1980). An agent's world can be considered as a collection of situations, and the agent should be able to discriminate among them. Devlin (1991b) extends Barwise and Perry's *Situation Theory* (Barwise, 1981; Barwise & Perry, 1983) and proposes a representation using a concept called *infon*, which is an informational item of the form "objects a_1, …, a_n do/do not stand in the relation P". A situation, formally, is then some part of the world that is supported by a set of *infons*.

This work considers SA as being a phenomenon that refers to the information flow (Devlin, 1991a) from a situation to a subject such that the subject can reason about the situation. Endsley's (1988) model of human SA describes this information flow as a process with three successive levels. Level-1 begins when a person starts perceiving information as environmental cues. This part of Endsley's SA model has a direct resemblance with acquiring information about the presence of object a_1… a_n for developing relevant *infons* in a situation. Level-2 in Endsley's model explains that the person should be able to extract meaning from what has already been perceived. Level-3 of the model says that the meaning of cues should enable a person to foresee something shortly. Kokar, Matheus, and Baclawski (2009) developed an ontology, called situation theory ontology (STO), that defines semantics for situation theory by including a meta-class describing the types of things (individuals, individual's properties and relations among them) that constitute a situation as a type in accord with Barwise and Devlin's situation semantics. Inference on the available facts (*infons*) with some background knowledge about the objects and their relations within the ontological framework not only supports level-2 of Endsley's SA model but also gives potential to achieve level-3 SA. For example, if an agent knows that fire lit in an oil container should not be put out with water, only then can the agent preempt somebody from doing so. For that, the agent should project the current information about the position of the fire and the water source approaching the oil container into a future state using a rule that exploits some predicate like `fireEscalates(oil, water)`. STO satisfies many characteristics of Endsley's SA model, and it was implemented in the Web Ontology Language (OWL) using the full profile (OWL-Full). Now that OWL changed in 2009 and the support for OWL-Full, which is required to fulfill the theoretical requirements of Barwise and Devlin's approach to situation modeling, is unavailable, STO is difficult for use as a platform for modeling SA.

The concept of *context* in the literature related to AI is similar to the *situation* in the SA literature. Sowa (1984, 2000) uses conceptual graphs (CG) to represent context or situations. CGs are an extension of Peirce's existential graphs (c. 1882) with features taken from semantic networks of AI and linguistics. CGs are bipartite graphs where boxes are used to represent concepts, and circles are used to show relations. As a simple example, a situation "Cat is on mat" can be represented in a CG using a linear notation as: `[Cat]`→`(On)`→`[Mat]`, where `Cat` and `Mat` are two concepts (each for one object/individual in the real world) related to each other by the relation `On`.

Sowa (2000), and Akman and Surav (1996) say that both *context* and *situation* are the same notions. Kokar et al. (2009) report that contexts (situations) in AI are dealt with using *predicates* such as `isa(c, p)` to mean that the proposition `p` holds `true` in the context `c`.

Predicates in FOL are building blocks of the system based on it. CG is computationally equivalent to FOL (Sowa, 2000). Rules in FOL are considered as hard constraints in that a world is thought to exist only when the rules are valid. This is contrary to situations in real life. A rule like *smoke causes cancer* in FOL is always valid, so an agent that smokes certainly has cancer. But this is not the situation in the real world where rules are violated, and the violation is only a matter of limitation regarding the frequency of cases where the rule is not observed.

(Domingos & Lowd, 2009) consider FOL rules as hard constraints that limit the progress in AI research, and offer a method to describe *soft rules* using MLNs. Soft rules are formed by assigning weights to the FOL rules in MLNs. The weights determine how likely the entities of the world might follow a rule. The higher the value of the weight, the harder the rule becomes. The present work uses MLNs to construct a model for situations in emergency scenarios, particularly those arising on offshore petroleum platforms. The purpose is to create software agents for training in VEs, where an agent exploits environmental cues to understand different emergency situations. This way, the agent can be given an ability to construct a repertoire of situations that it observes. Such agents can be expected to make experience-based decisions when exposed to emergencies in a solo or a group training environment. Applications of such agent models can be found in many fields, including pilot behavior modeling (Hu, Li, & Zhang, 2018) during midair encounter, game programming, and so on.

Being aware of a situation is not merely an outcome of a typical feature matching mechanism, as some authors suggest (Nowroozi, Shiri, Aslanian, & Lucas, 2012). Awareness helps categorization of things according to certain common grounds. In other words, recognition of a situation, should mean first, to model a situation using a knowledge representation schema, and second, to devise a mechanism whereby inference can be performed on the stored knowledge to extract new knowledge. Since MLNs support inference — even on incomplete data — the resulting model of SA has some resemblance to Endsley's SA model. Moreover, as MLNs allow conflicting rules, it is a more natural choice for modeling situations in which cues at different times and space could take different meanings.

Social agents can interact with human participants during an emergency egress scenario to form a group-training situation to learn from human responses and then to guide other computing modules for evaluation of human responses. Participants can also learn from these agents to respond in a scenario. The use of these agents in training exercises reduces the necessity of having a large number of real people in a large-scale group training (Nakanishi, Shimizu, & Isbister, 2005). Also, the rehearsals with agents are more effective than with human counterparts because of the consistent, usually scripted, agent behavior. A more realistic approach is to replace the scripted agent's behavior to more natural, human-like behavior so that a participant can trust the agent responses and may consider it a colleague, rather than a robot. The works in (Danial et al., 2018; Danial, Smith, Khan, & Veitch, 2019) focus on route learning for agents and propose a model where an agent can exhibit behavior that is similar to a human participant while learning a new escape route. Risks associated with human responses during an evolving emergency are assessed in (Norazahar, Smith, Khan, & Veitch, 2018). The authors assert that hazards (like fires, smoke), weather condition, malfunctioning equipment, and inadequate emergency preparedness such as that related with the recognition of platform alarms are important factors that affect the human response. Musharraf, Smith, Khan, Veitch, and MacKinnon (2018) propose a methodology to account for individual differences in agent modeling for emergency response training. The problem of modeling SA for such agents is still another important area that has potential implications in the way agents make decisions in evolving emergencies. (Chowdhury, 2016) explores various situations that occur on offshore rigs, platforms, and installations. The author explains how fire and evacuation situations are indicated on different platforms.

Section 4.2 describes some recent work in situation awareness. Section 4.3 describes the proposed methodology to model SA based on MLN. Section 4.4 describes a case study and experimental results that serve to assess the validity of the proposed model. Section 4.5 contains a discussion of the results, and Section 4.6 presents concluding remarks and future directions.

4.2 Previous works

With the increasing demand of intelligence-based systems, encompassing from smart cars to smart homes, the use of situation recognition has become a focal point in research because of its importance in enabling artificial intelligence. Récopé, Fache, Beaujouan, Coutarel, and Rix-Lièvre (2019) attempt to discover the reasons for interindividual differences in volleyball players' defensive behavior during different identical situations. The authors raised an important question, "Might other dimensions of situation assessment, which have so far not been studied to any great extent, be involved?" Based on an experiment involving two volleyball teams, the authors conclude that an individual's activity is governed by a specific norm that organizes, orients and enhances understanding of the actions as a coherent totality. In other words, there is a subconscious sensemaking that individuals use in order to determine the relevance of cues corresponding to different situations.

In order to assess network security within an *Internet of Things* (IoT), Xu, Cao, Ren, Li, and Feng (2017) propose an ontology-based model for SA for network security of IoT. Again, ontological knowledge helps identifying concepts and relations in order to understand what type of situation is currently being observed. An IoT security situation is described by employing knowledge about the context, attack, vulnerability, and network flow. A model of how SA spreads among agents in a multiagent system is presented in (Bosse & Mogles, 2014). Nasar and Jaffry (2018) study this work (Bosse & Mogles, 2014) and extend it, using Agent Based Modeling (ABM) and Population Based Modeling (PBM) techniques, by incorporating *trust* in the SA model. Thus, the resulting agents' beliefs and decisions about the environment have been shown to be affected by their trust in other agents. Johnson, Duda, Sheridan, and Oman (2017) addressed the issue of decrease in SA when the flight control mode changes from automatic to manual mode. The authors proposed a cognitive model based on "perceive-think-decide-do" scheme that estimates the effects of change in the flight mode on operator behavior. The primary contribution of the proposed model is an attention executive module, which is responsible to detect changes in attention on specific control loops based on changes in priorities. The authors of (Kingston, Nurse, Agrafiotis, & Milich, 2018) develop a model that uses social media posts and process them, by clustering consistent posts, in the way that a user can gain more better insights by reading different views (or world view) that the system has generated. This approach is not particular to model situation awareness for agents; however, people can assess a situation, described through posts, better by reading the world views about the posts on tweeter or any other social media platform that exploits the proposed technique.

Yang, Wang, Zeng, Yue, and Siritanawan (2019) develop a probabilistic model for robots to decide about a role that otherwise would have been fulfilled by a human had there been the same situation. Situations are classified here as: *easy*, *medium*, and *hard*. The model takes input as 2D and 3D images, and the robot model should get its role first, and then decides upon actions per role and the situation as recognized through the images. Roles are recognized by fusing the results of two indicators, the distance-based inference (DBI), and the knowledge-based inference (KBI). The DBI uses a relative distance between humans and mission critical objects to determine the probability of a possible role. The KBI uses a Bayesian network that integrates human actions and object existence to determine a possible role. The final role is determined as a fusion of DBI and KBI by using information entropy measure. The actions of a person that is detected as target, because he is carrying the mission critical object, is a major contributor of changes in the situation. Situation levels are determined by using the target person actions (moving, stationary) and the relative position of several mission related entities at some time t by using a Bayesian network. Actions are decided based on the situation level and the inferred role. The proposed approach is robust in recognizing roles because of the fusion of different inference results, it would be useful if situations to be encountered are of fundamentally the same type, so that they can be classified as easy, medium, and hard. For example, what would a robot do if the situation is complex, as is the case of an offshore emergency where the environment is cluttered with many objects, crew, alarms, exit signs, announcements, and so on. In such conditions, different situations are possible, and the question of classifying a situation into easy, medium, and hard seems an idealistic assumption. Hu et al. (2018) developed model for predicting pilot behavior during midair collision recognition-primed decision model. Features extracted from the environment are compared with the stored attributes of situations, and an already encoded situation is retrieved based on a Bayesian classifier as a similarity criterion.

Naderpour, Lu, and Zhang (2014) developed a cognition-driven SA support system for safety-critical environments using Bayesian networks. The system consists of four major components to deal with (1) receiving cues from environments, (2) assessing situation based on dynamic Bayesian network and fuzzy risk estimation method, (3) recovering from a situation, that advises measures to reduce the risk of a situation, and (4) an interface for better interaction with people. Another study (Snidaro, Visentini, Bryan, & Foresti, 2012) categorizes maritime anomalies, such as speeding of a vessel, according to the levels in the JDL data fusion model (Llinas et al., 2004). Szczerbak, Bouabdallah, Toutain, and Bonnin (2013) use conceptual graphs to represent ordinary real-world situations and introduce a method to reason about similar situations. Liu, Deng, and Li (2017) propose an information fusion model with three layers for event recognition in a smart space where sensory data is collected in the first layer, context is represented as MLN in the second layer. The third layer maps the contextual information of the second layer to corresponding events. To fuse uncertain knowledge and evidence Snidaro, Visentini, and Bryan (2015) develop an MLN based SA model for maritime events.

Gayathri, Easwarakumar, and Elias (2017) use MLN to develop an ontology that can be used to recognize activities in smart homes. The purpose is to detect an abnormal activity (or a situation) and inform the remote caretaker. Using a technique called *Event Pattern Activity Modeling* (Gayathri, Elias, & Shivashankar, 2014), observations collected through sensors have been parsed into concepts in an ontology, and the relevant descriptive logic rules are generated. These rules are then converted into FOL equivalents, and weights are assigned to FOL rules to develop the MLN based activity model. Given the observations through sensors, the MLN activity model can be used to suggest different interpretations of the observed data in a probabilistic sense. The use of MLNs enable representation of cyclic dependency among the rules, which is a major advantage of MLNs over Bayesian networks.

4.3 A method to model situation awareness

Take S to be a countable set and $\wp(S)$ to define the set of all subsets of S, where the points of S are sites, each of which can either be empty or occupied by an object (such as a *formula* in a logical framework or a particle as it appears in the statistical mechanics literature). The sites of S can be represented by binary variables $X_1, X_2, \ldots, X_n$. The subset $\Upsilon \in \wp(S)$ is regarded as describing a situation when the points of Υ are occupied and the points of $S - \Upsilon$ are not. The elements of $\wp(S)$ are sometimes called configurations. The set S, representing the sites, may have some additional structure. As sites are connected, S can be considered as forming an undirected graph G (Preston, 1974), so the points of S are the vertices of some finite graph $G(S, E)$, where E is the set of edges. The present work involves modeling a probability measure (defined in the following subsections), restricted to the sample space $\Sigma = \{0, 1\}^S$, having a kind of *spatial Markov property* given in terms of neighbor relations of G (Grimmett, 2010), called a Markov random field (Isham, 1981; Kindermann & Snell, 1980; Pearl, 1988).

Definition. $G(S, E)$ is countable and does not contain multiple edges and loops. If x, $y \in S$ and there is an edge of the graph G between x and y, then x and y are considered neighbors of each other (Preston, 1974). Formally, the function f: $S \times S \rightarrow \{0, 1\}$ is given by

$$f(x,y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are neighbors,} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Definition. If $\Upsilon \in \wp(S)$ then the boundary $\partial\Upsilon \in \wp(S)$ is defined as:

$$\partial\Upsilon = \{y \in S - \Upsilon \mid f(x,y) = 1, \text{ for some } x \in \Upsilon\} \tag{4.2}$$

A Markov network (MN) is composed of 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[12] in 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, …, X_n can be developed to understand the influence of a site, i.e., a variable, on its neighbors (Raedt, Kersting, Natarajan, & Poole, 2016) as defined below:

[12] A clique of a graph G is a *complete subgraph* of G.

$$P(X = x) = \frac{1}{Z} \prod_{k} \phi_k(x_{[k]}) \tag{4.3}$$

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, $Z = \sum_{x \in \Omega} \prod_k \phi_k(x_{[k]})$.

4.3.1 Markov Logic Network

Because a random variable assigned with a value can be considered as a proposition (Halpern, 2003 p. 58), Domingos and Richardson (2007) define MN by first considering the variables as rules/formulas in FOL. Unlike FOL, a formula in MLN is assigned a weight (a real number), not just the Boolean `true` or `false`. Formally, an MLN U is defined as a set of pairs (F_i, w_i) with F_is being the formulas and w_is being the weights assigned to the formulas.

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

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i\, n_i(x)\right) = \frac{1}{Z} \prod_i \phi_i(x_{[i]})^{n_i(x)} \tag{4.4}$$

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 $\phi_i(x_{[i]}) = e^{w_i}$.

4.3.2 The `FIRE` and `EVACUATE` emergency situations

Fire and evacuate are among the important types of emergencies that occur on offshore petroleum installations (Spouge, 1999). Chowdhury (2016 pp. 176-177) describes various emergencies, such as fire/blowout, evacuate, H_2S release, and the types of alarms used on different offshore rigs. A fire may erupt due to many reasons, such as a gas release near an igniting source, or an electrical spark near a fuel line. Explosions also result in fires. In any case, if a fire event occurs a fire alarm is raised, and people on board must leave their work and report to their designated muster station, which is usually their primary muster station. This type of situation is called a `FIRE` situation, and it will end when an *all-clear* alarm sounds, which means that the fire has been taken care of and the people can now return to their duties. In case a `FIRE` situation escalates, meaning that the fire spreads and blocks various paths so that personnel's safety could be further compromised, an `EVACUATE` situation may come into effect, and this new situation is communicated to people by another alarm, different from the fire alarm. In the `EVACUATE` situation, people must report to their designated secondary muster station, the lifeboat station, from where the final evacuation from the platform can proceed.

4.3.3 Knowledge representation of emergency situations

An interesting aspect of modeling a situation is to identify the factors that lead to the situation of interest. Typically, a situation involves preconditions or events, some of which are observable, and some are not directly visible (Snidaro et al., 2015). Since MLNs are based on FOL rules, the basic methodology as described in (Domingos & Lowd, 2009; Domingos & Richardson, 2007), and followed here, requires developing FOL rules, followed by assigning the weights, and finally performing the required inference. Nonetheless, there is no straightforward way of writing FOL rules for a knowledge domain. Writing FOL rules requires experience and thorough domain knowledge. Also, the developed FOL rules must fulfill some criteria of acceptance. For example, a rule like "smoke causes cancer" has been given serious attention among medical practitioners (Cornfield et al., 2009) since the constitution of a study group in 1957 ("Smoking and Health: Joint Report of the Study Group on Smoking and Health," 1957). This group was appointed by several institutes, including the National Cancer Institute, and it concludes, by considering the scientific evidence, that cigarette smoking is a causative factor for a rapid increase in the incidence of human epidermoid carcinoma of the lung.

Figure 4.1 proposes a methodology that incorporates the basic steps of constructing an MLN (Domingos & Lowd, 2009; Domingos & Richardson, 2007) iteratively so that each rule could be judged against some heuristic criteria of acceptance, for example, by assigning the weights to rules through empirical findings using a learning algorithm (Singla & Domingos, 2005) and then seeing if the weights make sense. In any case, if many of the rules come up as negatively weighted, then such a knowledgebase will have little practical value, and one must look into the training samples and/or the rules themselves. In the former case, it is possible that the training sample includes little evidence where the rules were successful. In the latter case, it is possible that the rules were not constituted correctly, regarding the specification of different predicates, their connections using logical connectives, and their implication into a consequent. In short, one must go back and update the rules and/or training-testing data sample, as shown in Figure 4.1 until the desired results are met. The choice of a learning algorithm is also a point to consider. Since discriminative learning does not model dependencies between inputs within the training sample, it often produces results (Singla & Domingos, 2005) better than generative learning techniques. Using the testing samples as evidence, the probability that a query

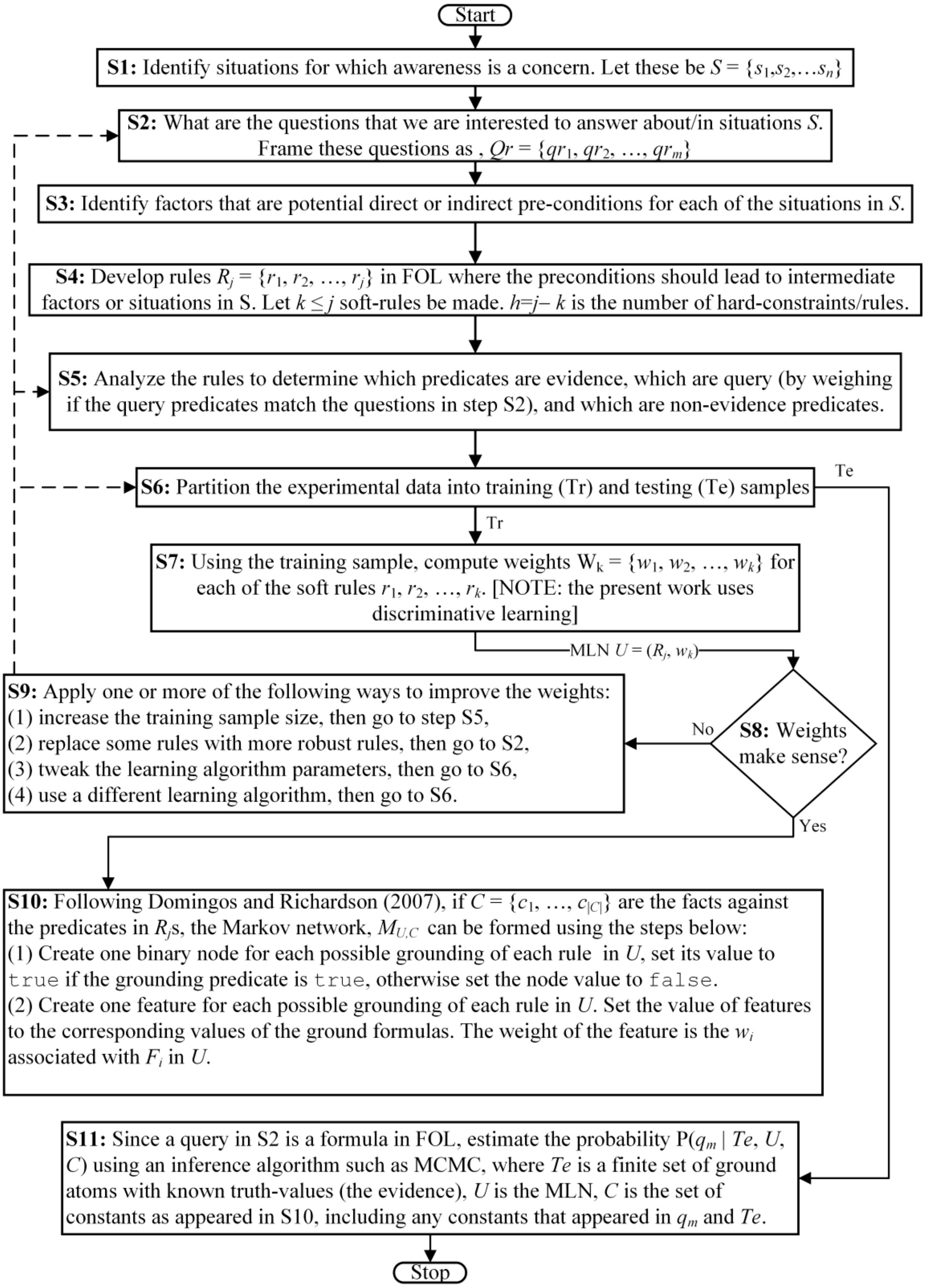


Figure 4.1. The proposed methodology to develop a situationally aware agent model based on MLN.

Table 4.1. Variable/predicate names and description

Variables	Predicate name	Parameter types	Description
Listens	`L`	(agent, alarm, time)	An *agent listens* to an alarm during time interval *time*.
Recognizes	`R`	(agent, alarm, time)	An *agent* recognizes an *alarm* during *time*.
HasIntentToReach	`HITR`	(agent, musterLoc, time)	An *agent* has intention to reach a muster location during *time*.
HasEmrgSit	`HES`	(agent, emgSitType, time)	An *agent* has an *emergency situation* during *time*.
SeesThreat	`ST`	(agent, threatType, time)	An *agent* sees a *threat* during *time*.
HasFocusOn	`HFO`	(agent, pa, time)	An *agent* has focus on a PA during *time*.
HasSomeEmrgSit	`HSES`	(agent)	An *agent* gets a sense of some emergency situation.
FollowsPA	`FPA`	(agent, pa, time)	An *agent* understands and follows a PA during *time*.
KnowsEmrgTypeOfAlarm	`KETA`	(emgSitType, alarm)	An agent knows which *alarm* is used in a given *emergency type*.
KnowsEmrgTypeOfThreat	`KETT`	(threatType, emgSitType)	An agent knows which *threat type* would give rise to a particular *emergency situation*.
KnowsEmrgTypeOfPA	`KETPA`	(pa, emgSitType)	An agent knows what *emergency situation* is being announced in PA.
BeforeSeeingThreat	`BST`	(agent, alarm, time)	BST is paired with HITR with logical 'and' connective to mean that HITR is `true` only when the agent has determined the muster location before seeing a threat.

predicate holds is estimated by employing an inference mechanism, such as by using the MC-SAT algorithm (Poon & Domingos, 2006).

Table 4.1 lists the variables studied in this work for SA about the situations discussed earlier in Section 4.3.2, the `FIRE` situation, which asks all personnel to move to the primary muster station, and the `EVACUATE` situation, which involves escalation of a

Table 4.2. The FOL rules that are showing the knowledge base for basic emergency preparedness.

#	Rules
1	`¬L(ag,al,τ)=> ¬R(ag,al,τ).`
2	`L(ag,+al,τ)^HITR(ag,+mloc,τ)^BST(ag,+al,τ)=> R(ag,+al,τ)`
3	`L(ag,al,τ)=>HSES(ag)`
4	`ST(ag,thrt,τ)=>HSES(ag)`
5	`(HFO(ag,+p_a,τ)^FPA(ag,+p_a, τ)^KETPA(+p_a,+eTyp))v (ST(ag,+thrt,τ)^KETT(+thrt,+eTyp))v(L(ag,+al, τ)^ HITR(ag,+mloc, τ)^KETA(+al,+eTyp)^BST(ag,+al, τ)) => HES(ag, +eTyp, τ)`
6	`HES(ag,FIRE, τ)=> ¬HES(ag,EVACUATE, τ)`
7	`HES(ag,EVACUATE, τ)=> ¬HES(ag,FIRE, τ)`
8	`HES(ag,FIRE, τ0)^HES(ag,EVACUATE, τ1)^ Gt(τ1, τ0)=>¬HES(ag,FIRE, τ1)`

fire into a larger fire that obstructs the primary escape-route leading to the primary muster station, thereby necessitating re-routing to the alternative or lifeboat station. A set of FOL rules are proposed in Table 4.2 so that an agent recognizes these situations like the way a human counterpart recognizes them. The preconditions (antecedents of FOL rules) used here are common among experts and have been suggested in earlier studies (Chowdhury, 2016; ExxonMobil, 2010; Proulx, 2007; J. Smith, 2015; Sneddon, Mearns, & Flin, 2013; Spouge, 1999; Thilakarathne, 2015; Tong & Canter, 1985; Tutolo, 1979; Wankhede, 2017). The query predicates determine the probability of recognizing alarms, having a `FIRE` situation, having an `EVACUATE` situation, and having some (unknown) situation given the evidence predicates.

4.3.4 Reasoning

The variability in the emergency alarm systems and indicators used at different offshore installations is a source of confusion when a real emergency occurs, especially for personnel who frequently move from one to another platform for performing special tasks. Alarm recognition is considered a major contributor to the awareness of an emergency type (Chowdhury, 2016). Different alarms mean different situations requiring a different course of actions by the personnel onboard. The scope of the present work is limited to SA and does not extend to finding a suitable course of action in case of an emergency. Recognition of alarms is something that cannot directly be observed unless the person is asked, so a search for further factors that indicate that an alarm has been recognized is required. An alarm cannot be recognized if it was not heard, whereas listening needs attention towards the alarm signal (Reason, 1990). Emergency alarm signals are so loud that it is hard not to hear them, but that does not mean that people will always recognize which situation the present alarm is for. An agent can exploit rule # 1 in Table 4.2 to express the behavior of *not recognizing* an alarm if, for any reason, such as the inertial tendency of people to keep doing what they are doing (Winerman, 2004), the agent does not listen to it. Several studies (Proulx, 2007; Tong & Canter, 1985) show that people do not start evacuating a building or moving to a muster location automatically when they hear alarms unless they are trained to do so, and there are some other factors or cues that lead them to act as needed in that situation.

Rule#2 uses two more factors to frame the conclusion of recognizing an alarm beside just listening. The first factor reflects a person's ability to develop the intention of moving to the required muster station. The required muster station is referred to by the variable `mloc` that takes values from the set `{MESSHALL, LIFEBOAT}`. Literature shows that intention is an important cognitive state that affects one's ability to participate in a decision-making process (Bratman, 1987; Thilakarathne, 2015).

Intention is modeled here as a predicate `HITR` that takes a value `true` if the agent develops the intention to move to `mloc` during a time interval τ. An agent's intention can be inferred by observing which route is taken up immediately after listening to the alarm. The agent can also be delayed in developing the intention to reach `mloc` and may require other cues for building up this intention. Therefore, to know if an alarm is recognized without the help of other cues, such as observing smoke, it is necessary to know when the agent develops the intention of moving to the required muster station after listening to an alarm. `HITR` is used in conjunction with the predicate `BST` that ensures the intention of moving to the muster location is developed before seeing a threat because if an agent sees a threat, it would be unclear if its intention of moving to `mloc` is due to the threat or the alarm. The probability of recognizing the alarm is determined by using the conjunction of the three predicates. If any of the antecedent predicates fail, the chances of recognizing the alarm will be reduced.

The variable `ST` (see Table 4.1) is used to indicate that the agent observes a threat. An agent who sees a threat (such as smoke or blowout) is highly likely to discover the type of emergencies involved (`FIRE` or `EVACUATE`). Rules # 3 and 4 say that an agent will be aware of 'some' emergency if it just listens to an alarm or observes a threat.

Public address (PA) announcements are also important cues for getting to know details about a developing situation (Chowdhury, 2016; ExxonMobil, 2010; Spouge, 1999; Wankhede, 2017). PAs are verbal announcements with clear words detailing the situation. The details include the location of a threat or hazard, what actions are needed, and what areas are affected. The agent can take advantage of the PA to learn about a developing emergency. However, this needs a focus on the words in the PA. The literature on distraction explains how people get distracted in different situations. Tutolo (1979) says that children's ability to listen without being distracted improves with age. Inattention to the available information has been studied for the offshore drilling environment in (Sneddon et al., 2013). The authors discuss other factors, such as stress, that influence focus of attention by producing a narrowing or tunneling effect so that a person is left focusing on only a limited number of cues under some stressors. Tversky and Kahneman (1974) call this *cognitive tunnel vision*. 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 actions according to the PA is considered to have focused 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 responding to the PA. In general, gestures can be noticed to determine if an agent has a focus on an ongoing PA or not. 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 focused on the PA's words, it is confused or does not have an 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`. Rules # 6 & 7 are to ensure that `FIRE` and `EVACUATE` are two distinct types of situations, besides that `EVACUATE` may occur because of a fire (Chowdhury, 2016; Spouge, 1999).

Rule # 8 says that if during some initial time interval $\tau 0$ a FIRE situation is observed, and during some later interval $\tau 1$ (where $\tau 0 < \tau 1$) this situation escalates to EVACUATE, then the FIRE situation will no longer exist during $\tau 1$, although one may witness real fires during the EVACUATE situation.

4.4 Case studies: SA during offshore emergency scenarios

This work uses two case studies developed using the experiment performed in (J. Smith, 2015) to acquire training and testing data for SA during offshore platform egress scenarios so that the proposed model (in Table 4.2) can be judged against the empirical data. The objective of Smith's experiment was to assess VE training effects

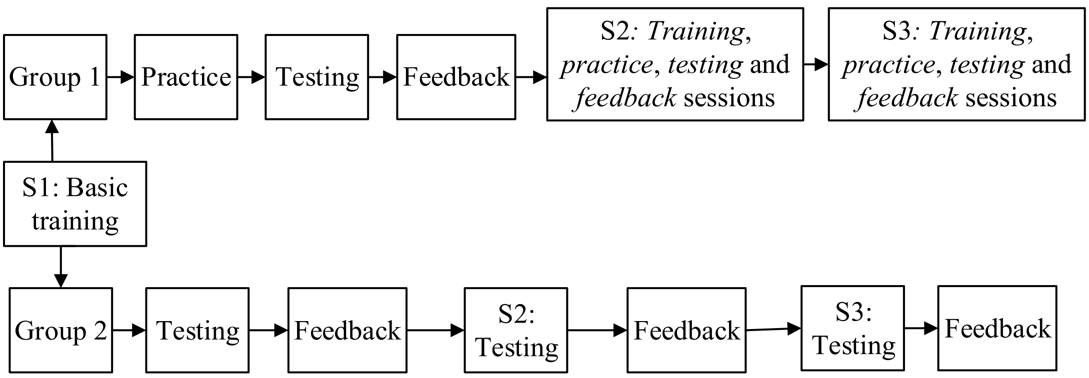


Figure 4.2. Training exposure to participants. Sessions S1, S2, and S3. The datasets are obtained from S3 for both groups. Source: Adopted from (Smith, 2015).

on people's ability to learn and respond during offshore egress scenarios involving fire hazards. The distribution of training of the participants and testing their performance is shown in Figure 4.2. The experiment targeted six learning objectives: (1) establish spatial awareness of the environment, (2) routes and mapping, (3) emergency alarm recognition, (4) continually assess situation and avoid hazards on route, (5) register at temporary refuge, and (6) general safe practices such as closing the doors when there is an emergency alarm in effect due to fire or smoke hazard. There were three sessions with increasing complexity. Session 1 (S1) involved training, practice, and testing for the learning objectives 1, 2, 5 & 6, session 2 (S2) used scenarios involving the learning objectives 3, 5 & 6, and session 3 targeted the objectives 3, 4, 5 & 6. The experiment involved 36 participants divided into two groups: Group 1 contained 17, and Group 2 contained 19 participants. Group 1 was trained in several sessions, whereas Group 2 participants received only a single training session. The VE used in this experiment was All-hands Virtual Emergency Response Trainer (AVERT). AVERT is a research simulator of an offshore petroleum facility (see Appendix B.1). It is used to train participants to improve their response should they face an emergency such as a fire or an explosion. The present work uses only the third and the fourth learning objectives because they deal with the SA the participants exhibited during each scenario. The data was obtained by a careful reading of the log files and watching the replay videos of session S3 recorded for each participant during the testing phase of the relevant scenarios.

4.4.1 Situations in Experimental Scenarios

Smith's experiment (Smith, 2015) involves emergencies in which, initially, there is a fire in the galley. After some time, the fire escalates so that the primary muster station, which is the mess hall on deck A of the platform, becomes compromised. An audible fire alarm (the General Platform Alarm, `GPA`) followed by the relevant PA is made right after the initial fire event. The escalation of the fire in the galley to fire in the mess hall is then announced by a Prepare to Abandon Platform Alarm (`PAPA`), followed by another PA. Initially, a participant is situated in their cabin (see the floor

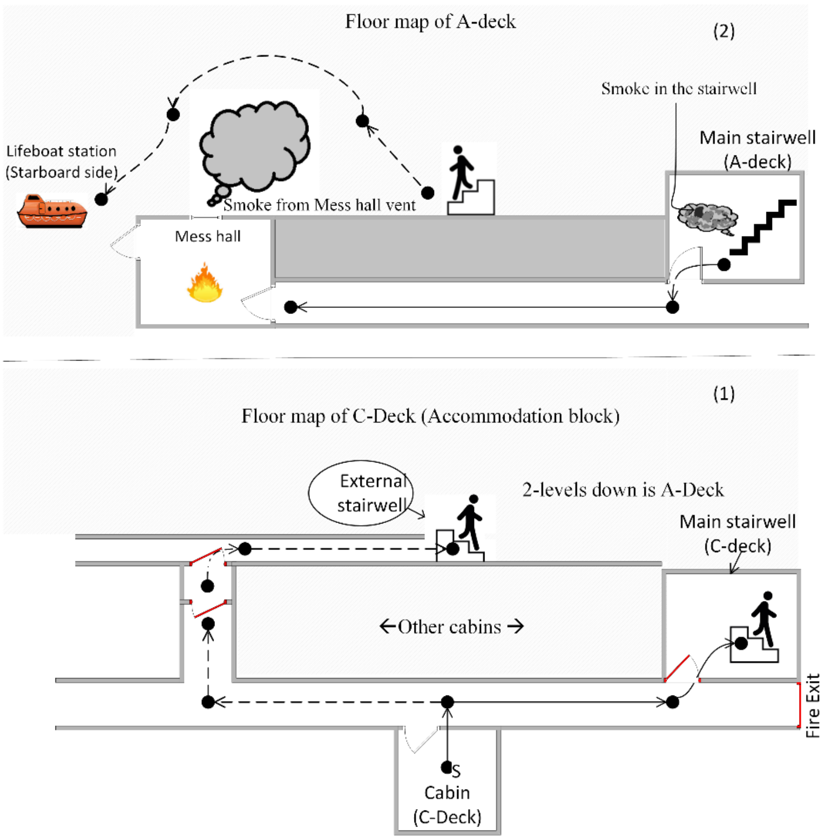


Figure 4.3. Floor map for decks A and C in AVERT simulator. A participant starts from Cabin (S) in part (1) and ends either at the mess hall or the lifeboat station in part (2) using external stairwell or main stairwell. The dotted lines show the alternate route, and the solid lines refer to the primary route.

map in Figure 4.3-1) when a GPA alarm activates, followed by a platform announcement. The PA announcement directs the participant to muster at their designated muster station, which is the mess hall on A-deck for a FIRE situation. Upon hearing the GPA, the participant needs to move out of the cabin and choose from the primary route (the solid lines, which goes through the main stairwell), or the secondary escape route (the dotted lines, which uses the external stairwell) to reach A-deck. The participants were trained to deal with these situations earlier using escape route training videos and instructions in the training session S1. While moving toward the mess hall, after a fixed interval of time $\tau 0$, the participant receives a call to abandon the platform. This is the PAPA alarm, which indicates to the participants that they should immediately move to the secondary or alternative muster location, which is the lifeboat station at the starboard side of the platform (see Figure 4.3-2). The time interval when PAPA is activated to the end of a scenario is termed $\tau 1$. Thus, $\tau 0$ is the time interval in which the participants get all cues related with the FIRE emergency, such as smoke in the stairwell, GPA alarm, and PA announcement that includes the words "fire in the galley". Similarly, $\tau 1$ is the time interval that starts when $\tau 0$ expires and ends at the end of the scenario. During the $\tau 1$ period, the participant receives cues related with an EVACUATE situation. The PAs use clear words as to what needs to be done in an emergency and what parts of the escape route are expected to be blocked due to fire or smoke. Although GPA and PAPA are activated at different times, indicating two different situations, the other environmental cues can be observed at any time during their lifetimes. For example, smoke in the main stairwell is considered as a cue for a FIRE situation. Some participants reached at this spot in the

Table 4.3. A sample of validation data for two participants, P1G1 and P2G1. A 'Y' before a list of parameter values means that the agent has observed these values, an 'N' means that these values have not been observed empirically. LFB stands for LIFEBOAT station.

Predicates	Parameters	
	P1G1	P2G1
L	Y: P1G1, GPA, τ0; Y: P1G1, PAPA, τ1;	Y: P2G1, GPA, τ0 Y: P2G1, PAPA, τ1
HML	Y: GPA, MSH; Y: PAPA, LFB;	Y: GPA, MSH Y: PAPA, LFB
HITR	Y: P1G1,MSH, τ0; Y:P1G1, MSH, τ1; Y:P1G1, LFB, τ1; N: P1G1, LFB, τ0;	Y:P2G1,MSH, τ0 Y:P2G1,LFB, τ1
R	Y: P1G1, GPA, τ0; N: P1G1, PAPA, τ1;	Y: P2G1, GPA, τ0 Y: P2G1, PAPA, τ1
HSES	Y: P1G1;	Y: P2G1
ST	Y:P1G1, SMK_MSHA, τ1; Y:P1G1, SMK_STAI, τ1; Y: P1G1, SMK_VENT, τ1	Y:P2G1,SMK_VENT, τ0
HES	Y: P1G1, FIRE, τ0; Y: P1G1, FIRE, τ1; Y: P1G1, EVACUATE, τ1	Y: P2G1,FIRE, τ0 Y:P2G1,EVACUATE, τ1
HFO	Y: P1G1, PA_GPA, τ0; N: P1G1, PA_PAPA, τ1	Y: P2G1,PA_GPA, τ0 Y: P2G1,PA_PAPA, τ1
FPA	Y: P1G1, PA_GPA, τ0; Y: P1G1, PA_PAPA, τ1;	Y:P2G1,PA_GPA, τ0 Y: P2G1, PA_PAPA,τ1
KETA	Y: GPA, FIRE; Y: PAPA, EVACUATE;	Y: GPA, FIRE Y: PAPA, EVACUATE
KETT	Y: SMK_VENT,EVACUATE; Y: SMK_STAI,FIRE; Y:SMK_MSHA, EVACUATE;	Y:SMK_VENT,EVACUATE Y:SMK_STAI,FIRE Y:SMK_MSHA,EVACUATE
KETPA	Y: PA_GPA, FIRE; Y: PA_PAPA,EVACUATE;	Y: PA_GPA, FIRE Y:PA_PAPA, EVACUATE
Greater	Y: τ1, τ0;	Y: τ1, τ0

main stairwell after the PAPA was activated. Situations like these are complex because of confusion due to conflicting cues.

4.4.2 Data set for training and testing the model

4.4.2.1 EMPIRICAL DATA SET (D1)

The empirical dataset D1 comprises the data collected from 17 participants in Group 1. For brevity, the data from only two participants are shown in Table 4.3. Each predicate takes typed variables, so corresponding ground atoms are shown in the second and third columns of the table. The data set D1 is split into two parts. Based on the methodology in Figure 4.1, the model in Table 4.2 was trained with different sizes of training/testing ratios, like 50/50, 60/40, 80/20. Eventually, an 80/20 split of D1 was found to produce good results. That is, 80% of the data in D1 was used for training the rules in Table 4.2, and 20% of the data was used here for testing the model.

4.4.2.2 EMPIRICAL DATASET (D2)

The empirical dataset D2 comprises the data collected from all 19 participants in Group 2. Again, based on the methodology in Figure 4.1, different samples sizes were tried for partitioning the dataset D2; the 80/20 ratio for training and testing samples was used here.

4.4.2.3 SETTING UP THE MODEL

We consider close world assumption for all predicates except `KETA`, `KETT`, and `KETPA`. The predicates `KETA`, `KETT,` and `KETPA` employ open world assumption because these predicates are designed to be present in the model as a container for the background knowledge. `KETA` is `true` when the agent has knowledge about which alarm is for which emergency situation type, i.e., the fact that the `GPA` alarm sounds for the `FIRE` type emergency, and the `PAPA` alarm is activated for `EVACUATE` type. `KETT` is used to mean which type of threat is observed in an emergency. For example, a fire confined to a small area, at most, could mean to move to the primary muster station. Three types of threats are considered in this study. The threat *smoke in the stairwell* (SMK_STAI) should be recognized as a FIRE type emergency. If an agent sees smoke coming out of the mess hall vent (SMK_VENT), or the agent enters into the mess hall and sees smoke (SMK_MSHA) there, it means the situation is of type EVACUATE because the primary muster station is compromised. If KETT is true, it means that the agent knows the relationships between a threat and possible type of emergency situation that could originate from this threat. Similarly, the KETPA predicate is true if the agent knows which words in the PA would lead to a particular emergency type. For example, the sentences, "a fire in the galley" or "move to primary muster station" mean that the emergency type is FIRE. On the other hand, the words, "primary escape route is blocked" or "a fire has escalated" mean that the situation is EVACUATE. This knowledge was given to the participants of Smith's experiment as part of the training curriculum. Therefore, during training of the model the truth values of KETA, KETT, and KETPA are taken as true to mean that the agents based on the proposed model have this background knowledge.

4.4.2.4 CALCULATING THE MODEL WEIGHTS

We use the software package Alchemy 2.0 (2012) for developing the proposed MLN model. The non-evidence predicates used for both D1 and D2 are R, HES and HSES. The model is trained separately for data sets D1 and D2 using a discriminative learning method so that weights can be assigned to the rules presented in Table 4.2. It was observed that some participants did not listen to an alarm even though it was audible. The use of Listens (L) as a predicate came up (see Table 4.2) with the empirical observations, where, with some participants the predicate takes a `false` value. On the other hand, if *Hears* were used instead of `Listens`, then there would not be any case with a `false` value for *Hears* because all the participants had hearing abilities in the normal range. Similar considerations were taken for other rules. Table 4.4 shows the weights. A portion of ground MN obtained by grounding the rules#2-5 is depicted in Figure 4.4, which shows how the nodes corresponding to each predicate are related.

4.5 Results and discussion

Querying the proposed MLN based model of agent SA is the same as querying a

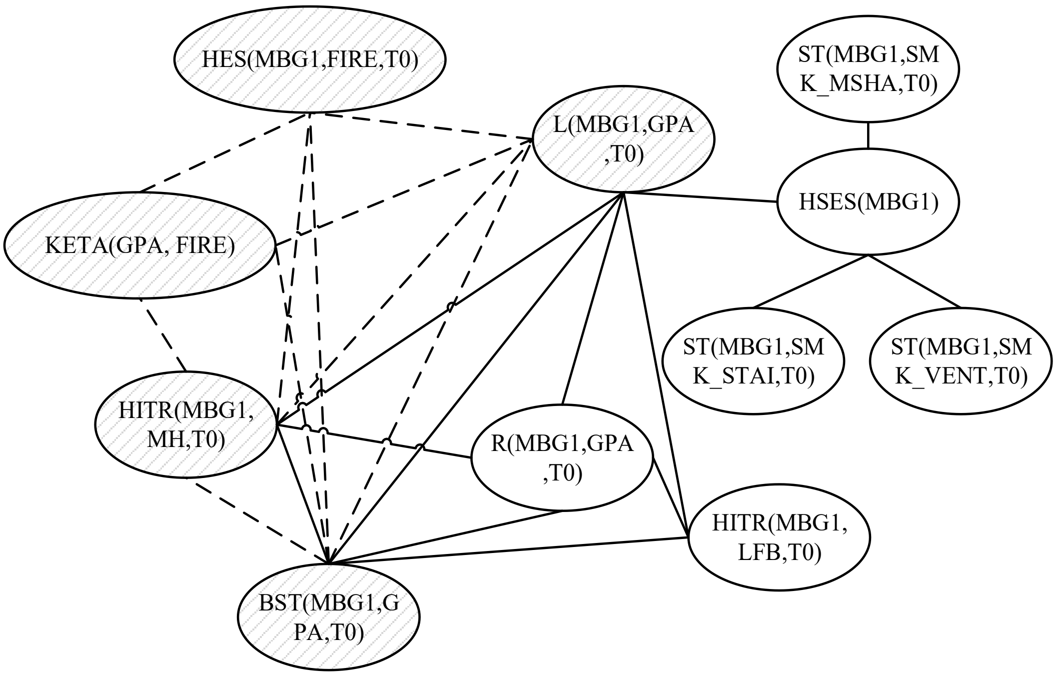


Figure 4.4. Portion of ground MN obtained using grounding of the predicates in rules 2-5.

Table 4.4. Weights assigned to rules using datasets D1 and D2. Only 12 out of a total of 59 ground rules obtained by different groundings of the rules in Table 4.2 are shown for brevity.

#	Rules	w_{D1}	w_{D2}
1	`¬L(ag,al,τ)⇒ ¬R(ag,al, τ).`	∞	∞
2	`L(ag,GPA, τ)^HITR(ag,MSH, τ)^BST(ag,+al, τ) ⇒R(ag,+al, τ)`	2.1	1.9
3	`L(ag,PAPA, τ)^HITR(ag,MSH, τ)^BST(ag,+al, τ) ⇒R(ag,+al, τ)`	0.8	0.2
4	`L(ag,PAPA, τ)^HITR(ag,LFB, τ)^BST(ag,+al, τ) ⇒R(ag,+al, τ)`	2.7	2.7
5	`L(ag,al, τ) ⇒ HSES(ag)`	1.3	1.4
6	`L(ag,al, τ)^ ¬R(ag,al, τ) ⇒ HSES(ag)`	0.3	0.4
7	`ST(ag,thrt, τ) ⇒ HSES(ag)`	0.9	1.0
8	`(HFO(ag,PA_GPA,τ)^FPA(ag,+p_a,τ)^KETPA(+p_a, FIRE))v(ST(ag,SMK_VENT,τ)^KETT(+thrt, +eTyp))v(L(ag,GPA,τ)^HITR(ag,MSH,τ)^ KETA(+al,+eTyp)^BST(ag,+al,τ)) ⇒HES(ag,+eTyp,τ)`	0.3	0.3
9	`(HFO(ag,PA_GPA,τ)^FPA(ag,+p_a,τ)^KETPA(+p_a, FIRE))v(ST(ag,SMK_VENT,τ)^KETT(+thrt, +eTyp))v(L(ag,GPA,τ)^HITR(ag,LFB,τ)^ KETA(+al,+eTyp)^BST(ag,+al,τ)) ⇒ HES(ag,+eTyp,τ)`	0.2	0.2
10	`HES(ag,FIRE,τ) ⇒ ¬HES(ag,EVACUATE, τ)`	1.4	1.5
11	`HES(ag,EVACUATE, τ) ⇒ ¬HES(ag,FIRE, τ)`	1.4	1.5
12	`HES(ag,FIRE,τ0)^HES(ag,EVACUATE,τ1)^Gt(τ1,τ0) ⇒ ¬HES(ag,FIRE,τ1)`	–1.4	–0.4

knowledgebase. We use the MC-SAT algorithm using the Alchemy inference engine for querying. Now if the model is used in an agent program as a part of situation assessment logic, the evidence would come via the available sensors. Given the evidence predicates, the agent can determine the chances that a query predicate is `true` in the present conditions. The most important things an agent seeks in an evolving emergency are the recognition of alarms and determination of the type of emergency it is in at a given time. For this reason, the query predicates are obtained by grounding the following predicates:

```
? R(agent ag, alarm al, time τ),
? HES(agent ag, emgSitType e, time τ),    (4.5)
and
? HSES(agent ag),
```

where, the predicate `R` is read as the `agent`, `ag`, recognizes an `alarm`, `al`, during the `time` interval τ. `HES` means that the `agent`, `ag`, has an emergency, `e`, of type `emgSitType`, during `time` τ, and the predicate `HSES` represents an `agent`, `ag`, who has got some sense of emergency. If in any case, the truth value of `HSES` is `true` and `HES` is `false`, it would mean that the agent is unable to determine the type of emergency despite that it has sensed the emergency situation. The predicates obtained after grounding the predicates listed in Table 4.2 other than the query predicates mentioned in (4.5) are used as part of the evidence predicates that need to be provided to the inference engine to obtain the results of the queries presented in (4.5).

Table 4.5 presents the probabilities estimated against the queries for the cases in the testing datasets. The test datasets were formed by taking 20% of the total samples from D1 and D2 respectively, as reported in Section 4.4.2.

With regards to the training and testing datasets for the model, the total duration each participant spends during a training or testing session has been divided into two intervals. The first is the interval `τ0` that starts from the beginning of a session until the time when the `GPA` alarm stops. The second interval is termed `τ1`, which is the interval that follows immediately after `τ0` ends, and it ends at the end of each session. `τ0` covers the period when there is `FIRE` type emergency, and `τ1` covers the duration when there is `EVACUATE` type emergency. This division of time is important to assess the importance of cues relevant to each emergency type. For example, if an agent observes smoke in the central stairwell, then this is an important cue for `FIRE` type emergency because in that case, the agent should move to the primary muster station, the mess hall. On the other hand, smoke in the central stairwell should not be considered during $\tau 1$, or when the `PAPA` alarm sounds, because `PAPA` alarm is a call to gather at the secondary, or alternative muster station, the `LIFEBOAT` station. Often in such cases, the primary muster station may have been compromised, or the routes that lead to the primary muster station may have been blocked.

Table 4.5 presents the results that are obtained for seven participants P1G1, P2G1, P3G1, P1G2, P2G2, P3G2, and P4G2. The names of these participants are kept hidden due to privacy. The information obtained by watching the replay videos and by observing the log files is divided into two columns with the view that those predicates that are used as part of the evidence in the inference algorithm are kept under the heading of *evidence* and those that are used to query the model are kept as empirical results. Both columns contain the empirical results obtained from Smith's experiment. The truth values of the empirical results are used for validating the model output that is described as the last column in Table 4.5.

4.5.1 Simulation results against the participant P1G1

Now consider the case when the participant, P1G1, was tested in AVERT. The evidence predicates suggest that immediately after hearing the alarm, P1G1 developed the intention to move to the mess hall, the primary muster station, which was correct, but the participant spent more time than needed and so reached the mess hall when $\tau 0$ had already expired. On the other hand, this also means that P1G1 recognized the `GPA` alarm, `R(P1G1, GPA, τ0)`, and developed awareness about

Table 4.5. Query results. The symbol '¬' is the logical *not* operator. A predicate followed by a symbol '¬' has a truth value of false, otherwise true. The column for empirical results contains the results obtained from participants. Corresponding to each empirical result is a probability the model generated for that predicate. For example, `R(P1G1, GPA, τ0)` meaning that the participant P1G1 has recognized the `GPA` alarm during time interval `τ0`. The probability that this predicate `R(P1G1, GPA, τ0)` is true is 0.91. Notice that the constant `EVAC` means `EVACUATE` in the following predicates.

#		Evidence	Empirical result	Model output probability
1.	1	L(P1G1, GPA, τ0)	R(P1G1, GPA, τ0)	0.91
	2	HITR(P1G1, MSH, τ0)	HES(P1G1, FIRE, τ0)	0.92
	3	BST(P1G1, GPA, τ0)	HES(P1G1, FIRE, τ1)	0.74
	4	HITR(P1G1, MSH, τ1)	¬HES(P1G1, EVAC,τ1)	0.16
	5	ST(P1G1, SMK_MSHA, τ1)	¬HES(P1G1, EVAC,τ0)	0.12
	6	ST(P1G1, SMK_STAI, τ1)	HSES(P1G1)	0.99
	7	ST(P1G1, SMK_VENT, τ1)	¬R(P1G1, PAPA,τ1)	0.0
	8	HFO(P1G1, PA_GPA, τ0)		
	9	FPA(P1G1, PA_GPA, τ0)		
	10	¬L(P1G1, PAPA, τ1)		
	11	¬BST(P1G1, PAPA, τ1)		
	12	¬HFO(P1G1, PA_PAPA, τ1)		
	13	¬FPA(P1G1, PA_PAPA, τ1)		
	14	HITR(P1G1, LFB, τ1)		
2.	1	L(P2G1, GPA, τ0)	R(P2G1, GPA, τ0)	0.87
	2	HITR(P2G1, MSH, τ0)	HES(P2G1, FIRE,τ0)	0.94
	3	BST(P2G1, GPA, τ0)	¬HES(P2G1, FIRE, τ1)	0.29
	4	ST(P2G1, SMK_VENT, τ0)	R(P2G1, PAPA, τ1)	0.92
	5	HFO(P2G1, PA_GPA, τ0)	HES(P2G1, EVAC, τ1)	0.98
	6	FPA(P2G1, PA_GPA, τ0)	¬HES(P2G1,EVAC,τ0)	0.07
	7	L(P2G1, PAPA, τ1)	HSES(P2G1)	0.98
	8	HFO(P2G1, PA_PAPA, τ1)		
	9	FPA(P2G1, PA_PAPA, τ1)		
	10	BST(P2G1, PAPA, τ1)		
	11	HITR(P2G1, LFB, τ1)		
3.	1	L(P3G1, GPA, τ0)	¬R(P3G1, GPA, τ0)	0.49
	2	¬HITR(P3G1,MSH, τ0)	¬HES(P3G1, FIRE, τ0)	0.44
	3	¬BST(P3G1, GPA, τ0)	¬HES(P3G1, FIRE, τ1)	0.15
	4	ST(P3G1, SMK_VENT, τ0)	R(P3G1, PAPA, τ1)	0.93
	5	¬HFO(P3G1,PA_GPA, τ0)	HES(P3G1, EVAC,τ1)	0.99
	6	¬FPA(P3G1, PA_GPA, τ0)	¬HES(P3G1,EVAC,τ0)	0.24
	7	L(P3G1, PAPA, τ1)	HSES(P3G1)	0.90
	8	HFO(P3G1, PA_PAPA, τ1)		

#		Evidence	Empirical result	Model output probability
	9	FPA(P3G1, PA_PAPA, τ1)		
	10	BST(P3G1, PAPA, τ1)		
	11	HITR(P3G1, LFB, τ1)		
4.	1	L(P1G2,GPA, τ0)	R(P1G2, GPA, τ0)	0.88
	2	HITR(P1G2,MSH, τ0)	HES(P1G2, FIRE, τ0)	0.88
	3	BST(P1G2,GPA, τ0)	¬HES(P1G2,FIRE, τ1)	0.08
	4	HITR(P1G2,MSH, τ1)	R(P1G2, PAPA, τ1)	0.94
	5	ST(P1G2, SMK_VENT, τ0)	HES(P1G2,EVAC, τ1)	0.98
	6	ST(P1G2, SMK_VENT, τ1)	HSES(P1G2)	0.99
	7	HFO(P1G2, PA_GPA, τ0)	HES(P1G2,EVAC,τ0)	0.15
	8	FPA(P1G2, PA_GPA, τ0)		
	9	L(P1G2, PAPA, τ1)		
	10	HFO(P1G2, PA_PAPA, τ1)		
	11	FPA(P1G2, PA_PAPA, τ1)		
	12	HITR(P1G2, LFB, τ1)		
	13	BST(P1G2,PAPA, τ1)		
5.	1	L(P2G2,GPA, τ0)	R(P2G2,GPA, τ0)	0.87
	2	HITR(P2G2,MSH, τ0)	¬R(P2G2, GPA, τ1)	0.0
	3	BST(P2G2, GPA, τ0)	¬R(P2G2, PAPA, τ0)	0.0
	4	HITR(P2G2,MSH, τ1)	¬R(P2G2,PAPA,τ1)	0.49
	5	ST(P2G2,SMK_MSHA, τ1)	HES(P2G2,FIRE,τ0)	0.93
	6	ST(P2G2,SMK_VENT, τ1)	HES(P2G2,FIRE,τ1)	0.52
	7	ST(P2G2,SMK_STAI, τ1)	¬HES(P2G2,EVAC,τ0)	0.06
	8	HFO(P2G2,PA_GPA, τ0)	¬HES(P2G2,EVAC,τ1)	0.47
	9	FPA(P2G2,PA_GPA, τ0)	HSES(P2G2)	0.99
	10	L(P2G2,PAPA, τ1)		
	11	¬BST(P2G2,PAPA, τ1)		
	12	¬HFO(P2G2,PA_PAPA, τ1)		
	13	¬FPA(P2G2,PA_PAPA, τ1)		
	14	HITR(P2G2,LFB, τ1)		
6.	1	L(P3G2,GPA, τ0)	¬R(P3G2,GPA,τ0)	0.5
	2	¬HITR(P3G2,MSH,τ0)	¬R(P3G2,GPA, τ1)	0.0
	3	¬BST(P3G2,GPA, τ0)	¬R(P3G2,PAPA, τ0)	0.0
	4	¬ST(P3G2,SMK_MSHA,τ0)	R(P3G2,PAPA, τ1)	0.91
	5	ST(P3G2,SMK_VENT,τ1)	HES(P3G2,FIRE, τ0)	0.99
	6	ST(P3G2,SMK_STAI,τ1)	¬HES(P3G2,FIRE, τ1)	0.13
	7	HFO(P3G2,PA_GPA,τ0)	HES(P3G2,EVAC, τ1)	0.98
	8	FPA(P3G2,PA_GPA,τ0)	HSES(P3G2)	0.99
	9	L(P3G2,PAPA, τ1)	¬HES(P3G2,EVAC,τ0)	0.05
	10	HFO(P3G2,PA_PAPA, τ1)		
	11	BST(P3G2,PAPA, τ1)		
	12	FPA(P3G2,PA_PAPA, τ1)		
	13	HITR(P3G2,LFB, τ1)		

#		Evidence	Empirical result	Model output probability
7.	1	L(P4G2,GPA, τ0)	R(P4G2,GPA, τ0)	0.87
	2	HITR(P4G2,MSH, τ0)	HES(P4G2,FIRE, τ0)	0.93
	3	BST(P4G2,GPA, τ0)	HES(P4G2,FIRE, τ1)	0.56
	4	HITR(P4G2,MSH, τ1)	¬R(P4G2,PAPA, τ1)	0.49
	5	ST(P4G2,SMK_MSHA,τ1)	¬HES(P4G2,EVAC,τ1)	0.47
	6	ST(P4G2,SMK_VENT,τ1)	HSES(P4G2)	0.99
	7	ST(P4G2,SMK_STAI,τ1)		
	8	HFO(P4G2,PA_GPA, τ0)		
	9	FPA(P4G2,PA_GPA, τ0)		
	10	L(P4G2,PAPA, τ1)		
	11	¬BST(P4G2,PAPA, τ1)		
	12	HFO(P4G2,PA_PAPA,τ1)		
	13	¬FPA(P4G2,PA_PAPA,τ1)		
	14	HITR(P4G2,LFB,τ1)		

the FIRE situation, HES(P1G1, FIRE, τ0), during the initial time interval τ0. But as a slow mover, P1G1 observed the smoke in the stairwell, mess hall, and the smoke coming through the mess hall ventilation during τ1. P1G1 also did not pay attention to the PAPA alarm, which is the reason for ¬L(P1G1, PAPA,τ1), which was activated when P1G1 was still in the main stairwell. P1G1 took about 20 seconds more in τ1, ignoring the fact that the PAPA alarm implies a re-route towards the lifeboat station through the secondary escape route. So, unnoticed from the PAPA alarm and the relevant PA, P1G1 entered the mess hall and saw thick smoke. Studies (Posner, Nissen, & Klein, 1976; Sinnett, Spence, & Soto-Faraco, 2007) suggest that humans show dominance on visual information than on other types of sensory cues such as auditory information. Observing smoke drew the P1G1's attention on smoke, and he instantly realized a need to move out of the mess hall, which was done by re-routing to the lifeboat. But this realization of the situation comes only when P1G1 saw smoke, and it was not due to the PAPA alarm or the relevant PA. In a real situation, entering an area filled with smoke due to fire or any other toxic element could be lethal. Also, observing a fire or smoke is a natural cue that would develop awareness about a fire situation. It is, nevertheless, hard to develop awareness about an evacuation situation by watching a fire or smoke unless the relevant alarms and/or platform announcements are heard and recognized. This is the reason why P1G1, although mustered at the lifeboat station, is considered to be poor in responding to the evacuation situation, and that is why we have ¬R(P1G1, PAPA,τ1) and ¬HES(P1G1, EVACUATE,τ1) in the empirical results for P1G1. Similarly, P1G1 spent a fraction of the interval τ1 maintaining the impression of a fire situation, although the fire situation had already been escalated to an evacuation situation, which is why we have a predicate HES(P1G1, FIRE, τ1) in the empirical results. The model output is probabilities obtained against the query predicates, as shown in the last column of Table 4.5.

Ideally, a high probability is a good fit for a queried predicate when the corresponding empirical result has a truth value of `true`. Similarly, a low output probability should serve a good fit for the queries predicate when its empirical truth value is `false`. This is very much evident for P1G1. Given the listed evidence for P1G1, the probability that an agent would recognize a GPA is 0.91, and the probability the same agent would get immediate fire emergency awareness is 0.92. However, there are fewer chances (only 16%) that the agent would respond to the escalating situation from FIRE to EVACUATE because the likelihood of recognition of the PAPA alarm is zero, as the agent does not listen to or has no focus on the sounding alarm. In any case, if we change the evidence truth value for the predicate 1.10 in Table 4.5 from `false` to `true`, the corresponding probability of recognizing `PAPA` during `τ1` would increase from 0.0 to 0.48. The reason for getting a zero probability is due to the hard constraint (rule#1) listed in Table 4.4. Similarly, if P1G1 realized the presence of smoke in the stairwell during `τ0` rather than `τ1`, for example, if P1G1 had moved fast, then the chances for having a `FIRE` situation during `τ1` would have been lowered from 0.74 to 0.46, and the chances for getting awareness about the `EVACUATE` situation would be increased from 16% to 23% during `τ1`. This is because the `SMK_STAI`, i.e., seeing smoke in the stairs, is a positive cue for a fire situation, but when one observes it in the presence of a cue that is for an evacuation situation, for example, a `PAPA` alarm, the two conflicting cues would cause confusion, and the agent needs to decide which cue should be considered. P1G1 preferred `SMK_STAI` during `τ1` over the `PAPA` alarm and so entered the mess hall, although this decision was wrong as it wasted egress time and exposed the participant to a hazard.

4.5.2 Simulation results against the participant P2G1

The case of participant P2G1 shows a slight deviation between the model output and the empirical results at only one place (see empirical result # 2.3 and corresponding model output probability in Table 4.5). The model output probability of keeping the impression of a fire situation, though the situation had turned into an evacuation situation, is a bit high (0.29) compared to the empirical result where the truth value of the involved predicate, `HES(P2G1, FIRE, τ1)`, was `false`. The rest of the model output probabilities, estimated for modeling P2G1's behavior, are reasonable.

4.5.3 Simulation results against the participants P3G1 and P1G2

The only thing participant P3G1 took into consideration during `τ0` was the smoke coming out from the mess hall ventilation. P3G1 did not recognize the `GPA` alarm nor heed the PA for the `FIRE` emergency. P3G1 never had any intention to move to the mess hall. The model output for recognizing the `GPA` alarm (0.49) during `τ0` is reasonable because the time when the `GPA` starts sounding is the time when the participant is in the cabin, and there are no other available cues except the alarm sound and the relevant PA. The model output probabilities are in good agreement with the empirical results except for a slightly larger value of 0.44 for the probability of having awareness about `FIRE` emergency during `τ0`, whereas P3G1 remained unaware about the fire emergency, and from the beginning of the scenario P3G1 had decided to muster at the `LIFEBOAT` station. The results obtained against the evidence for the participant P1G2 are all in good agreement with the empirical values.

4.5.4 Simulation results against the participant P2G2

By giving the evidence of P2G2, the model recognizes the fire alarm during `τ0` with 0.87 probability. P2G2 did not recognize the `PAPA` during the experiment, and the model output is 0.49 for the predicate `R(P2G2, PAPA, τ1)`. The reason for having a probability near 0.5 is that when the interval shifted from `τ0` to `τ1`, there are only two cues suggesting that the situation has escalated from `FIRE` to `EVACUATE` (smoke from the vents and the smoke in the mess hall) and the smoke in the stairwell is a cue for moving to the mess hall. This is a conflicting situation. Moreover, as P2G2 moved into the mess hall while the `PAPA` alarm was still on along with the relevant PA, the predicate `BST(P2G2, PAPA, τ1)` takes a `false` value in the evidence that reduced the probability of recognizing `PAPA` during `τ1` from 0.94 (if `BST(P2G2, PAPA, τ1)` is `true`) to 0.49 when the predicate BST is `false`, as in the case of P2G2. Similar reasoning is `true` for recognizing the `FIRE` and `EVACUATE` situations during `τ1`. If we set `BST(P2G2, PAPA, τ1) true` in the evidence dataset for P2G2, then the new values for probabilities for having awareness about `FIRE` and `EVACUATE` situations during `τ0` and `τ1` come out to be 0.94 for a `FIRE` at `τ0` and 0.96 for `EVACUATE` at `τ1`. This shows the importance of recognizing the alarm before seeing any real threat.

4.5.5 Simulation results against the participants P3G2 and P4G2

The participant P3G2 did not recognize the `GPA` alarm, and the model probability against the query predicate is 0.5 for similar reasons we observed in the case of P3G1. The rest of the results for P3G2, as reported in Table 4.5, support the empirical results for P3G2. Similar reasons are there for the results obtained against the query predicates for P4G2.

4.6 Conclusions

A MLN-based model of SA for agents in a VE is proposed in this work. The methodology used here involves assessing the environmental and cognitive factors, such as alarms, fire/smoke, intention, and focus of attention, for potential impact on awareness of emergencies. The proposed model has been used to represent two case studies that involve fire and evacuation situations on an offshore petroleum platform. The case studies were carried out in a VE with real people. Data obtained from the case studies are used to validate the model output. Empirical and simulated results agree in asserting the importance of alarm recognition and focus of attention for awareness about the emergency situations involving smoke and fire.

Endsley's SA model describes how people get awareness about a situation, but it does not provide how such a model can be used for software agents (Kokar et al., 2009). The present work shows a potential approach to modeling SA for software agents. Agents based on this model can be used in several application areas. For example, one can exploit such agents so that different situations can be considered as different experiences, and hence a repertoire of situations can be made as a basis for decision-making regarding choosing actions in a given a situation. Virtual training environments are good examples of using such agents for cohort training where agents, based on the proposed methodology, can exhibit different behaviors in different situations for training purposes. Due to the inherent stochasticity of the proposed approach, the model is dynamic, and it has an advantage over other models, such as ontology-based SA models (Kokar et al., 2009; Kokar, Shin, Ulicny, & Moskal, 2014; Malizia, Onorati, Diaz, Aedo, & Astorga-Paliza, 2010), and case-based SA models (Nwiabu, Allison, Holt, Lowit, & Oyeneyin, 2012), in that it can recognize a situation even if some of the FOL rules violate.

This work has the potential to be used in Naturalistic Decision-Making (NDM) environments where situations are central entities to decision making (Gore, Flin, Stanton, & Wong, 2015). Another application is in intelligent tutoring where the model can be used to make student models in a VE for training people for different tasks of SA. Different kinds of agents can be developed — even without using training-testing samples, by manually selecting weights (Jain, 2011) — for tutoring different behaviors. For example, an agent that has *poor* capabilities of recognizing alarms should use a real positive number near zero as a weight for rule#2. Similarly, an agent that acts as an expert should have high values of weights in the rules, and the evidence database should contain as much of the needed information as possible so that the agent acts as an expert in retrieving cues from the environment.

Declarations

Author's contributions

S. N. Danial performed the design and implementation of the MLN model and extracted the empirical data from J. Smith's (2015) experiment for validation of the model. J. Smith performed the experiment presented in (J. Smith, 2015) and verified the data extracted from the experiment. F. Khan supervised the MLN model development. B. Veitch supervised the entire study and performed the editorial process. All authors read and approved the final draft.

Acknowledgments

The authors would like to thank Mr. Saqib Munawar for sharing his experience of emergency egress on various offshore rigs.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The training and testing samples used for the validation of the MLN-based SA model are publicly available. Full datasets are provided here as supplementary files. The replay videos used to create the training and testing samples have restricted access because AVERT simulator is not available for public use.

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.

References

Akman, V., & Surav, M. (1996). Steps toward formalizing Context. *AI Magzine*, *17*(3). https://doi.org/https://doi.org/10.1609/aimag.v17i3.1231

Alchemy. (2012). *Alchemy: A software for statistical relational learning and probabilistic logic inference based on Markov logic representation*. Washington DC.

Barwise, J. (1981). Scenes and Other Situations. *The Journal of Philosophy*, *78*(7), 369. https://doi.org/10.2307/2026481

Barwise, J., & Perry, J. (1980). *The Situation Underground*. California: Stanford Cognitive Science Group 1980, Section D.

Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge, MA: MIT Press.

Bosse, T., & Mogles, N. (2014). Spread of situation awareness in a group: Population-based vs. agent-based modelling. *Proceedings - 2014 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2014*, *3*, 1117–1124. https://doi.org/10.1109/WI-IAT.2014.169

Bratman, M. (1987). *Intention, plans, and practical reason*. Cambridge, MA: Harward University Press.

Chowdhury, S. (2016). *Optimization and Business Improvement: Studies in Upstream Oil and Gas Industry*. New Jersey: Wiley.

Cornfield, J., Haenszel, W., Hammond, E. C., Lilienfeld, A. M., Shimkin, M. B., & Wynder, E. L. (2009). Smoking and lung cancer: Recent evidence and a discussion of some questions. *International Journal of Epidemiology*, *38*(5), 1175–1191. https://doi.org/10.1093/ije/dyp289

Cullen, L. W. D. (1993). The public inquiry into the Piper Alpha disaster. *Drilling Contractor (United States)*, *49:4*.

Danial, S. N., Khan, F., & Veitch, B. (2018). A Generalized Stochastic Petri Net model of route learning for emergency egress situations. *Engineering Applications of*

Artificial Intelligence, *72*, 170–182. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0952197618300733

Danial, S. N., Smith, J., Khan, F., & Veitch, B. (2019). Human-Like Sequential Learning of Escape Routes for Virtual Reality Agents. *Fire Technology*, *55*(3), 1057–1083. https://doi.org/10.1007/s10694-019-00819-7

Devlin, K. J. (1991a). *Logic and Information*. Cambridge: Cambridge University Press.

Devlin, K. J. (1991b). Situations as Mathematical Abstractions. *Situation Theory and Its Applications Vol. 1*, 25–39.

Domingos, P., & Lowd, D. (2009). Markov Logic: An interface layer for Artificial Intelligence. In T. Brachman, R. J.;Dietterich (Ed.), *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Seatle: Morgan & Claypool Publishers.

Domingos, P., & Richardson, M. (2007). Markov Logic: A Unifying Framework for Statistical Relational Learning. In B. Getoor, L.;Taskar (Ed.), *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Endsley, M. (1988). Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *32*. https://doi.org/10.1177/154193128803200221

ExxonMobil. (2010). *OIMS: System 10-2 Emergency Preparedness and Response.* Retrieved from https://www.cnsopb.ns.ca/sites/default/files/inline/12450_so41877.1_spill_response_soei_0.pdf

Gayathri, K. S., Easwarakumar, K. S., & Elias, S. (2017). Probabilistic ontology based activity recognition in smart homes using Markov Logic Network. *Knowledge-Based Systems*, *121*, 173–184. https://doi.org/10.1016/j.knosys.2017.01.025

Gayathri, K. S., Elias, S., & Shivashankar, S. (2014). *An Ontology and Pattern Clustering Approach for Activity Recognition in Smart Environments*. https://doi.org/10.1007/978-81-322-1771-8_72

Gore, J., Flin, R., Stanton, N., & Wong, B. L. W. (2015). Applications for naturalistic decision-making. *Journal of Occupational and Organizational Psychology*, *88*(2), 223–230. https://doi.org/10.1111/joop.12121

Grimmett, G. (2010). *Probability on Graphs: Random Processes on Graphs and Lattices*. Cambridge: Cambridge University Press.

Halpern, J. Y. (2003). *Reasoning about uncertainty*. Cambridge, MA: MIT Press.

Hu, Y., Li, R., & Zhang, Y. (2018). Predicting pilot behavior during midair encounters using recognition primed decision model. *Information Sciences*, *422*, 377–395. https://doi.org/10.1016/j.ins.2017.09.035

Isham, V. (1981). An Introduction to Spatial Point Processes and Markov Random Fields. *International Statistical Review / Revue Internationale de Statistique*, *49*(1), 21. https://doi.org/10.2307/1403035

Jain, D. (2011). Knowledge Engineering with Markov Logic Networks: A review. In G. Beierle, C.; Kern-Isberner (Ed.), *Proceedings of Evolving Knowledge in Theory and Applications. 3rd Workshop on Dynamics of Knowledge and Belief (DKB-2011) at the 34th Annual German Conference on Artificial Intelligence, KI-2011, vol. 361.* (pp. 16–30). Berlin, Germany: Fakultät für Mathematik und Informatik, FernUniversität in Hagen.

Johnson, A. W., Duda, K. R., Sheridan, T. B., & Oman, C. M. (2017). A Closed-Loop Model of Operator Visual Attention, Situation Awareness, and Performance Across Automation Mode Transitions. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *59*(2), 229–241. https://doi.org/10.1177/0018720816665759

Khan, B., Khan, F., Veitch, B., & Yang, M. (2018). An operational risk analysis tool to analyze marine transportation in Arctic waters. *Reliability Engineering & System Safety*, *169*, 485–502. https://doi.org/10.1016/j.ress.2017.09.014

Kindermann, R., & Snell, J. L. (1980). Markov random fields and their applications. In *Science* (Vol. 1). https://doi.org/10.1109/TVCG.2009.208

Kingston, C., Nurse, J. R. C., Agrafiotis, I., & Milich, A. B. (2018). Using semantic clustering to support situation awareness on Twitter: the case of world views. *Human-Centric Computing and Information Sciences*, *8*(1), 22. https://doi.org/10.1186/s13673-018-0145-6

Klein, G. A. (1998). *Sources of Power*. Cambridge, MA: MIT Press.

Kokar, M. M., Matheus, C. J., & Baclawski, K. (2009). Ontology-based situation awareness. *Information Fusion*, *10*(1), 83–98. https://doi.org/10.1016/j.inffus.2007.01.004

Kokar, M. M., Shin, S., Ulicny, B., & Moskal, J. (2014). Inferring relations and individuals relevant to a situation: An example. *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 18–194. https://doi.org/10.1109/CogSIMA.2014.6816561

Liu, F., Deng, D., & Li, P. (2017). Dynamic Context-Aware Event Recognition Based on Markov Logic Networks. *Sensors*, *17*(3), 491. https://doi.org/10.3390/s17030491

Llinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E., & White, F. (2004). Revisiting the JDL Data Fusion Model II (2004). In P. Svensson & J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004), June 28-July 1, 2004.* Stockholm, Sweden.

Luck, M., & Aylett, R. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, *14*(1), 3–32. https://doi.org/10.1080/088395100117142

Malizia, A., Onorati, T., Diaz, P., Aedo, I., & Astorga-Paliza, F. (2010). SEMA4A: An ontology for emergency notification systems accessibility. *Expert Systems with Applications*, *37*(4), 3380–3391. https://doi.org/10.1016/j.eswa.2009.10.010

Musharraf, M., Smith, J., Khan, F., Veitch, B., & MacKinnon, S. (2018). Incorporating individual differences in human reliability analysis: An extension to the virtual experimental technique. *Safety Science*, *107*, 216–223. https://doi.org/10.1016/j.ssci.2017.07.010

Naderpour, M., Lu, J., & Zhang, G. (2014). An intelligent situation awareness support system for safety-critical environments. *Decision Support Systems*, *59*, 325–340. https://doi.org/10.1016/j.dss.2014.01.004

Nakanishi, H., Shimizu, S., & Isbister, K. (2005). Sensitizing social agents for virtual training. *Applied Artificial Intelligence*, *19*(3–4), 341–361. https://doi.org/10.1080/08839510590910192

Nasar, Z., & Jaffry, S. W. (2018). Trust-Based Situation Awareness: Comparative Analysis of Agent-Based and Population-Based Modeling. *Complexity*, *2018*, 1–17. https://doi.org/10.1155/2018/9540726

Norazahar, N., Smith, J., Khan, F., & Veitch, B. (2018). The use of a virtual environment in managing risks associated with human responses in emergency situations on offshore installations. *Ocean Engineering*, *147*, 621–628. https://doi.org/10.1016/j.oceaneng.2017.09.044

Nowroozi, A., Shiri, M. E., Aslanian, A., & Lucas, C. (2012). A general computational recognition primed decision model with multi-agent rescue simulation benchmark. *Information Sciences*, *187*, 52–71. https://doi.org/10.1016/j.ins.2011.09.039

Nwiabu, N., Allison, I., Holt, P., Lowit, P., & Oyeneyin, B. (2012). Case-based situation awareness. *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support. 6-8 March 2012*, 22–29. https://doi.org/10.1109/CogSIMA.2012.6188388

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inferences*. San Mateo, CA: Morgan Kaufmann.

Poon, H., & Domingos, P. (2006). Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, 458–463. Retrieved from http://dl.acm.org/citation.cfm?id=1597538.1597612

Posner, M. I., Nissen, M. J., & Klein, R. M. (1976). Visual dominance: An information-processing account of its origins and significance. *Psychological Review*, *83*(2), 157–171. https://doi.org/10.1037/0033-295X.83.2.157

Preston, C. J. (1974). *Gibbs States on Countable Sets*. Cambridge: Cambridge University Press.

Proulx, G. (2007). Response to fire alarms. *Fire Protection Engineering*, *33*, 8–14. Retrieved from http://www.cfaa.ca/Files/flash/CODES/FIRE ALARM ARTICLES FOR THE AHJ/Fire Alarm Response.pdf

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

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Récopé, M., Fache, H., Beaujouan, J., Coutarel, F., & Rix-Lièvre, G. (2019). A study of the individual activity of professional volleyball players: Situation assessment and sensemaking under time pressure. *Applied Ergonomics*, *80*, 226–237. https://doi.org/10.1016/j.apergo.2018.07.003

Singla, P., & Domingos, P. (2005). Discriminative Training of Markov Logic Networks. *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, 868–873. Retrieved from http://dl.acm.org/citation.cfm?id=1619410.1619472

Sinnett, S., Spence, C., & Soto-Faraco, S. (2007). Visual dominance and attention: The Colavita effect revisited. *Perception & Psychophysics*, *69*(5), 673–686. https://doi.org/10.3758/BF03193770

Smith, J. (2015). *The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios [Master thesis]*. Memorial University of Newfoundland, St. John’s, NL, Canada. Retrieved from: https://research.library.mun.ca/8401/1/thesis.pdf

Smoking and Health: Joint Report of the Study Group on Smoking and Health. (1957). *Science*, *125*(3258), 1129–1133. https://doi.org/10.1126/science.125.3258.1129

Sneddon, A., Mearns, K., & Flin, R. (2013). Stress, fatigue, situation awareness and safety in offshore drilling crews. *Safety Science*, *56*, 80–88. https://doi.org/10.1016/j.ssci.2012.05.027

Snidaro, L., Visentini, I., & Bryan, K. (2015). Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks. *Information Fusion*, *21*, 159–172. https://doi.org/10.1016/j.inffus.2013.03.004

Snidaro, L., Visentini, I., Bryan, K., & Foresti, G. L. (2012). Markov Logic Networks for context integration and situation assessment in maritime domain. *2012 15th International Conference on Information Fusion*, 1534–1539.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks/Cole Thomson Learning.

Spouge, J. (1999). *A Guide to Quantitative Risk Assessment for Offshore Installations*. Aberdeen, UK: CMPT Publication.

Szczerbak, M., Bouabdallah, A., Toutain, F., & Bonnin, J.-M. (2013). A Model to Compare and Manipulate Situations Represented as Semantically Labeled Graphs. In H. D. Pfeiffer, D. I. Ignatov, J. Poelmans, & N. Gadiraju (Eds.), *Conceptual Structures for STEM Research and Education* (pp. 44–57). Berlin, Heidelberg: Springer Berlin Heidelberg.

Thilakarathne, D. J. (2015). Modelling of situation awareness with perception, attention, and prior and retrospective awareness. *Biologically Inspired Cognitive Architectures*, *12*, 77–104. https://doi.org/10.1016/j.bica.2015.04.010

Tong, D., & Canter, D. (1985). The decision to evacuate: a study of the motivations which contribute to evacuation in the event of fire. *Fire Safety Journal*, *9*(3), 257–265. https://doi.org/10.1016/0379-7112(85)90036-0

Tutolo, D. (1979). Attention: Necessary Aspect of Listening. *Language Arts*, *56*(1), 34–37. Retrieved from http://www.jstor.org/stable/41404756

Tversky, A., & Kahneman, D. (1974). Judgment under Uncertainty: Heuristics and Biases. *Science*, *185*(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

Wankhede, A. (2017, September). Different Types of Alarms on Ships. *Marine Insight*. Retrieved from https://www.marineinsight.com/marine-safety/different-types-of-alarms-on-ship/

Winerman, L. (2004). Fighting fire with psychology. *Monitor on Pscyhology*, *35*(8), 28. Retrieved from https://www.apa.org/monitor/sep04/fighting

Xu, G., Cao, Y., Ren, Y., Li, X., & Feng, Z. (2017). Network Security Situation Awareness Based on Semantic Ontology and User-Defined Rules for Internet of Things. *IEEE Access*, *5*, 21046–21056. https://doi.org/10.1109/ACCESS.2017.2734681

Yang, C., Wang, D., Zeng, Y., Yue, Y., & Siritanawan, P. (2019). Knowledge-based multimodal information fusion for role recognition and situation assessment by using mobile robot. *Information Fusion*, *50*, 126–138. https://doi.org/10.1016/j.inffus.2018.10.007

Chapter 5

On the realization of the Recognition-Primed Decision model for artificial agents[§§§]

Co-authorship statement. A version of this chapter is in peer-review in the journal *Human-centric Computing and Information Science* published *by Springer-Verlag*. The author, Syed Nasir Danial, has developed and implement the model and extracted the empirical data using re-play video files for validation of the model. The co-author Jennifer Smith performed the experiment and verified the data extracted from the experiment. Co-authors Dr. Faisal Khan and Dr. Brian Veitch supervised the study. All authors read and approved the final draft.

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

[§§§] Danial, S. N., Smith, J., Khan, F., & Veitch, B. (2019). On the realization of the recognition-primed decision model for artificial agents. *Human-centric Computing and Information Sciences*. (Accepted for publication).

proposed here to deal with all such cases. The third component of RPDM is the proposal that human beings use intuition and imagination (mental simulation) 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.

5.1 Introduction

NDM is a relatively new approach to decision-making that relies on SA (Endsley, 1988) 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 G. Klein's (1998) *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. (Zsambok, 1997). 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 (Chase & Simon, 1973; de Groot, 1965). 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 (Means, Salas, Crandall, & Jacobs, 1993). de Groot (1965) 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 (1965) 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 seconds or so (Chase & Simon, 1973, p. 217). 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 (Press, 2018). 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 Figure 5.1) explains how human decision-makers plan, in the event of an emergency, to mitigate the aftereffects or to

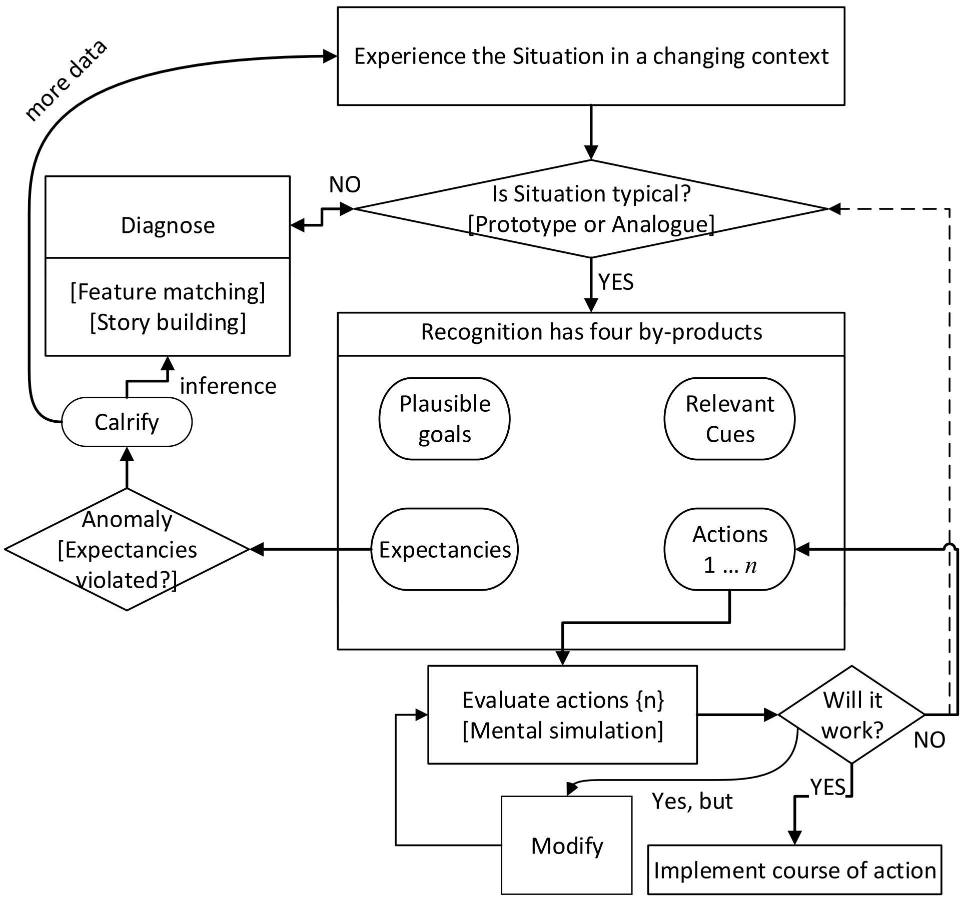


Figure 5.1. Integrated G. Klein's RPD model. Source (G. Klein, 1998, p. 27).

save life and property (Zsambok & Klein, 1997). The model argues that people are naturally inclined to making a plan based on their experiences (G. Klein, 1998) and intuition or intuitive knowledge (G. Klein, 2004), 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. (2012) proposed a model of RPDM 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 RPDM, and the authors claim that different sections of their work describe different constructs of RPDM. 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 RPDM. 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 world[13]? 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

[13] 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!

concept in RPDM 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 RPDM. Norling (2004), and Norling, Sonenberg, and Rönnquist (2000) proposed a BDI based agent model by integrating it with RPDM 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. (2007) proposed an RPDM based model that can be used to analyze drug effects. Based on the experience of how a military commander contributes to decision-making during warfare operations, Sokolowski (2003), 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 (Canellas & Feigh, 2016; Hassard, 2009; Hutton, Warwick, Stanard, McDermott, & McIlwaine, 2001; Norling et al., 2000; Nowroozi et al., 2012; Patterson, Fournier, Pierce, Winterbottom, & Tripp, 2009; Resnick, 2001). 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 an MLN that needs training to acquire experience. An informational theory based account on modeling SA is given in (Devlin, 1991a), which is based on Barwise and Perry's (1983) situation semantics. A common approach to quantitative modeling of SA involves BNs (Hu et al., 2018; Naderpour et al., 2014). 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 (2007) proposed Markov logic, whereby a Markov network, which supports cycles, is developed based on information represented in the form of FOL rules. The mental simulation component is considered here as a *cause-and-effect* phenomenon (G. Klein, 1998 pp. 89-90), and is proposed to be represented in the form of Bayesian formalism (Pearl, 1988). 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 (2000) 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 (Hu et al., 2018) 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. (2018) use Bayesian network for SA unlike previous attempts, e.g. (Nowroozi et al., 2012) where the authors use a direct count on the number of matched features, e.g., by using a similarity criterion, see (Fan et al., 2010), as a sufficient representation of SA. The pilot models are important to study midair encounter scenarios. The model proposed in (Hu et al., 2018) 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

Table 5.1. A comparison / mapping of the major concepts of Klein's RPD model with the components in the proposed realization, and in some previous works. Model A is proposed in (Hu et al., 2018), Model B is proposed in (Nowroozi et al., 2012), and Model C is proposed in (Mueller, 2009).

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 plans	Stored as part of 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

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 (G. Klein, 1998 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 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 5.1 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 (Endsley, 1988), 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.

Section 5.2 describes some background concepts, which will help develop an understanding of this study. Section 5.3 explains the methodology proposed here, which includes the development of ELDS and OBR modules. In section 5.4, we present a case study that explains how the methodology of section 5.3 can be implemented in the form of an agent. The case study in section 5.4 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 section 5.3, are developed and explained in detail in this section, and simulation are performed. Section 5.5 concludes the study with future directions. Appendix B.1 discusses the simulator used in this study, and illustrates a floor map of the VE used by participants while performing the experiment as explained in Section 5.4. In Appendix B.2, we give an account on RPDM. Appendix C describes the assumptions that have been made during the knowledge elicitation process. Appendix D and Appendix E provide computer codes used in the development of the proposed agent model.

5.2 Background concepts

5.2.1 Ontology

Ontology is defined as, "The study of the categories of things that exist or may exist in some domain" (Sowa, 2000, 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 (Kabbaj, 2006; Kabbaj, Bouzoubaa, & Soudi, 2005), 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 CG, which is a bipartite graph between concept nodes and the relations among the concepts (Sowa, 1984). 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 (Section 5.3.2), 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 (Hadzic, Wongthongtham, Dillon, & Chang, 2009).

5.2.2 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, …, X_n can be developed to understand the influence of a site, i.e., a variable, on its neighbors (Raedt et al., 2016) as defined below:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{[k]}) \tag{5.1}$$

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, $Z = \sum_{x \in \Omega} \prod_k \phi_k(x_{[k]})$.

5.2.2.1 Markov Logic Network

Because a random variable assigned with a value can be considered as a proposition (Halpern, 2003, p. 58). Domingos and Richardson (2007) 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_is being the formulas and w_is being the weights assigned to the formulas.

If $C = \{c_1, c_2, \ldots, c_{|C|}\}$ 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:

$$P(X = x) = \frac{1}{Z}\exp\left(\sum\nolimits_i w_i\, n_i(x)\right) = \frac{1}{Z}\prod\nolimits_i \phi_i\left(x_{[i]}\right)^{n_i(x)} \quad (5.2)$$

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 $\phi_i\left(x_{[i]}\right) = e^{w_i}$.

5.3 Methodology

The kind of situations suitable for constructing a realization of the RPDM approach for artificial agents should include the ingredients of NDM (Orasanu & Connolly, 1993). 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 (Bratman, 1987). Specifically, these mental attitudes are a belief, desire, and intention, which are the basis of the BDI-agent model (Rao & Georgeff, 1995). The proposed agent model has a *beliefbase* that contains context information, past experiences, an ontology (Sowa, 2000) 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 5.2 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 performs mental simulation as a cause-and- effect mechanism

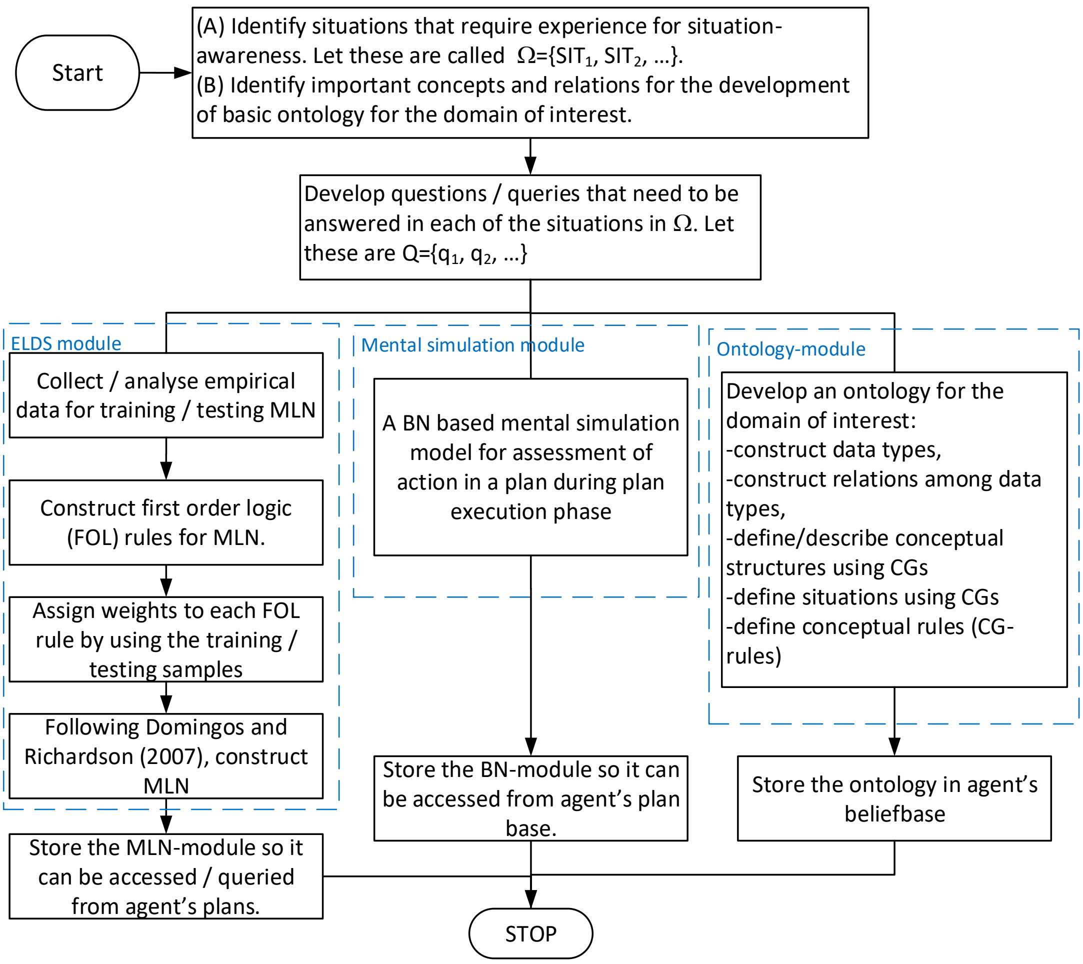


Figure 5.2. Basic steps to implement the method of realization of the RPDM based agent decision-making approach.

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 5.3 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

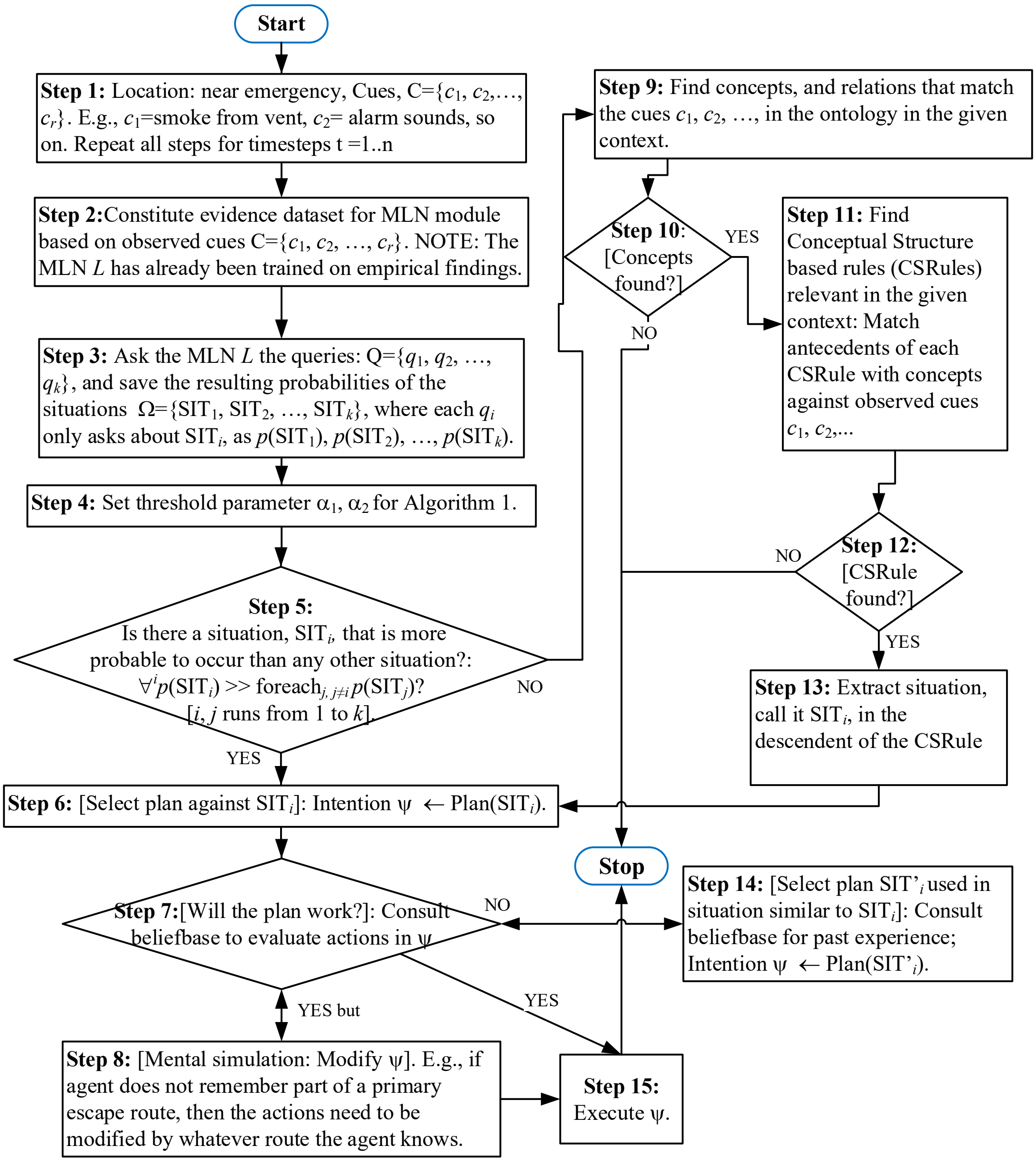


Figure 5.3. Activities in the process of developing a realization of RPDM.

Algorithm 5.1: A general higher-level decision-making.

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

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.

for *timesteps* = 1 to *n* **do**
    if $p(\texttt{SIT}_1) > \alpha_1$ && $[p(\texttt{SIT}_2) < \alpha_2, p(\texttt{SIT}_3) < \alpha_2, \ldots, p(\texttt{SIT}_m) < \alpha_2]$ **then**
        $\Phi \leftarrow \texttt{PLAN}_{\texttt{SIT}_1}$
    else if $p(\texttt{SIT}_2) > \alpha_1$ && $[p(\texttt{SIT}_1) < \alpha_2, p(\texttt{SIT}_3) < \alpha_2, \ldots, p(\texttt{SIT}_m) < \alpha_2]$ **then**
        $\Phi \leftarrow \texttt{PLAN}_{\texttt{SIT}_2}$
    ⋮
    else if $p(\texttt{SIT}_k) > \alpha_1$&& $[p(\texttt{SIT}_1) < \alpha_2, \ldots, p(\texttt{SIT}_{k-1}) < \alpha_2, p(\texttt{SIT}_{k+1}) < \alpha_2, \ldots, p(\texttt{SIT}_m) < \alpha_2]$ **then**
        $\Phi \leftarrow \texttt{PLAN}_{\texttt{SIT}_k}$
    else if $|p(\texttt{SIT}_1) - p(\texttt{SIT}_2)| \leq \varepsilon$ &&, …, && $|p(\texttt{SIT}_{m-1}) - p(\texttt{SIT}_m)| \leq \varepsilon$ **then**
        $\Phi \leftarrow \texttt{PLAN}_{\texttt{DIAGNOSE-SITUATION}}$
return Φ.

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 5.1 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 Figure 5.3. 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 (G. Klein, 1998 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 $\{c_1, c_2, \ldots\}$, based on MLN L that is developed by using the FOL rules.

5.3.1 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. Klein's RPDM

Table 5.2. The FOL rules for developing the MLN *L* suitable for emergency response in `FIRE` and `EVACUATE` emergencies.

#	Rules
1.	`¬L(ag, al, t) ⇒ ¬R (ag, al, t).`
2.	`L(ag,+al,t)^HITR(ag,+mloc,t)^ BST(ag,+al,t)⇒R(ag,+al,t)`
3.	`L(ag, al, t) ⇒ HSES(ag)`
4.	`ST(ag, thrt, t) ⇒ HSES(ag)`
5.	`(HFO(ag,+p_a,t)^FPA(ag,+p_a,t)^KETPA(+p_a,+emgType))v (ST(ag,+thrt,t)^KETT(+thrt,+emgType))v (L(ag,+al,t)^HITR(ag,+mloc,t)^KETA(+al,+emgType)^BST(ag, +al,t)) ⇒ HES(ag, +emgType, t)`
6.	`HES(ag,FIRE,t0)^HES(ag,EVACUATE,t1)^Gt(t1,t0) ⇒ ¬HES(ag, FIRE, t1)`
7.	`HES(ag,FIRE,t) ⇒ ¬HES(ag, EVACUATE,t)`
8.	`HES(ag, EVACUATE, t) ⇒ ¬HES(ag, FIRE, t)`
9.	`HFO(ag, +p_a,+t)^FPA(ag, +p_a,+t)^KMLPA(+p_a,+mloc) ⇒ HITR(ag,+mloc,+t)`
10.	`L(ag,+al,t)^R(ag,+al,t)^BST(ag,+al,t)^KMLA (+al,+mloc) ⇒ HITR(ag,+mloc,t)`

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 (1988), 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 5.2. As an example of how agents with different experiences can be made in a real system, consider rule#9 in Table 5.2:

$$\mathtt{HFO(a_1,+p\_a,+t)\hat{}FPA(ag,+p\_a,+t)\hat{}KMLPA(+p\_a,+mloc)} \Rightarrow \mathtt{HITR(ag,+mloc,+t)}$$

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, 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={PAGPA, PAPAPA}, T={t0, t1}, and M={MESSHALL, 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^{16} 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 5.3, 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 $p(\{\mathrm{HFO}, \mathrm{FPA}, \mathrm{KMLPA}, \neg\mathrm{HITR}\})$ is equal to $1/Z$ is less likely than all other

Table 5.3. Joint probability table for possible worlds entails by rule#9. The probability p({HFO,FPA,KMLPA,¬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 '⇒' is for logical implication.

HFO	FPA	KMLPA	J_1=HFO^FPA^KMPLA	J_2=HITR	$J_1 \Rightarrow J_2$	$p(.)$
¬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

probabilities as shown in Table 5.3, provided $w > 0$. Here Z is the partition function as described in Section 5.2.2. 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.

5.3.1.1 An Explanation of the FOL rules

A set of FOL rules are proposed in Table 5.2 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 (Chowdhury, 2016; ExxonMobil, 2010; Proulx, 2007; J. Smith, 2015; Sneddon et al., 2013; Spouge, 1999; Thilakarathne, 2015; Tong & Canter, 1985; Tutolo, 1979; Wankhede, 2017). Similar work is reported in (Musharraf, Smith, Khan, & Veitch, 2018) 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 5.2 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 (Proulx, 2007; Tong & Canter, 1985; Winerman, 2004) show that people need training to be vigilant about alarm sounds.

There are several sources that give *intention* a vital role in deliberation (Bratman, 1987; Thilakarathne, 2015). 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 (Bratman, 1987, 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 (Bratman, 1987, 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 & 4 have the same descendent: *Has Some Emergency Situation*, which is referred to here by the predicate name `HSES` (see Table 5.2). 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 & 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 (Chowdhury, 2016; ExxonMobil, 2010; Spouge, 1999; Wankhede, 2017). 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 (Sneddon et al., 2013). 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 (Bratman, 1987) 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 & 8 are to ensure that FIRE and EVACUATE are two distinct types of situations, besides that EVACUATE may occur because of a fire (Chowdhury, 2016; Spouge, 1999). 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

(Bratman, 1987, p. 56) is based on deliberation about the act of listening and recognizing the alarm type.

5.3.1.2 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 (Domingos & Lowd, 2009; Singla & Domingos, 2005) using the software package Alchemy (2012) so that weights can be assigned to the rules presented in Table 5.2. A fragment of the MLN *L* is depicted in Figure 5.4. The nodes in Figure 5.4 are obtained for each possible grounding of each predicate appearing in a formula. An edge between two nodes means that the

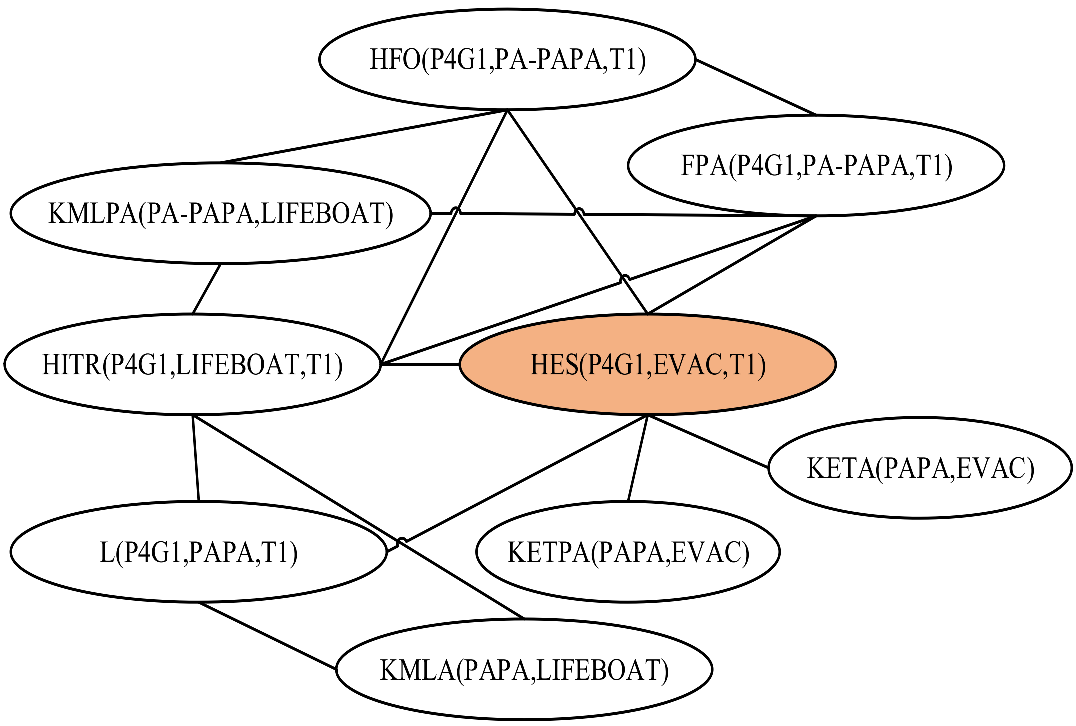


Figure 5.4. A portion of the MLN *L* obtained by grounding the predicates in Rules 2, 5, and 9 using the constants/facts obtained from Group 1 dataset. The above network was obtained by using facts/data for the participant P4G1 only.

corresponding ground predicates have appeared at the same time in at least one grounding of one formula in *L*.

5.3.2 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 (1984, 2000), that is, by using CGs. Figure 5.5 shows a fragment of important concepts represented in the proposed ontology for offshore emergency situations.

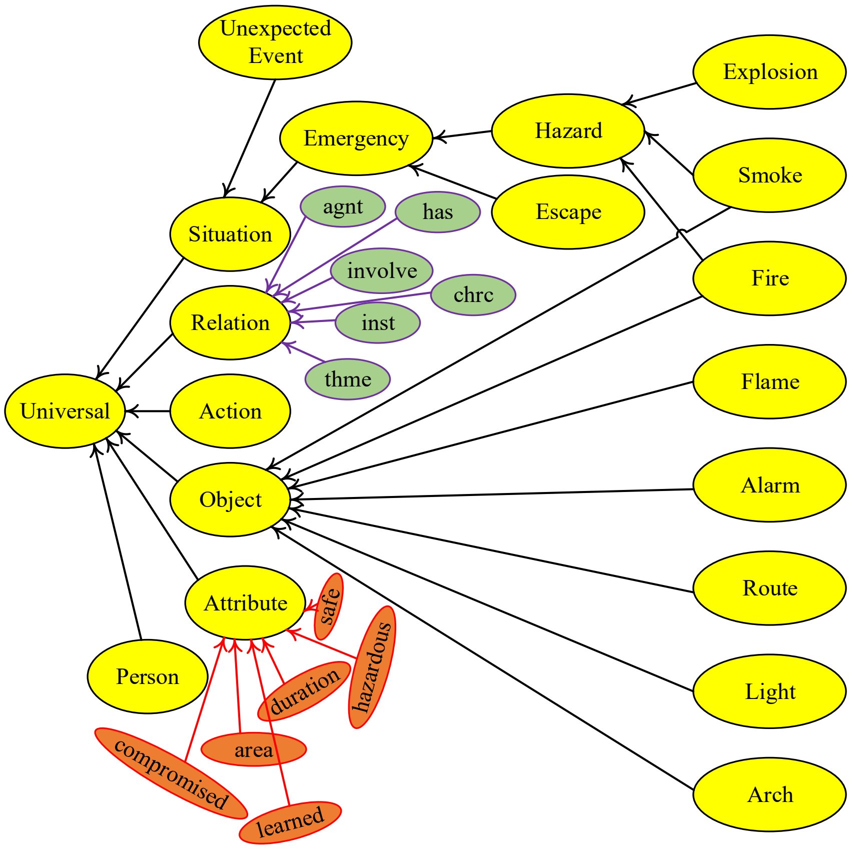


Figure 5.5. Fragment of the proposed ontology for offshore emergency awareness.

The conceptual relations: *agent* (`agnt`), *attribute* (`attr`), *characteristic* (`chrc`), *experiencer* (`expr`), *instrument* (`inst`), *object* (`obj`), and *theme* (`thme`) are used here as defined in (Sowa, 1984, pp. 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 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 5.1. 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:

[MoveTo] – (`agnt`) → [Person],
– (`attr`) → [Destination].

Definition 5.2. The relation `attr` links [Entity: *x] to [Entity: *y], where *x has an attribute *y. Example: Fire has flame. The CG is: [Fire] → (`attr`) → [Flame] such that Fire and Flame are represented as two concepts of type Entity, and Fire has an attribute Flame.

Definition 5.3. 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`] → (`chrc`) →[`danger`] → [`Person_Property`].

Definition 5.4. 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]` → `(expr)` → `[Person]`.

Definition 5.5. The relation `inst` links an [Entity] to an [Act] in which the entity is causally involved. For example, the CG `[Fire]` ← `(obj)` → `[Produce]` → `(inst)` → `[Combustion]` reflects a causal relationship between the chemical process of combustion and the birth of a fire.

Definition 5.6. The relation `obj` links an [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 descendent of a CS-Rule as:

Antecedent part

```
[MESSHALL] – (attr) → [Compromised],
           – (expr) → [Person].
```

Descendent part

```
[MoveTo] – (agnt) → [Person],
         – (attr) – [Destination] – (obj) → [LIFEBOAT].
```

Definition 5.7. 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)` → `[Hazard]` (see (Sowa, 2000, pp. 506-512) for a detail account on thematic roles in ontologies).

Definition 5.8. 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) →[Hazard],
        – expr → [Person].
```

Descendent part:

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

Definition 5.9. The concept `Combustion` is defined as an act of burning. The CG is: `[Combustion] –(actOf)→[Burning]`.

Definition 5.10. 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:

```
[Fire] – (attr) →[Heat],
       – (attr) →[Flame],
       – (attr) →[Light],
       ← (obj) – [Produce]–(inst)→ [Combustion].
```

Definition 5.11. The concept Smoke is defined as a child concept of `[Hazard]` that is produced as a result of combustion. The CG is:

```
[Produce] – (inst) →[Combustion],
          – (obj) →[Hazard: super].
```

Definition 5.12. The concept of muster station is defined as a place of temporary refuge during an emergency. It is represented as:

```
[MusterStation] – (attr) –
            → [TemporaryRefugeArea] – (attr) →[Duration] –
            – (involve) → [Emergency].
```

Definition 5.13. The concept of emergency is classified as a situation, and as a state. It is formally defined in terms of a CG as:

```
[Emergency] –
        –(isa)→[UnexpectedEvent]–(isa)→[Situation :super ],
        –(req)→[ImmediateAction],
        –(attr) → [Duration],
        –(attr) → [Area],
        –(chrc) → [Danger]–(to)→[Person_Property],
        –(involve) → [Hazard],
        –(expr) → [Person],
        –(notifiedBy) → [Alarm].
```

Definition 5.14. 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:

```
[MESSHALL] –
        –(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].
```

5.4 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 Figures 5.2 & 5.3. 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 Section 3 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 (G. Klein, 1998). We will describe an experiment that has been used here for developing some situations in which the proposed methodology of Section 5.3 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.

5.4.1 Human-competence measurement in a virtual environment

J. Smith (2015) performed an experiment to assess how training in a 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 (IMO, 2009) describes specific guidelines about the use of exit signs in escape routes on offshore petroleum platforms. The OSHA fact sheet (OSHA, 2018) 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 (Chowdhury, 2016; Crowl & Louvar, 2011). 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 (Figure 5.6).

5.4.2 Evacuation scenarios and decision tasks

The training curriculum of Smith's study (J. Smith, 2015) 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 Section 5.3. The agent is supposed to operate given the same input as was perceived by participants in 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-minute 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

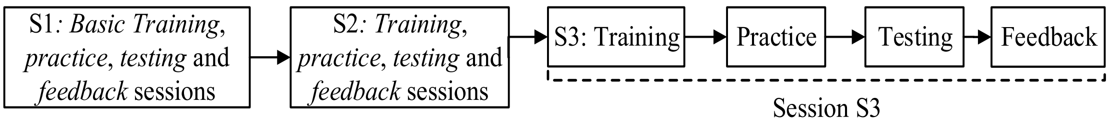


Figure 5.6. Each session S1-S3 comprises various training, practice, and testing sessions. Group 1 participants received repeated training and testing throughout the experiment.

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 (J. Smith, 2015). 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

Table 5.4. 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[14] 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 *before seeing a threat* (`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 & 4 an agent will get a sense of some emergency without getting further details.

proceeded in a scenario towards making a required decision. Factors that play important roles in deciding about the kind of emergency (`FIRE` or `EVACUATE`),

[14] 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.".

recognizing alarms, and developing an intention to move to a particular muster location using an escape route are listed in Table 5.4.

5.4.3 Data collection

All observations were collected in the form of Boolean variables or predicates reported in Table 5.4. Table 5.5 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 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" (Crandall, Klein, & Hoffman, 2006, pp. 14-15). There are three methods to perform this type of data collection. The first is based on the approach called *theory 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 (Davies & Stone, 1995). The second approach to mind reading, called *rationality theory* (RT), exploits the use of principles of rationality (Dennett, 1987) 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 (Blakemore & Decety, 2001; Shanton & Goldman, 2010) as simulation theory (ST). Appendix C describes 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

Table 5.5. A sample of empirical observations for the decision choices made by the five participants. 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.

	Predicates							
Participant	L (.,GPA,t0)	R (.,GPA,t0)	BST (.,GPA,t0)	HITR (.,MSH,t0)	ST	HFO (.,PAGPA,t0)	FPA (.,PAGPA,t0)	HES (.,FIRE,t0)
P4G1	True	False	False	False	True[15]	True	False	True
P5G1	True	True	True	True	True[16]	True	True	True
P6G1	True	True	True	False[17]	True[18]	True	True	True[19]
P7G1	True	True	True	False	True[20]	True	True	True
P10G1	True	True	True	True	True[21]	True	True	True
	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[22]	True	True	True
P5G1	True	True	True	True	True[23]	True	True	True
P6G1	True	True	True	True	True[24]	False	False	True[25]

[15] Parameters observed are: P4G1, SMK_VENT, and `t0`

[16] Parameters observed are: P5G1, SMK_STAI, and `t0`

[17] The intention for moving to the messhal was developed late during the beginning of the interval `t1`.

[18] Parameters observed are: P6G1, SMK_STAI, and `t0`

[19] P6G1 kept the impression of `FIRE` until it recognizes `EVACUATE` situation in `t1`.

[20] Parameters observed are: P7G1, SMK_STAI, and `t0`

[21] Parameters observed are: P10G1, SMK_MSHA, and `t0`

[22] P4G1 did not see any threat during `t1`

[23] Parameters observed are: P5G1, SMK_MSHA, and `t1`

[24] Parameters observed are: (P6G1, SMK_MSHA, `t1`), and (P6G1, SMK_STAI,`t1`)

[25] P6G1 kept impression of `FIRE` emergency during the beginning of the interval `t1`, 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`).

Participant	Predicates L (.,GPA,t0)	R (.,GPA,t0)	BST (.,GPA,t0)	HITR (.,MSH,t0)	ST	HFO (.,PAGPA,t0)	FPA (.,PAGPA,t0)	HES (.,FIRE,t0)
P7G1	True	True	True	True	True[26]	False	False	True[27]
P10G1	True	False	False	False	True[28]	False	False	False[29]

[26] Parameters observed are: P7G1, SMK_MSHA, and t1

[27] P7G1 behaved same as P6G1 and kept impression of FIRE during some time in the beginning of the interval t1

[28] Parameters observed are: P10G1, SMK_MSHA, and t1

[29] P10G1 could not recognize EVACUATE situation and kept the impression of FIRE situation till the end of the scenario.

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 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.

5.4.4. 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 (Dastani & Testerink, 2014), which is available as a Java API, (2) Alchemy 2.0 (2012) that supports Markov logic network development, and (3) the Amine platform (Kabbaj, 2006; Kabbaj, Bouzouba, El Hachimi, & Ourdani, 2006; Kabbaj et al., 2005) 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 (Poon & Domingos, 2006) inference algorithm is used for querying the ELDS module. Table 5.6 reports the simulated results along with the evidence data Te that is used to make inference from the MLN in ELDS module.

5.4.4.1 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 `t0`, 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 5.6. 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 `t0`, and 0.46 for moving to `LIFEBOAT` station during `t0` (i.e., `HITR(., LIFEBOAT, t0)`), 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 `t0` 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 5.1 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 5.1 based on its implementation in Appendix D.1, determines that during `t0` 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

Table 5.6. 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.

	Predicates							
Sit #	`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[30] (0.94_{H1},0.64_{H2})
2A	True	True (0.87)	True	True (0.66_{M1}, 0.43_{M2})	True[31]	True	True	True (0.96_{H1}, 0.26_{H2})
3A	False	False (0.0)	False	False (0.51_{M1},0.51_{M2})	True[32]	False	False	False (0.44_{H1},0.24_{H2})
	`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	True (0.46_{M1},0.50_{M2})	True[33]	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})

NOTE:

(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 n_{H1}, and the probability that an `EVACUATE` emergency has occurred is referred to by n_{H2}, where $0 \leq n \leq 1$.

30 The participant kept impression of `FIRE` during some time in the beginning of the interval `t1`.

31 Observed SMK_VENT during `t0`

32 Observed SMK_VENT during `t0`

33 SMK_MSHA, SMK_STAI, SMK_VENT all observed during `t1`.

primary muster station (see the value `true` in the last column of row 1A in Table 5.6) during interval `t0`.

5.4.4.2 Situation # 1B

The empirical findings during the second half of the testing scenario for participant P1G1 is reported in Sit#1B in Table 5.6 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 `t1`, 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 `t0`, the interval `t0` had ended and the `PAPA` alarm started sounding. The presence of smoke was a visual cue that has a dominance (Reason, 1990) 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 `t1` 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 5.2) 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, t1) results true has been found to be 0.5. This value is inconclusive based on Algorithm 5.1. While the agent is present in the MESSHALL (due to the decision in Situation 1A as reported in Section 5.4.4.1), and smoke was in the MESSHALL, the agent perceived the smoke, determined its current position (which was MESSHALL), and passed this information in the form of the following CG:

```
[MESSHALL]-(thme) →[Smoke]                                        (5.3)
```

to the OBR- module (reported in Section 5.3.2). A match of the CG in (5.3) 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:

```
[MESSHALL]-                                                       (5.4)
            - (attr) → [Compromised],
            - (expr) → [Person].
```

The above CG (5.4) has further been considered as antecedent of CS-rule#2, and the final inferred output is the following CG:

```
[MESSHALL]-                                                       (5.5)
     - (attr) → [Compromised],
     - (thme) → [Smoke],
     - (expr) →[Person]←(agnt)-[MoveTo]-
          - (attr)→[Destination]-(obj)→[LIFEBOAT].
```

This final CG (5.5) 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.

5.4.4.3 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 5.6 (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 5.1 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`.

5.4.4.4 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 5.6 (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.

5.4.4.5 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 (Reason, 1990) 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 5.6 (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:

```
[MESSHALL]-                                              (5.6)
          - (attr) → [Compromised],
          - (expr) → [Person].
```

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 (5.7).

```
[MESSHALL]-                                                  (5.7)
    - (attr) → [Compromised],
    - (thme) → [Smoke],
    - (expr) →[Person]←(agnt)-[MoveTo]-
        - (attr)→[Destination]-(obj)→[LIFEBOAT],
```

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

5.4.4.6 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 (5.7). 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, G. Klein (1998) 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 E, 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 (Buckland, 2004; Millington & Funge, 2009) 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`, `HITR`, and `HES` have been found to be 0.9, 0.94, and 0.96 respectively (see Table 5.6, 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 (Danial et al., 2018, 2019). 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 Figure 5.7) 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.

5.5. 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.

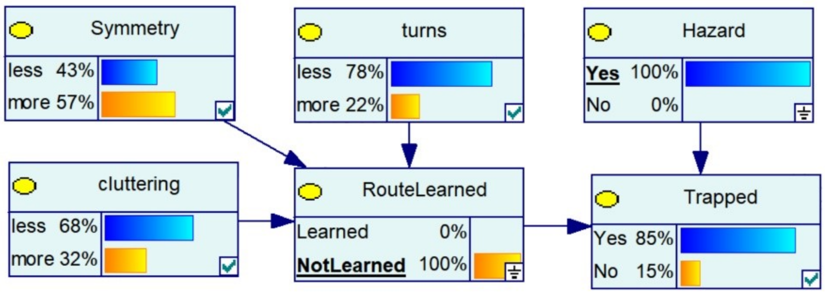


Figure 5.7. A model of mental simulation during deliberation of the plan of moving to the `LIFEBOAT` station. The agent weighs its chances of being trapped for each case of choosing PER and SER.

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 E describes a pseudo-code 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.

Declarations

Author's contributions

S. N. Danial performed the development and implementation of the proposed agent model and performed knowledge elicitation from Smith's (2015) experiment for validation of the model. J. Smith performed the experiment presented in (Smith, 2015) and verfied the data extracted from the experiment. F. Khan supervised the technical development of the proposed agent model. B. Veitch supervised the entire study and performed the editorial process. All authors read and approved the final draft.

Acknowledgments

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.

Competing interests

The authors declare that they have no competing interests.

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.

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.

References

Alchemy. (2012). *Alchemy: A software for statistical relational learning and probabilistic logic inference based on Markov logic representation*. Washington DC.

Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge, MA: MIT Press.

Blakemore, S.-J., & Decety, J. (2001). From the perception of action to the understanding of intention. *Nature Reviews Neuroscience*, *2*(8), 561–567. https://doi.org/10.1038/35086023

Bratman, M. (1987). *Intention, plans, and practical reason*. Cambridge, MA: Harward University Press.

Buckland, M. (2004). *Programming Game AI by Example*. Jones & Bartlett Learning.

Canellas, M. C., & Feigh, K. M. (2016). Toward Simple Representative Mathematical Models of Naturalistic Decision Making Through Fast-and-Frugal Heuristics. *Journal of Cognitive Engineering and Decision Making*, *10*(3), 255–267. https://doi.org/10.1177/1555343416656103

Chase, W. G., & Simon, H. A. (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

Chowdhury, S. (2016). *Optimization and Business Improvement: Studies in Upstream Oil and Gas Industry*. New Jersey: Wiley.

Crowl, D. A., & Louvar, J. F. (2011). *Chemical Process Safety: Fundamentals with applications* (3rd ed.). Boston: Pearson Education, Inc.

Danial, S. N., Khan, F., & Veitch, B. (2018). A Generalized Stochastic Petri Net model of route learning for emergency egress situations. *Engineering Applications of Artificial Intelligence*, *72*, 170–182. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0952197618300733

Danial, S. N., Smith, J., Khan, F., & Veitch, B. (2019). Human-Like Sequential Learning of Escape Routes for Virtual Reality Agents. *Fire Technology*, *55*(3), 1057–1083. https://doi.org/10.1007/s10694-019-00819-7

Dastani, M., & Testerink, B. (2014). From Multi-Agent Programming to Object Oriented Design Patterns. In F. Dalpiaz, J. Dix, & M. B. van Riemsdijk (Eds.), *Engineering Multi-Agent Systems. EMAS 2014. Lecture Notes in Computer Science, Volume 8758.* (pp. 204–226). https://doi.org/10.1007/978-3-319-14484-9_11

Davies, M., & Stone, T. (Eds.). (1995). *Folk Psychology: The Theory of Mind Debate*. Oxford, UK: Blackwell Publishers.

de Groot, A. D. (1965). *Thought and Choice in Chess* (1st ed.). The Hague, The Netherlands: Mouton Publishers.

Dennett, D. C. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.

Devlin, K. J. (1991). *Logic and Information*. Cambridge: Cambridge University Press.

Domingos, P., & Lowd, D. (2009). Markov Logic: An interface layer for Artificial Intelligence. In T. Brachman, R. J.;Dietterich (Ed.), *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Seatle: Morgan & Claypool Publishers.

Domingos, P., & Richardson, M. (2007). Markov Logic: A Unifying Framework for Statistical Relational Learning. In B. Getoor, L.;Taskar (Ed.), *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Endsley, M. R. (1988). Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *32*. https://doi.org/10.1177/154193128803200221

ExxonMobil. (2010). *OIMS: System 10-2 Emergency Preparedness and Response.* Retrieved from https://www.cnsopb.ns.ca/sites/default/files/inline/12450_so41877.1_spill_response_soei_0.pdf

Fan, X., McNeese, M., Sun, B., Hanratty, T., Allender, L., & Yen, J. (2010). Human–Agent Collaboration for Time-Stressed Multicontext Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *40*(2), 306–320. https://doi.org/10.1109/TSMCA.2009.2035302

Hadzic, M., Wongthongtham, P., Dillon, T., & Chang, E. (2009). Current issues and the need for ontologies and agents. In *Ontology-Based Multi-Agent Systems* (pp. 1–14). Berlin, Heidelberg: Springer-Verlag.

Halpern, J. Y. (2003). *Reasoning about uncertainty*. Cambridge, MA: MIT Press.

Hassard, S. T. (2009). The variations of recognition primed decision-making and how it informs design decision-making. *NDM'09 Proceedings of the 9th Bi-Annual International Conference on Naturalistic Decision Making*, 57. Swindon, UK: BCS Learning & Development.

Hu, Y., Li, R., & Zhang, Y. (2018). Predicting pilot behavior during midair encounters using recognition primed decision model. *Information Sciences*, *422*, 377–395. https://doi.org/10.1016/j.ins.2017.09.035

Hutton, R. J. B., Warwick, W., Stanard, T., McDermott, P. L., & McIlwaine, S. (2001). Computational Model of Recognition-Primed Decisions (RPD): Improving Realism in Computer-Generated Forces (CGF). *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *45*(26), 1833–1837. https://doi.org/10.1177/154193120104502607

IMO. (2009). *SOLAS: Consolidated Edition* (5th ed.). London, UK: International Maritime Organization.

Ji, Y., Massanari, R. M., Ager, J., Yen, J., Miller, R. E., & Ying, H. (2007). A fuzzy logic-based computational recognition-primed decision model. *Information Sciences*, *177*(20), 4338–4353. https://doi.org/10.1016/j.ins.2007.02.026

Kabbaj, A. (2006). Development of Intelligent Systems and Multi-Agents Systems with Amine Platform. In S. H., P. Hitzler, & P. Øhrstrøm (Eds.), *Conceptual Structures: Inspiration and Application. ICCS 2006. Lecture Notes in Computer Science, Volume 4068*. Berlin, Heidelberg: Springer.

Kabbaj, A., Bouzouba, K., El Hachimi, K., & Ourdani, N. (2006). Ontologies in Amine Platform: Structures and Processes. In H. Schärfe, P. Hitzler, & P. Øhrstrøm (Eds.), *Conceptual Structures: Inspiration and Application. ICCS 2006. Lecture Notes in Computer Science* (pp. 300–313). https://doi.org/10.1007/11787181_22

Kabbaj, A., Bouzoubaa, K. M., & Soudi, A. (2005). Amine Platform: an Artificial Intelligence Environment For the Development of Intelligent Systems. *First Information and Communication Technologies International Symposium ICTIS*. Tetuan, Morocco.

Klein, G. (1998). *Sources of Power*. Cambridge, MA: MIT Press.

Klein, G. (2004). *The Power of Intuition*. Doubleday.

Means, B., Salas, E., Crandall, B., & Jacobs, T. O. (1993). Training Decision Makers for the Real World. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods1* (pp. 306–326). Norwood, NJ: Ablex Publishing Corporation.

Millington, I., & Funge, J. (2009). *Artificial Intelligence for Games* (2nd ed.). Burlington, MA: Morgan Kaufmann Publishers.

Mueller, S. T. (2009). A Bayesian Recognitional Decision Model. *Journal of Cognitive Engineering and Decision Making*, *3*(2), 111–130. https://doi.org/10.1518/155534309X441871

Musharraf, M., Smith, J., Khan, F., & Veitch, B. (2018). Identifying route selection strategies in offshore emergency situations using decision trees. *Reliability Engineering & System Safety*. https://doi.org/10.1016/j.ress.2018.06.007

Naderpour, M., Lu, J., & Zhang, G. (2014). An intelligent situation awareness support system for safety-critical environments. *Decision Support Systems*, *59*, 325–340. https://doi.org/10.1016/j.dss.2014.01.004

Norling, E. (2004). Folk psychology for human modelling: Extending the BDI paradigm. *AAMAS '04 Proceedings of the Third International Conference on*

Autonomous Agents and Multiagent Systems. New York, NY: IEEE Computer Society, Washington DC.

Norling, E., Sonenberg, L., & Rönnquist, R. (2000). Enhancing Multi-Agent Based Simulation with Human-Like Decision Making Strategies. In S. Moss & P. Davidsson (Eds.), *Multi-Agent-Based Simulation. MABS 2000. Lecture Notes in Computer Science, Volume 1979*. Berling, Heidelberg: Springer.

Nowroozi, A., Shiri, M. E., Aslanian, A., & Lucas, C. (2012). A general computational recognition primed decision model with multi-agent rescue simulation benchmark. *Information Sciences*, *187*, 52–71. https://doi.org/10.1016/j.ins.2011.09.039

Orasanu, J., & Connolly, T. (1993). The reinvention of decision making. In G. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 3–20). Norwood, NJ: Ablex Publishing Corporation.

OSHA. (2018). Emergency exit routes factsheet. Retrieved April 22, 2019, from https://www.osha.gov/OshDoc/data_General_Facts/emergency-exit-routes-factsheet.pdf

Patterson, R., Fournier, L., Pierce, B., Winterbottom, M., & Tripp, L. (2009). Modeling the dynamics of recognition-primed decision making. *Proceedings of NDM9, the 9th International Conference on Naturalistic Decision Making*. London, UK: British Computer Society.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inferences*. San Mateo, CA: Morgan Kaufmann.

Poon, H., & Domingos, P. (2006). Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, 458–463. Retrieved from http://dl.acm.org/citation.cfm?id=1597538.1597612

Press, G. (2018, February 7). *The Brute Force of IBM Deep Blue and Google DeepMind*. Retrieved from https://www.forbes.com/sites/gilpress/2018/02/07/the-brute-force-of-deep-blue-and-deep-learning/#3729cb5c49e3

Proulx, G. (2007). Response to fire alarms. *Fire Protection Engineering*, *33*, 8–14. Retrieved from http://www.cfaa.ca/Files/flash/CODES/FIRE ALARM ARTICLES FOR THE AHJ/Fire Alarm Response.pdf

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

Rao, A. S., & Georgeff, M. P. (1995). BDI Agents: From theory to practice. In V. Lesser & L. Gasser (Eds.), *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* (pp. 312–319). Retrieved from 0262621029

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Resnick, M. (2001). Recognition Primed decision making in E-Commerce. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Santa Monica: Sage Journals Ltd.

Shanton, K., & Goldman, A. (2010). Simulation theory. *Wiley Interdisciplinary Reviews: Cognitive Science*, *1*(4), 527–538. https://doi.org/10.1002/wcs.33

Singla, P., & Domingos, P. (2005). Discriminative Training of Markov Logic Networks. *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, 868–873. Retrieved from http://dl.acm.org/citation.cfm?id=1619410.1619472

Smith, B. M. (2007). Working minds. In *Phi Delta Kappan* (Vol. 88). https://doi.org/10.1177/003172170708801001

Smith, J. (2015). *The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios [Master thesis]*. Memorial University of Newfoundland, St. John's, NL, Canada. Retrieved from: https://research.library.mun.ca/8401/1/thesis.pdf

Sneddon, A., Mearns, K., & Flin, R. (2013). Stress, fatigue, situation awareness and safety in offshore drilling crews. *Safety Science*, *56*, 80–88. https://doi.org/10.1016/j.ssci.2012.05.027

Sokolowski, J. A. (2003). Enhanced Decision Modeling Using Multiagent System Simulation. *SIMULATION*, *79*(4), 232–242. https://doi.org/10.1177/0037549703038886

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks/Cole Thomson Learning.

Spouge, J. (1999). *A Guide to Quantitative Risk Assessment for Offshore Installations*. Aberdeen, UK: CMPT Publication.

Thilakarathne, D. J. (2015). Modelling of situation awareness with perception, attention, and prior and retrospective awareness. *Biologically Inspired Cognitive*

Architectures, *12*, 77–104. https://doi.org/10.1016/j.bica.2015.04.010

Tong, D., & Canter, D. (1985). The decision to evacuate: a study of the motivations which contribute to evacuation in the event of fire. *Fire Safety Journal*, *9*(3), 257–265. https://doi.org/10.1016/0379-7112(85)90036-0

Tutolo, D. (1979). Attention: Necessary Aspect of Listening. *Language Arts*, *56*(1), 34–37. Retrieved from http://www.jstor.org/stable/41404756

Wankhede, A. (2017, September). Different Types of Alarms on Ships. *Marine Insight*. Retrieved from https://www.marineinsight.com/marine-safety/different-types-of-alarms-on-ship/

Winerman, L. (2004). Fighting fire with psychology. *Monitor on Pscyhology*, *35*(8), 28. Retrieved from https://www.apa.org/monitor/sep04/fighting

Zsambok, C. E. (1997). Naturalistic decision making: where are we now? In C. E. Zsambok & G. Klein (Eds.), *Naturalistic decision making* (pp. 3–16). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Zsambok, C. E., & Klein, G. A. (Eds.). (1997). *Naturalistic Decision Making*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Chapter 6

Conclusions & Recommendations

6.1 Conclusions

This thesis presents four research papers that describe the development of an agent model that exhibits a capability to make human-like decisions in evolving situations such as offshore emergencies. The agent model has three major components: (1) a spatial-learning module for learning escape routes, which are designated routes for emergency evacuation, (2) a situation awareness module, which is used to recognize emergency situations, and (3) a decision-support module that exploits modules in (1) and (2), and implements an RPDM based decision-logic for producing human-like responses during emergencies.

RPDM involves decision-making on the basis of *knowledge about the environment*. The environment is the place or platform where decisions are supposed to transform into actions. It could be an industrial setup where `FIRE` or other types of emergencies can occur, or an offshore oil & gas platform. RPDM also requires an ability to *perceive the changes in the situation* occurring in the environment, and develop an understanding so that the decision-maker can recognize new or emerging situations and interpret them, moment by moment, in the form of important cues, *plausible goals*, *actions to be done*, and expected results of performing the actions. RPDM was developed by considering how experts make decisions in real situations, therefore, it demands a good deal of experience of dealing with the situations in which decision-making is sought. Further, RPDM requires an ability to diagnose a situation on the basis of in-depth analysis of the input signals or cues in cases when the decision-maker is confronted with a totally new situation for which his experience has turned out to be insufficient. One distinguishing aspect of RPDM is the concept of *story building*. Story building works at two levels. The first is when the decision-maker foresees the impact of actions on the current state of the environment/situation. The second is when an unknown situation is discovered. In this case, the task of a decision-maker is to define the new situation in terms of previously seen situations.

The proposed agent model transforms the philosophical faculties of RPDM into concrete computational terms. Thus,

(1) the 'knowledge about the environment' is modeled as a spatial-learning module based on generalized stochastic Petri net, presented in Chapters 2 and 3. The approach used in the spatial learning module is unique in the sense that it exploits environmental features required in learning routes rather than providing the agent the complete knowledge of the environment, as is the case of conventional wayfinding methods such as A* algorithm, in the form of a navigation graph. This approach enables one to develop agents with different route knowledge in the same environment, i.e., an agent can be created that has partial or full route knowledge of a particular route. Agents with different skills of learning escape routes can be developed by using different sets of stochastic transition rates. Different sets of stochastic rates have been tested here and a bound for the range of rates of transitions have been developed on the basis of empirical findings. This means that the model output cannot go beyond these limits unless the ranges of

the rates of transitions defined in this work are altered. Based on this, an agent that has been created with transition rates selected near the upper bound of the range was found to be more prone to error (forgetting part of a route) than those agents that used rates near the lower bound.

(2) the 'constantly perceiving cues to recognize a situation' is modeled (see Chapters 4 & 5) keeping in mind that the presence of a cue has a special meaning as in 'smoke means fire'. Thus, FOL rules are considered the first step in quantifying a situation. Because FOL rules are hard constraints, a Markov logic network is developed to represent the situations of `FIRE` and `EVACUATE` emergencies. The network training exploits empirical findings. Based on training datasets, two different agents have been developed. The first agent was better at recognizing the `GPA` alarm than the second one. The second agent was found to be better in recognizing the `FIRE` and `EVACUATE` emergencies when more than one cue was present (such as platform alarms, PAs, hazard types, etc.) than the first one. The second agent was also found to be better in recognizing the `PAPA` alarm. Also, the first agent was more prone to keep the impression of a `FIRE` situation despite that the `FIRE` situation had escalated to an `EVACUATE` situation. The behavior of agents correspond to the average behavior of two groups of people who participated in J. Smith's (2015) experiment in recognizing the same situations.

(3) the 'diagnosing of a situation' is modeled (Chapter 5) as a method that uses ontological reasoning to classify situations. This part of the work models story

building in the form of formally representing a situation, and then inferring another situation in the same way a story progresses in real life. An agent shows this behavior only when the required experience to resolve a situation is inadequate. Conceptual graphs have been used to represent general knowledge about the situations involved. On top of conceptual graphs are conceptual structure rules that help agents infer a new situation on the basis of an observed situation. This phenomenon can also be regarded as level-3 of the Endsley's SA model (1988, 1995, 2000) that says people project their comprehension about a situation to foresee a future status of the situation. For example, an agent, as reported in section 5.4.4.5, makes a CG on the basis of perceived cues as [MESSHALL]–(thme)→[Smoke]. Because the agent possesses the ontological knowledge about how to interpret the situation when a messhall is filled with smoke, the agent can infer a new situation, [MESSHALL]–(attr)→[Compromised]. This new situation is further matched in the agent's repertoire of situations and resulted in still another situation (as reported in CG (5.7)) where the agent could find relevant actions to be performed, which was 'move to the LIFEBOAT station'.

The author concludes that agents developed in this work can produce human-like behaviors (decisions) as observed in empirical findings related with decision-making in offshore emergency evacuation situations. The agents reported here do not copy human-behavior, nonetheless, the decisions made are found to be more natural choices that were difficult to produce using a conventional decision-theoretic logic.

6.2 Contributions fulfilling the Research Objectives

Table 6.1. A side-by-side comparison of the contributions and the research objectives of this dissertation.

Title	Research objectives	Contributions fulfilling the research objectives
Chapter 2.	• To develop an agent model that simulates how much of a route will be remembered after being exposed to a route for the first time. The model should simulate human-like behavior of remembering a route when a person, on the average, sees the same route.	• Landmark saliency based classification is introduced in Algorithm 2.3 (Chapter 2). Based on this algorithm, the GSPN model (see Figure 2.4.) quantify how an agent can exhibit forgetting and remembering of the landmarks along a route that it traverses the first time. The probability of forgetting or remembering depends on the saliency of the nearby landmarks as well as on the stochastic transition rates so that the output satisfies the empirical evidence.
	• To develop agents with different skills of remembering parts of a route.	• Using different rates for the stochastic transitions as parameters to the GSPN model, one can develop agents with different memorizing abilities. For instance, see the results of using different types of agents in Figure 2.7 on page 65. This was achieved by employing different sets of transition rates as depicted in Table 2.2 on 57.

Title	Research objectives	Contributions fulfilling the research objectives
Chapter 3	To develop agents with different route knowledge in an environment.	• The method presented in Figure 3.8 on page 100 shows how an agent can run through learning a route over a series of training exposures. Each iteration makes the agent more informed about the route being traversed. By using agents with different ability to remember a route (using different sets of transition rates) one can witness how agents differ in their knowledge of a route.
Chapter 4	To develop agents with different experiences of recognizing fire and evacuation emergencies.	The method presented in Chapter 4 on page 131 in Figure 4.1 is used to develop SA model based on Markov logic network. This model was trained using two different training datasets obtained from an experiment as described in Section 4.4. The result of training are two agents which show differences in the SA recognition skills. In the same way several other agents can be trained with varying abilities to perform recognition tasks of certain situations.
Chapter 5	To develop agents with different ability to make decisions as to what needs to be done in an emergency situation in a similar way that people make decisions.	In chapter 5, a realization of RPDM based decision-making model for agent is proposed. The SA part of the model exploits experiences as well as domain knowledge to recognize evolving emergency situations. This agent model has been tested for different fire and emergency situations and the results are found to be consistent with the empirical findings.

6.3 Limitations, Recommendations & Future Work

1. In the present study, spatial learning is achieved as a linear process whereby the difficulty of remembering a landmark is gradually decreased, in a linear way, during successive training episodes. This can be improved by incorporating a dynamical system's approach where a learning curve is used to adjust the rates of stochastic transitions in successive training episodes. An interesting aspect of such modeling is in line with the concept of mass/action kinetics in chemical or bio-chemical reactions. Thus, rates of stochastic transitions will depend on some other factors, for example, the agent's ability to focus on environmental features, fatigue, the agent's ability to memorize, the amount of light available to see the environmental features, and the like. Interesting literature in this regard should begin with concepts in (Heiner & Gilbert, 2011).
2. Route learning gives rise to survey knowledge of the environment when a path integration technique is used. Agents having the ability to integrate paths can find shortcuts during emergency evacuation. The spatial learning module can be enhanced by employing path integration.
3. Based on the preceding paragraph, agents with the ability to create cognitive maps of an environment can be developed by exploiting techniques such as the one reported in (Golledge, 1977) where a real environment is mapped into agent's memory. The mapped environment does not include correct distances and exact turn angles, but still it works as an efficient guide to navigate along a path in the environment.

4. An important way to improve situation awareness should incorporate a distinction between important and unimportant cues, and relevant or irrelevant cues. Questions arise, such as how one cue is relevant at a single instance of time and how it can become irrelevant at some other times. For example, if an agent knows that there is a fire event because a fire is currently being observed, then the agent should consider this situation a `FIRE` situation. But at the same time, if the agent starts listening to a `PAPA` alarm, then it should forget about the `FIRE` situation and consider the present moment as having an `EVACUATE` situation.
5. Detection of anomaly after applying actions according to a plan is an important module in RPDM. Future work should consider how this can be achieved. The author suggests that only those agents that have detailed knowledge about the objects and their relationships (i.e., the ontological knowledge) might be suitable for anomaly detection because only then reasons of an anomaly could be discovered. Important techniques in this regard should include a fault tree analysis starting with the anomaly as a top node and digging down to the possible causes that may lead to the failure of actions in the selected plan.
6. An interesting area of study involves modeling the activity of writing a plan using simple actions, where actions are ordered in a particular fashion to produce the desired result. In contemporary times, an agent or any other computer program selects ready made plans and they do not write new plans from scratch. How can a new plan, due to a particular demand, be made at the execution time of a computer program? This is an important research problem. In the present work, the author has implemented a way to modify a given plan

(see Appendix E) but a general technique of making new plans out of a need is an interesting research problem with uncountable applications.

7. In the last, this work can be used to provide human-like abilities to agents acting as operators, or to agents assisting real operators in a complex system such as a nuclear plant. Literature on *human reliability analysis* (HRA) (Azarkhil, 2013; Azarkhil & Mosleh, 2014; Coyne, 2009; Y. Li, 2013; Mosleh & Chang, 2004; Reason, 1990) confirms that human fallibility has profound consequences in sensitive installations during emergencies. Because the decision-making (as modeled here) makes use of two types of situation assessment methods, the one based on *experience*, and the other based on *story building*, an interesting question for future work may be to explore how the proposed agent model can benefit HRA-based decision-making models[34] to facilitate human-error analysis.

[34] As an example see *Accident Dynamic Simulator-Information*, *Decision*, *and Action in a Crew* (ADS-IDAC) model of operator response in complex systems (Chang & Mosleh, 2007a, 2007b, 2007c, 2007d, 2007e) https://www.risksciences.ucla.edu/software/ads-idac

Bibliography

Abū Ḥāmid Muḥammad ibn Muḥammad Al-Ghazālī. (1998). *The niche of lights: A parallel English-Arabic text (translated by David Buchman)* (1st ed.; D. Buchman, trans.). Provo, UT, USA: Brigham Young University.

Ajmone Marsan, M. (1990). *Stochastic Petri Nets*. Springer.

Ajmone Marsan, M., Conte, G., & Balbo, G. (1984). A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Trans. Comput. Syst.*, *2*(2), 93–122. https://doi.org/10.1145/190.191

Akman, V., & Surav, M. (1996). Steps toward formalizing Context. *AI Magzine*, *17*(3). https://doi.org/https://doi.org/10.1609/aimag.v17i3.1231

Alanen, L. (2014). The Second Meditation and the nature of the human mind. In D. Cunning (Ed.), *The Cambridge Companion to Descartes' Meditations* (pp. 88–106). https://doi.org/10.1017/CCO9781139088220.005

Alchemy. (2012). *Alchemy: A software for statistical relational learning and probabilistic logic inference based on Markov logic representation*. Washington DC.

Allen, G. L. (1999). Spatial abilities, cognitive maps and wayfinding: bases for individual differences in spatial cognition and behavior. In R. G. Golledge (Ed.), *Wayfinding behavior* (pp. 46–80). The John Hopkins University Press.

Aristotle. (2001). *Aristotle's On the soul and on memory and recollection [Aristotle's De Anima]* (J. Sachs, trans.). Ann Arbor, Michigan, USA: Green Lion Press.

Azarkhil, M. (2013). *Dynamic behavior of operating crew in complex systems: an object-based modeling approach [PhD Dissertation]* (University of Maryland). Retrieved from https://drum.lib.umd.edu/bitstream/handle/1903/14179/Azarkhil_umd_0117E_14228.pdf?sequence=1&isAllowed=y

Azarkhil, M., & Mosleh, A. (2014). Dynamic behavior of operating crew in complex systems. *Proceedings - Annual Reliability and Maintainability Symposium*. https://doi.org/10.1109/RAMS.2014.6798446

Aziz, A. (2000). Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, *1*(1), 162–170.

Balbo, G., Chiola, G., & Bruell, S. C. (1992). An example of modeling and evaluation of a concurrent program using colored stochastic Petri nets: Lamport's fast mutual exclusion algorithm. *IEEE Transactions on Parallel and Distributed Systems*, *3*(2), 221–240.

Barwise, J. (1981). Scenes and Other Situations. *The Journal of Philosophy*, *78*(7), 369. https://doi.org/10.2307/2026481

Barwise, J., & Perry, J. (1980). *The Situation Underground*. California: Stanford Cognitive Science Group 1980, Section D.

Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge, MA: MIT Press.

Bause, F., & Kritzinger, P. S. (1996). *Stochastic Petri Nets: An Introduction to the Theory*. Verlag Viewweg.

BBC, A. N. (2013, October). *Nairobi siege: How the attack happened*. Retrieved from https://www.bbc.com/news/world-africa-24189116

Beusmans, J. M., Aginsky, V., Harris, C. L., & Rensink, R. A. (1995). Analyzing situation awareness during wayfinding in a driving simulator. In D. J. Garland & M. R. Endsley (Eds.), *Proceedings of the International Conference on Experimental Analysis and Measurement of Situation Awareness* (pp. 245–251). Daytona Beach, Florida: Embry-Riddle Aeronautical University Press.

Blakemore, S.-J., & Decety, J. (2001). From the perception of action to the understanding of intention. *Nature Reviews Neuroscience*, *2*(8), 561–567. https://doi.org/10.1038/35086023

Borowiec, S., & Lien, T. (2016, March). *AlphaGo beats human Go champ in milestone for artificial intelligence*. Retrieved from https://www.latimes.com/world/asia/la-fg-korea-alphago-20160312-story.html

Bosse, T., & Mogles, N. (2014). Spread of situation awareness in a group: Population-based vs. agent-based modelling. *Proceedings - 2014 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2014*, *3*, 1117–1124. https://doi.org/10.1109/WI-IAT.2014.169

Bratman, M. (1987). *Intention, plans, and practical reason*. Cambridge, MA: Harward University Press.

Buckland, M. (2004). *Programming Game AI by Example*. Jones & Bartlett Learning.

Caduff, D., & Timpf, S. (2005). The Landmark Spider: Representing Landmark Knowledge for Wayfinding Tasks. *AAAI Spring Symposium - Technical Report*, 30–35.

Canellas, M. C., & Feigh, K. M. (2016). Toward Simple Representative Mathematical Models of Naturalistic Decision Making Through Fast-and-Frugal Heuristics. *Journal of Cognitive Engineering and Decision Making*, *10*(3), 255–267. https://doi.org/10.1177/1555343416656103

Chang, Y. H. J., & Mosleh, A. (2007a). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents: Part 1: Overview of the IDAC model. *Reliability Engineering & System Safety*, *92*(8), 997–1013. https://doi.org/10.1016/j.ress.2006.05.014

Chang, Y. H. J., & Mosleh, A. (2007b). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents: Part 3: IDAC operator response model. *Reliability Engineering & System Safety*, *92*(8), 1041–1060. https://doi.org/10.1016/j.ress.2006.05.013

Chang, Y. H. J., & Mosleh, A. (2007c). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents: Part 5: Dynamic probabilistic simulation of the IDAC model. *Reliability Engineering & System Safety*, *92*(8), 1076–1101. https://doi.org/10.1016/j.ress.2006.05.012

Chang, Y. H. J., & Mosleh, A. (2007d). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents. Part 2: IDAC performance influencing factors model. *Reliability Engineering & System Safety*, *92*(8), 1014–1040. https://doi.org/10.1016/j.ress.2006.05.010

Chang, Y. H. J., & Mosleh, A. (2007e). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents. Part 4: IDAC causal model of operator problem-solving response. *Reliability Engineering & System Safety*, *92*(8), 1061–1075. https://doi.org/10.1016/j.ress.2006.05.011

Chase, W. G., & Simon, H. A. (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

Chowdhury, S. (2016). *Optimization and Business Improvement: Studies in Upstream Oil and Gas Industry*. New Jersey: Wiley.

Cornfield, J., Haenszel, W., Hammond, E. C., Lilienfeld, A. M., Shimkin, M. B., & Wynder, E. L. (2009). Smoking and lung cancer: Recent evidence and a discussion of some questions. *International Journal of Epidemiology*, *38*(5), 1175–1191. https://doi.org/10.1093/ije/dyp289

Coyne, K. A. (2009). *A predictive model of nuclear power plant crew decision-making and performace in a dynamic simulation environment [PhD Dissertation]* (University of Maryland). Retrieved from https://drum.lib.umd.edu/bitstream/handle/1903/9888/Coyne_umd_0117E_10842.pdf?sequence=1&isAllowed=y

Crowl, D. A., & Louvar, J. F. (2011). *Chemical Process Safety: Fundamentals with applications* (3rd ed.). Boston: Pearson Education, Inc.

Cullen, L. W. D. (1993). The public inquiry into the Piper Alpha disaster. *Drilling Contractor (United States)*, *49:4*.

Danial, S. N., Khan, F., & Veitch, B. (2018). A Generalized Stochastic Petri Net model of route learning for emergency egress situations. *Engineering Applications of Artificial Intelligence*, *72*, 170–182. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0952197618300733

Danial, S. N., Smith, J., Khan, F., & Veitch, B. (2019). Human-Like Sequential Learning of Escape Routes for Virtual Reality Agents. *Fire Technology*, *55*(3), 1057–1083. https://doi.org/10.1007/s10694-019-00819-7

Dastani, M., & Testerink, B. (2014). From Multi-Agent Programming to Object Oriented Design Patterns. In F. Dalpiaz, J. Dix, & M. B. van Riemsdijk (Eds.), *Engineering Multi-Agent Systems. EMAS 2014. Lecture Notes in Computer Science, Volume 8758.* (pp. 204–226). https://doi.org/10.1007/978-3-319-14484-

9_11

Davies, M., & Stone, T. (Eds.). (1995). *Folk Psychology: The Theory of Mind Debate*. Oxford, UK: Blackwell Publishers.

de Groot, A. D. (1965). *Thought and Choice in Chess* (1st ed.). The Hague, The Netherlands: Mouton Publishers.

Dennett, D. C. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.

Derdikman, D., & Moser, E. I. (2010). A manifold of spatial maps in the brain. *Trends in Cognitive Science*, *14*(12).

Descartes, R. (2015). *The passions of the soul and other late philosophical writings* (M. Moriarty, Trans.). Oxford, UK: Oxford University Press.

Devlin, K. J. (1991a). *Logic and Information*. Cambridge: Cambridge University Press.

Devlin, K. J. (1991b). Situations as Mathematical Abstractions. *Situation Theory and Its Applications Vol. 1*, 25–39.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. https://doi.org/10.1007/BF01386390

Domingos, P., & Lowd, D. (2009). Markov Logic: An interface layer for Artificial Intelligence. In T. Brachman, R. J.;Dietterich (Ed.), *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Seatle: Morgan & Claypool Publishers.

Domingos, P., & Richardson, M. (2007). Markov Logic: A Unifying Framework for Statistical Relational Learning. In B. Getoor, L.;Taskar (Ed.), *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Dooley, B. J. (2017). *Why AI with augmented and virtual reality will be the next big thing*. Upside, 4th April.

Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. MIT Press.

Dyson, G. (2012). *Turing's Cathedral*. New York, NY: Pantheon Books.

Eilam, D. (2014). Of mice and men: building blocks in cognitive mapping. *Neuroscience and Biobehavioral Reviews*, *47*, 393–409.

Emo, B. (2014). Real-World Wayfinding Experiments - Individual Preferences, Decisions and the Space Syntax Approach at Street Corners Beatrix. Retrieved from http://discovery.ucl.ac.uk/1452725/1/Emo_PhD.pdf.REDACTED.pdf

Endsley, M. R. (1988). Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *32*. https://doi.org/10.1177/154193128803200221

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

Endsley, M. R. (2000). THEORETICAL UNDERPINNINGS OF SITUATION AWARENESS : A CRITICAL REVIEW Process More Data ≠ More Information.

In M. R. Endsley & D. J. Garland (Eds.), *Most* (Vol. 301, pp. 3–32). Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:THEORETICAL+UNDERPINNINGS+OF+SITUATION+AWARENESS+:+A+CRITICAL+REVIEW#0

ExxonMobil. (2010). *OIMS: System 10-2 Emergency Preparedness and Response.* Retrieved from https://www.cnsopb.ns.ca/sites/default/files/inline/12450_so41877.1_spill_response_soei_0.pdf

Fan, X., McNeese, M., Sun, B., Hanratty, T., Allender, L., & Yen, J. (2010). Human–Agent Collaboration for Time-Stressed Multicontext Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *40*(2), 306–320. https://doi.org/10.1109/TSMCA.2009.2035302

Febbraro, A. Di, Giglio, D., & Sacco, N. (2016). A deterministic and stochastic Petri net model for traffic-responsive signaling control in urban areas. *IEEE Transactions on Intelligent Transportation Systems*, *17*(2), 510–524.

Filippidis, L., Galea, E. R., Lawrence, P., & Gwynne, S. (2001). Visibility Catchment Area of exits and signs. *InterFlam 2001: 9 Th International Fire Science & Engineering Conference*, 1529–1534. Retrieved from https://fseg.gre.ac.uk/fire/visibility_catchment_area.html

Gale, N., Golledge, R. G., Pellegrino, J. W., & Doherty, S. (1990). The acquisition and integration of route knowledge in an unfamiliar neighborhood. *Journal of Environmental Psychology*, *10*, 3–25.

Galea, E. R. (2003). Pedestrian and Evacuation Dynamics. *Proceedings of the 2nd International Conference on Pedestrian and Evacuation Dynamics, Greenwich, UK, 20-22 August 2003*. London, UK: CMS Press.

Galea, E. R., Xie, H., & Lawrence, P. (2014). Experimental and Survey Studies on the Effectiveness of Dynamic Signage Systems. *Fire Safety Science*, *11*, 1129–1143. https://doi.org/10.3801/IAFSS.FSS.11-1129

Gayathri, K. S., Easwarakumar, K. S., & Elias, S. (2017). Probabilistic ontology based activity recognition in smart homes using Markov Logic Network. *Knowledge-Based Systems*, *121*, 173–184. https://doi.org/10.1016/j.knosys.2017.01.025

Gayathri, K. S., Elias, S., & Shivashankar, S. (2014). *An Ontology and Pattern Clustering Approach for Activity Recognition in Smart Environments*. https://doi.org/10.1007/978-81-322-1771-8_72

Ghahramani, S. (2005). *Fundamentals of Probability with Stochastic Processes* (3rd ed.). Pearson Prentice Hall.

Gilmore, D., & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self (Ed.), *Artificial intelligence and human learning: intelligent computer-aided instruction*.

Goldschmidt, D., Manoonpong, P., & Dasgupta, S. (2017). A Neurocomputational Model of Goal-Directed Navigation in Insect-Inspired Artificial Agents.

Frontiers in Neurorobotics, *11*, 20. https://doi.org/10.3389/fnbot.2017.00020

Golledge, R. G. (1977). Multidimensional analysis and environmental behavior and design. In I. Altman & J. F. Wohlwill (Eds.), *Human behavior and environment: advances in theory and research*. Heidelberg: Springer Berlin Heidelberg.

Golledge, R. G. (1990). The Conceptual and Empirical Basis of a General Theory of Spatial Knowledge. In *Spatial Choices and Processes* (pp. 147–168). https://doi.org/10.1016/B978-0-444-88195-3.50014-3

Golledge, R. G. (1991). Cognition of physical and built environment. In T. Garling & G. W. Evans (Eds.), *Environment, Cognition, and Action: An Integrated Approach*. New York, NY: Oxford University Press.

Golledge, R. G. (1993). Geographical perspectives on spatial cognition. In T. Garling & R. G. Golledge (Eds.), *Behavior and Environment: Psychological and Geographical Approaches* (pp. 16–46). Amsterdam: Elsevier Science Publishers B. V.

Golledge, R. G. (1999a). Human wayfinding and cognitive maps. In R. G. Golledge (Ed.), *Wayfinding behavior* (p. 9). Baltimore: The Johns Hopkins University Press.

Golledge, R. G. (1999b). Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes. In *Psycoloquy* (Vol. 10). Baltimore: The John Hopkins University Press.

Gore, J., Flin, R., Stanton, N., & Wong, B. L. W. (2015). Applications for naturalistic decision-making. *Journal of Occupational and Organizational Psychology*, *88*(2), 223–230. https://doi.org/10.1111/joop.12121

Gorton, I. (1993). Parallel program design using high-level Petri nets. *Concurency and Computation: Practice and Experience*, *5*(2), 87–104.

Götze, J., & Boye, J. (2016). Learning landmark salience models from users' route instructions. *J. Locat. Based Serv.*, *10*(1), 47–63. https://doi.org/10.1080/17489725.2016.1172739

Grimmett, G. (2010). *Probability on Graphs: Random Processes on Graphs and Lattices*. Cambridge: Cambridge University Press.

Grush, R. (2000). Self, world and space: the meaning and mechanisms of ego- and allocentric spatial representation. *Brain and Mind*, *1*, 59–92.

Gruszka, A., Hampshire, A., & Owen, A. M. (2010). Learned Irrelevance Revisited: Pathology-Based Individual Differences, Normal Variation and Neural Correlates. In A. Gruszka, G. Matthews, & B. Szymura (Eds.), *Handbook of Individual Differences in Cognition: Attention, Memory, and Executive Control* (pp. 127–144). https://doi.org/10.1007/978-1-4419-1210-7_8

Hadzic, M., Wongthongtham, P., Dillon, T., & Chang, E. (2009). Current issues and the need for ontologies and agents. In *Ontology-Based Multi-Agent Systems* (pp. 1–14). Berlin, Heidelberg: Springer-Verlag.

Halpern, J. Y. (2003). *Reasoning about uncertainty*. Cambridge, MA: MIT Press.

Harman, G. (1976). Practical Reasoning. *The Review of Metaphysics*, *29*(3), 431–463. Retrieved from https://www.jstor.org/stable/20126812

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on System Science and Cybernetics*, *4*(2), 100–107.

Hassard, S. T. (2009). The variations of recognition primed decision-making and how it informs design decision-making. *NDM'09 Proceedings of the 9th Bi-Annual International Conference on Naturalistic Decision Making*, 57. Swindon, UK: BCS Learning & Development.

Hassler, S. (2016). Marvin Minsky and the pursuit of machine understanding - Making machines-and people-think [Spectral Lines]. *IEEE Spectrum*, *53*(3), 7–7. https://doi.org/10.1109/MSPEC.2016.7420381

Hayden, S. (2015). *NASA is creating a virtual reality mission to MARS, "The Mars 2030 experience."* Retrieved from https://www.roadtovr.com/nasa-creating-virtual-reality-mission-mars-mars-2030-experience/

Hayes-Roth, B. (1995). An architecture for adaptive intelligent systems. *Artificial Intelligence*, *72*(1–2), 329–365. https://doi.org/10.1016/0004-3702(94)00004-K

Heath, B. L., & Hill, R. R. (2008). The early history of agent-based modeling. *IIE Annual Conference.Proceedings*, 971–976. Retrieved from https://search.proquest.com/docview/192465049?accountid=12378

Heiner, M., & Gilbert, D. (2011). How might Petri nets enhance your systems biology toolkit. In L. M. Kristensen & L. Petrucci (Eds.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 6709 LNCS* (pp. 17–37). https://doi.org/10.1007/978-3-642-21834-7_2

Heiner, M., Herajy, M., Liu, F., Rohr, C., & Schwarick, M. (2012). Snoopy - A unifying Petri net tool. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7347 LNCS*, 398–407. https://doi.org/10.1007/978-3-642-31131-4_22

Heiner, M., Rohr, C., & Schwarick, M. (2013). MARCIE - Model checking and reachability analysis done efficiently. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7927 LNCS*, 389–399. https://doi.org/10.1007/978-3-642-38697-8_21

Hu, Y., Li, R., & Zhang, Y. (2018). Predicting pilot behavior during midair encounters using recognition primed decision model. *Information Sciences*, *422*, 377–395. https://doi.org/10.1016/j.ins.2017.09.035

Hutton, R. J. B., Warwick, W., Stanard, T., McDermott, P. L., & McIlwaine, S. (2001). Computational Model of Recognition-Primed Decisions (RPD): Improving Realism in Computer-Generated Forces (CGF). *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *45*(26), 1833–1837. https://doi.org/10.1177/154193120104502607

IMO. (2009). *SOLAS: Consolidated Edition* (5th ed.). London, UK: International Maritime Organization.

Isham, V. (1981). An Introduction to Spatial Point Processes and Markov Random Fields. *International Statistical Review / Revue Internationale de Statistique*, *49*(1), 21. https://doi.org/10.2307/1403035

Jain, D. (2011). Knowledge Engineering with Markov Logic Networks: A review. In G. Beierle, C.; Kern-Isberner (Ed.), *Proceedings of Evolving Knowledge in Theory and Applications. 3rd Workshop on Dynamics of Knowledge and Belief (DKB-2011) at the 34th Annual German Conference on Artificial Intelligence, KI-2011, vol. 361.* (pp. 16–30). Berlin, Germany: Fakultät für Mathematik und Informatik, FernUniversität in Hagen.

Jensen, K. (1981). Coloured Petri nets and the invariant-method. *Theoretical Computer Science*, *14*, 317–336. https://doi.org/https://doi.org/10.1016/0304-3975(81)90049-9

Jensen, K. (1996). *Colored Petri Nets* (3rd ed.). Springer.

Ji, Y., Massanari, R. M., Ager, J., Yen, J., Miller, R. E., & Ying, H. (2007). A fuzzy logic-based computational recognition-primed decision model. *Information Sciences*, *177*(20), 4338–4353. https://doi.org/10.1016/j.ins.2007.02.026

Johnson, A. W., Duda, K. R., Sheridan, T. B., & Oman, C. M. (2017). A Closed-Loop Model of Operator Visual Attention, Situation Awareness, and Performance Across Automation Mode Transitions. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *59*(2), 229–241. https://doi.org/10.1177/0018720816665759

Kabbaj, A. (2006). Development of Intelligent Systems and Multi-Agents Systems with Amine Platform. In S. H., P. Hitzler, & P. Øhrstrøm (Eds.), *Conceptual Structures: Inspiration and Application. ICCS 2006. Lecture Notes in Computer Science, Volume 4068*. Berlin, Heidelberg: Springer.

Kabbaj, A., Bouzouba, K., El Hachimi, K., & Ourdani, N. (2006). Ontologies in Amine Platform: Structures and Processes. In H. Schärfe, P. Hitzler, & P. Øhrstrøm (Eds.), *Conceptual Structures: Inspiration and Application. ICCS 2006. Lecture Notes in Computer Science* (pp. 300–313). https://doi.org/10.1007/11787181_22

Kabbaj, A., Bouzoubaa, K. M., & Soudi, A. (2005). Amine Platform: an Artificial Intelligence Environment For the Development of Intelligent Systems. *First Information and Communication Technologies International Symposium ICTIS*. Tetuan, Morocco.

Kahneman, D., Slovic, P., & Tversky, A. (Eds.). (1982). *Judgment under Uncertainty: Heuristics and Biases*. Cambridge, United Kingdom: Cambridge University Press.

Kang, S.-J., Kim, Y., & Kim, C.-H. (2010). Live path: adaptive agent navigation in the interactive virtual world. *The Visual Computer*, *26*, 467–476.

Kay, A. (1984). Computer software. *Scientific American*, *251*(3), 53–59.

Khan, B., Khan, F., Veitch, B., & Yang, M. (2018). An operational risk analysis tool to analyze marine transportation in Arctic waters. *Reliability Engineering & System Safety*, *169*, 485–502. https://doi.org/10.1016/j.ress.2017.09.014

Khan, S. A. (1989). Whether man is capable of free thinking. In *Freedom of Thought in Islam*. Karachi, Pakistan: Royal Book Company.

Kindermann, R., & Snell, J. L. (1980). Markov random fields and their applications. In *Science* (Vol. 1). https://doi.org/10.1109/TVCG.2009.208

Kingston, C., Nurse, J. R. C., Agrafiotis, I., & Milich, A. B. (2018). Using semantic clustering to support situation awareness on Twitter: the case of world views. *Human-Centric Computing and Information Sciences*, *8*(1), 22. https://doi.org/10.1186/s13673-018-0145-6

Klein, D., Marx, J., & Fischcach, K. (2018). Agent-Based Modeling in Social Science, History, and Philosophy. An Introduction. *Historical Social Research*, *43*(1).

Klein, G. (1998). *Sources of Power*. Cambridge, MA: MIT Press.

Klein, G. (2004). *The Power of Intuition*. Doubleday.

Klein, G. (2008). Naturalistic decision making. *Human Factors*, *50*(3), 456–460. https://doi.org/10.1518/001872008X288385

Kokar, M. M., Matheus, C. J., & Baclawski, K. (2009). Ontology-based situation awareness. *Information Fusion*, *10*(1), 83–98. https://doi.org/10.1016/j.inffus.2007.01.004

Kokar, M. M., Shin, S., Ulicny, B., & Moskal, J. (2014). Inferring relations and individuals relevant to a situation: An example. *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 18–194. https://doi.org/10.1109/CogSIMA.2014.6816561

Koutamanis, A. (1995). *Multilevel Analysis of Fire Escape Routes in a Virtual Environment*.

Kristiansen, S. (2005). *Maritime Transportation: Safety Management and Risk Analysis*. Amsterdam: Elsevier Butterworth-Heinemann.

Kumar, J. S., & Bhuvaneswari, P. (2012). Analysis of Electroencephalography (EEG) Signals and Its Categorization–A Study. *Procedia Engineering*, *38*, 2525–2536. https://doi.org/https://doi.org/10.1016/j.proeng.2012.06.298

Kyritsis, M., Gulliver, S. R., & Morar, S. (2014). Cognitive and environmental factors influencing the process of spatial knowledge acquisition within virtual reality environments. *International Journal of Artificial Life Research*, *4*(1).

Lee, S. A., Shusterman, A., & Spelke, E. S. (2006). Reorientation and Landmark-Guided Search by Young Children: Evidence for Two Systems. *Psychological Science*, *17:7*, 577–582.

Lehtonen, E., Sahlberg, H., Rovamo, E., & Summala, H. (2017). Learning game for training child bicyclists' situation awareness. *Accident Analysis and Prevention*, *105*, 72–83. https://doi.org/http://dx.doi.org/10.1016/j.aap.2016.07.036

Li, L., & Yokota, H. (2009). Application of Petri nets in bone remodeling. *Gene Regulation and System Biology*, *3*, 105–114.

Li, Y. (2013). *Modeling and Simulation of operator knowledge-based behavior [PhD Dissertation]* (University of Maryland). Retrieved from https://www.semanticscholar.org/paper/MODELING-AND-SIMULATION-OF-OPERATOR-KNOWLEDGE-BASED-Li/099a18923c1fd3cb9749783cc3ed0f13b28217e8

Liu, F., Deng, D., & Li, P. (2017). Dynamic Context-Aware Event Recognition Based on Markov Logic Networks. *Sensors*, *17*(3), 491. https://doi.org/10.3390/s17030491

Llinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E., & White, F. (2004). Revisiting the JDL Data Fusion Model II (2004). In P. Svensson & J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004), June 28-July 1, 2004.* Stockholm, Sweden.

Lucas, J. R. (1961). Minds, machines and Gödel. *Philosophy*, *36*(137), 112–127. Retrieved from http://www.jstor.org/stable/3749270

Luck, M., & Aylett, R. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, *14*(1), 3–32. https://doi.org/10.1080/088395100117142

Lynch, K. (1960). *The Image of the City*. Cambridge, MA: MIT Press.

MacEachren, A. M. (1992). Application of environmental learning theory to spatial knowledge acquisition from maps. *Annals of the Association of American Geographers*, *82*(2), 245–274.

Maciel, P. R. M., Trivedi, K. S., Matias, R., & Kim, D. S. (2011). Dependability Modelling. In *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions* (pp. 53–97). IGI Global.

Mackintosh, N. (1973). Stimulus selection: Learning to ignore stimuli that predict no change in reinforcement. In R. A. Hinde & J. S. Hinder (Eds.), *Constraints on Learning* (pp. 75–96). London: Academic Press.

Maes, P. (1995). Artificial life meets entertainment: lifelike autonomous agents. *Communications of the ACM*, *38*(11), 108–114. https://doi.org/10.1145/219717.219808

Malizia, A., Onorati, T., Diaz, P., Aedo, I., & Astorga-Paliza, F. (2010). SEMA4A: An ontology for emergency notification systems accessibility. *Expert Systems with Applications*, *37*(4), 3380–3391. https://doi.org/10.1016/j.eswa.2009.10.010

Man, L. (2015). An Agent-based Approach to Automated Merge 4D Arrival Trajectories in Busy Terminal Maneuvering Area. *Procedia Engineering*, *99*, 233–243. https://doi.org/10.1016/j.proeng.2014.12.531

Marwan, W., Rohr, C., & Heiner, M. (2012). Petri nets in snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks. In J. van Helden, A. Toussaint, & D. Thieffry (Eds.), *Methods in Molecular Biology* (Vol. 804, pp. 409–437). https://doi.org/10.1007/978-1-61779-361-5_21

McCarthy, J. (1979). *Ascribing mental qualities to machines (Memo No. 326)*. Palo Alto, CA.

McKinlay, R. (2016). Technology: Use or lose our navigation skills. *Nature*, *531*(7596), 573–575.

Means, B., Salas, E., Crandall, B., & Jacobs, T. O. (1993). Training Decision Makers for the Real World. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods1* (pp. 306–326). Norwood, NJ: Ablex Publishing Corporation.

Millington, I., & Funge, J. (2009). *Artificial Intelligence for Games* (2nd ed.). Burlington, MA: Morgan Kaufmann Publishers.

Minsky, M. (1988). *The society of mind*. New York, NY: Simon & Schuster.

Minsky, M. (1991). Conscious machines. *Machinery of Consciousness. Proceedings of the National Research Council of Canada, 75th Anniversary Symposium on Science in Society, June 1991.* NRC.

Mo, J. (2013). Performance modeling of communication networks with Markov chains. In J. Walrand (Ed.), *Syntheis Lectures on Communication Networks # 5*. Morgan & Claypool Publishers.

Moore, G. T., & Golledge, R. G. (Eds.). (1976). *Environmental knowing: Theories, research, and methods*. Stroudsburg, Pennsylvania: Dowden, Hutchinson & Ross, Inc.

Mosleh, A., & Chang, Y. H. (2004). Model-based human reliability analysis: Prospects and requirements. *Reliability Engineering and System Safety*, *83*(2), 241–253. https://doi.org/10.1016/j.ress.2003.09.014

Mueller, S. T. (2009). A Bayesian Recognitional Decision Model. *Journal of Cognitive Engineering and Decision Making*, *3*(2), 111–130. https://doi.org/10.1518/155534309X441871

Muppala, J. K., & Trivedi, K. S. (1990). *GSPN Models: Sensitivity analysis and applications. Proceedings of the 28th Annual Southeast Regional Conference, Greenville, South Caroline. April 18-20, 1990*. Greenville, South Carolina: Association for Computing Machinery.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, *77*(4), 541–580.

Musharraf, M., Khan, F., Veitch, B., MacKnnon, S., & Imtiaz, S. (2013). Human reliability assessment during offshore emergency conditions. *Safety Science*, *59*, 19–27.

Musharraf, M., Smith, J., Khan, F., & Veitch, B. (2018). Identifying route selection strategies in offshore emergency situations using decision trees. *Reliability Engineering & System Safety*. https://doi.org/10.1016/j.ress.2018.06.007

Musharraf, M., Smith, J., Khan, F., Veitch, B., & MacKinnon, S. (2018). Incorporating individual differences in human reliability analysis: An extension to the virtual experimental technique. *Safety Science*, *107*, 216–223. https://doi.org/10.1016/j.ssci.2017.07.010

Naderpour, M., Lu, J., & Zhang, G. (2014). An intelligent situation awareness support system for safety-critical environments. *Decision Support Systems*, *59*, 325–340. https://doi.org/10.1016/j.dss.2014.01.004

Nakanishi, H., Shimizu, S., & Isbister, K. (2005). Sensitizing social agents for virtual training. *Applied Artificial Intelligence*, *19*(3–4), 341–361. https://doi.org/10.1080/08839510590910192

Nasar, Z., & Jaffry, S. W. (2018). Trust-Based Situation Awareness: Comparative Analysis of Agent-Based and Population-Based Modeling. *Complexity*, *2018*, 1–17. https://doi.org/10.1155/2018/9540726

Norazahar, N., Khan, F., Veitch, B., & MacKinnon, S. (2016). Prioritizing safety critical human and organizational factors of EER systems of offshore installations in a harsh environment. *Safety Science*.

Norazahar, N., Smith, J., Khan, F., & Veitch, B. (2018). The use of a virtual environment in managing risks associated with human responses in emergency situations on offshore installations. *Ocean Engineering*, *147*, 621–628. https://doi.org/10.1016/j.oceaneng.2017.09.044

Norling, E. (2004). Folk psychology for human modelling: Extending the BDI paradigm. *AAMAS '04 Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems*. New York, NY: IEEE Computer Society, Washington DC.

Norling, E., Sonenberg, L., & Rönnquist, R. (2000). Enhancing Multi-Agent Based Simulation with Human-Like Decision Making Strategies. In S. Moss & P. Davidsson (Eds.), *Multi-Agent-Based Simulation. MABS 2000. Lecture Notes in Computer Science, Volume 1979*. Berling, Heidelberg: Springer.

Nowroozi, A., Shiri, M. E., Aslanian, A., & Lucas, C. (2012). A general computational recognition primed decision model with multi-agent rescue simulation benchmark. *Information Sciences*, *187*, 52–71. https://doi.org/10.1016/j.ins.2011.09.039

Nwana, H. S. (1996). Software agents: an overview. *The Knowledge Engineering Review*, *11*(3), 205–244. https://doi.org/10.1017/S026988890000789X

Nwiabu, N., Allison, I., Holt, P., Lowit, P., & Oyeneyin, B. (2012). Case-based situation awareness. *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support. 6-8 March 2012*, 22–29. https://doi.org/10.1109/CogSIMA.2012.6188388

Nys, M., Gyselinck, V., Orriols, E., & Hickmann, M. (2015). Landmark and route knowledge in children's spatial representation of a virtual environment. *Frontiers in Psychology*, *5*, 1–15. https://doi.org/10.3389/fpsyg.2014.01522

Orasanu, J., & Connolly, T. (1993). The reinvention of decision making. In G. Klein,

J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods* (pp. 3–20). Norwood, NJ: Ablex Publishing Corporation.

OSHA. (2003). *Emergency-exit-routes-factsheet*. Retrieved from https://www.osha.gov/OshDoc/data_General_Facts/emergency-exit-routes-factsheet.pdf

OSHA. (2018). Emergency exit routes factsheet. Retrieved April 22, 2019, from https://www.osha.gov/OshDoc/data_General_Facts/emergency-exit-routes-factsheet.pdf

Parsons, S., & Wooldridge, M. (2002). Game Theory and Decision Theory in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, *5*(3), 243–254. https://doi.org/10.1023/A:1015575522401

Passini, R. (1977). *Wayfinding : a study of spatial problem solving with implications for physical design*. Pennsylvania State University, Pennsylvania, USA.

Patterson, R., Fournier, L., Pierce, B., Winterbottom, M., & Tripp, L. (2009). Modeling the dynamics of recognition-primed decision making. *Proceedings of NDM9, the 9th International Conference on Naturalistic Decision Making*. London, UK: British Computer Society.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inferences*. San Mateo, CA: Morgan Kaufmann.

Penrose, R. (1989). *The emperor's new mind: Concerning computers, minds, and the laws of Physics*. Oxford University Press.

Penrose, R. (1991). The emperor's new mind. *RSA Journal*, *139*(5420), 506–514. Retrieved from http://www.jstor.org/stable/41378098

Peterson, J. L. (1977). Petri nets. *ACM Computing Surveys (CSUR)*, *9*(3), 223–252.

Petri, C. A. (1966). *Kommunikation mit Automaten. Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2; (English translation)* (Vol. 1). Vol. 1. New York.

Plank, M., Snider, J., Kaestner, E., Halgren, E., & Poizner, H. (2014). Neurocognitive stages of spatial cognitive mapping measured during free exploration of a large-scale virtual environment. *Journal of Neurophysiology*, *113*, 740–753.

Poon, H., & Domingos, P. (2006). Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, 458–463. Retrieved from http://dl.acm.org/citation.cfm?id=1597538.1597612

Posner, M. I., Nissen, M. J., & Klein, R. M. (1976). Visual dominance: An information-processing account of its origins and significance. *Psychological Review*, *83*(2), 157–171. https://doi.org/10.1037/0033-295X.83.2.157

Press, G. (2018, February 7). *The Brute Force of IBM Deep Blue and Google DeepMind*. Retrieved from https://www.forbes.com/sites/gilpress/2018/02/07/the-brute-force-of-deep-blue-and-deep-learning/#3729cb5c49e3

Preston, C. J. (1974). *Gibbs States on Countable Sets*. Cambridge: Cambridge University Press.

Proulx, G. (2007). Response to fire alarms. *Fire Protection Engineering*, *33*, 8–14. Retrieved from http://www.cfaa.ca/Files/flash/CODES/FIRE ALARM ARTICLES FOR THE AHJ/Fire Alarm Response.pdf

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

Ramchurn, S. D., Huynh, T. D., Ikuno, Y., Flann, J., Wu, F., Moreau, L., … Roberts, S. J. (2016). A disaster response system based on human-agent collectives. *Journal of Artificial Intelligence Research*, *57*, 661–708.

Rao, A. S., & Georgeff, M. P. (1995). BDI Agents: From theory to practice. In V. Lesser & L. Gasser (Eds.), *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* (pp. 312–319). Retrieved from 0262621029

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Récopé, M., Fache, H., Beaujouan, J., Coutarel, F., & Rix-Lièvre, G. (2019). A study of the individual activity of professional volleyball players: Situation assessment and sensemaking under time pressure. *Applied Ergonomics*, *80*, 226–237. https://doi.org/10.1016/j.apergo.2018.07.003

Resnick, M. (2001). Recognition Primed decision making in E-Commerce. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Santa Monica: Sage Journals Ltd.

Ross, S. A. (1973). The economic theory of agency: the principal's problem. *American Economic Review*, *63*(2), 134–139.

Russell, S. J., & Subramanian, D. (1995). Provably Bounded-Optimal Agents. *Journal of Artificial Intelligence Research*, *2*, 575–609. https://doi.org/10.1613/jair.133

Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Upper Saddle River, NJ, USA: Addison-Wesley.

Selfridge, O. G. (1988). Pandemonium: a paradigm for learning. In *Neurocomputing: foundations of research* (pp. 115–122). Cambridge, MA: MIT Press.

Shanton, K., & Goldman, A. (2010). Simulation theory. *Wiley Interdisciplinary Reviews: Cognitive Science*, *1*(4), 527–538. https://doi.org/10.1002/wcs.33

Sharma, G., Kaushal, Y., Chandra, S., Singh, V., Mittal, A. P., & Dutt, V. (2017). Influence of Landmarks on Wayfinding and Brain Connectivity in Immersive Virtual Reality Environment. *Frontiers in Psychology*, *8*, 1220. https://doi.org/10.3389/fpsyg.2017.01220

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, *60*(1), 51–92. https://doi.org/10.1016/0004-3702(93)90034-9

Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-*

Theoretic, and Logical Foundations. Cambridge, MA: Cambridge University Press.

Singla, P., & Domingos, P. (2005). Discriminative Training of Markov Logic Networks. *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, 868–873. Retrieved from http://dl.acm.org/citation.cfm?id=1619410.1619472

Sinnett, S., Spence, C., & Soto-Faraco, S. (2007). Visual dominance and attention: The Colavita effect revisited. *Perception & Psychophysics*, *69*(5), 673–686. https://doi.org/10.3758/BF03193770

Smith, B. M. (2007). Working minds. In *Phi Delta Kappan* (Vol. 88). https://doi.org/10.1177/003172170708801001

Smith, D. C., Cypher, A., & Spohrer, J. (1994). KidSim: programming agents without a programming language. *Communications of the ACM*, *37*(7), 54–67. https://doi.org/10.1145/176789.176795

Smith, J. (2015). *The effect of virtual environment training on participant competence and learning in offshore emergency egress scenarios [Master thesis]* (Memorial University of Newfoundland). Retrieved from https://research.library.mun.ca/8401/1/thesis.pdf

Smoking and Health: Joint Report of the Study Group on Smoking and Health. (1957). *Science*, *125*(3258), 1129–1133. https://doi.org/10.1126/science.125.3258.1129

Sneddon, A., Mearns, K., & Flin, R. (2013). Stress, fatigue, situation awareness and safety in offshore drilling crews. *Safety Science*, *56*, 80–88. https://doi.org/10.1016/j.ssci.2012.05.027

Snidaro, L., Visentini, I., & Bryan, K. (2015). Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks. *Information Fusion*, *21*, 159–172. https://doi.org/10.1016/j.inffus.2013.03.004

Snidaro, L., Visentini, I., Bryan, K., & Foresti, G. L. (2012). Markov Logic Networks for context integration and situation assessment in maritime domain. *2012 15th International Conference on Information Fusion*, 1534–1539.

Sokolowski, J. A. (2003). Enhanced Decision Modeling Using Multiagent System Simulation. *SIMULATION*, *79*(4), 232–242. https://doi.org/10.1177/0037549703038886

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks/Cole Thomson Learning.

Spouge, J. (1999). *A Guide to Quantitative Risk Assessment for Offshore Installations*. Aberdeen, UK: CMPT Publication.

Sud, A., Andersen, E., Curtis, S., Lin, M., & Manocha, D. (2007). Real-time path planning for virtual agents in dynamic environments. *Proceedings of IEEE*

Virtual Reality, 91–98. https://doi.org/10.1109/VR.2007.352468

Szczerbak, M., Bouabdallah, A., Toutain, F., & Bonnin, J.-M. (2013). A Model to Compare and Manipulate Situations Represented as Semantically Labeled Graphs. In H. D. Pfeiffer, D. I. Ignatov, J. Poelmans, & N. Gadiraju (Eds.), *Conceptual Structures for STEM Research and Education* (pp. 44–57). Berlin, Heidelberg: Springer Berlin Heidelberg.

Thilakarathne, D. J. (2015). Modelling of situation awareness with perception, attention, and prior and retrospective awareness. *Biologically Inspired Cognitive Architectures*, *12*, 77–104. https://doi.org/10.1016/j.bica.2015.04.010

Tolman, E. C. (1948). Cognitive maps in rats and men. *The Psychological Review*, *55*(4).

Tong, D., & Canter, D. (1985). The decision to evacuate: a study of the motivations which contribute to evacuation in the event of fire. *Fire Safety Journal*, *9*(3), 257–265. https://doi.org/10.1016/0379-7112(85)90036-0

Trivedi, K. S. (2002). *Probability and Statistics with Reliability, Queuing, and Computer Science Applications* (2nd ed.). Wiley Interscience Publication.

Tuan, Y.-F. (1977). *Space and place: The perspective of experience*. Minneapolis: University of Minnesota Press.

Turing, A. M. (1947). *Lecture to the L.M.S. [London Mathematical Society] Feb. 20, 1947*. London, UK: This record is held by Cambridge University: King's College Archive Centre.

Turing, A. M. (1950). I.—Computing machinery and intelligence. *Mind*, *LIX*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

Tutolo, D. (1979). Attention: Necessary Aspect of Listening. *Language Arts*, *56*(1), 34–37. Retrieved from http://www.jstor.org/stable/41404756

Tversky, A., & Kahneman, D. (1974). Judgment under Uncertainty: Heuristics and Biases. *Science*, *185*(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

University of Greenwich, F. S. E. G. (2017). *EXODUS capabilities*. Retrieved from https://fseg.gre.ac.uk/exodus/exodus_products.html

Vandenberg, A. E. (2016). Human Wayfinding: Integration of Mind and Body. In R. H. Hunter, L. A. Anderson, & B. L. Belza (Eds.), *Community Wayfinding: Pathways to Understanding* (pp. 17–32). https://doi.org/10.1007/978-3-319-31072-5_2

von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press.

Waller, D., & Lippa, Y. (2007). Landmarks as beacons and associative cues: Their role in route learning. *Memory & Cognition*, *35*(5), 910–924. https://doi.org/10.3758/BF03193465

Wang, R. F., & Spelke, E. S. (2002). Human spatial representation: insights from animals. *Trends in Cognitive Sciences*, *6*, 376–382.

Wankhede, A. (2017, September). Different Types of Alarms on Ships. *Marine Insight*. Retrieved from https://www.marineinsight.com/marine-safety/different-types-of-alarms-on-ship/

Weinspach, P. M., Gundlach, J., Klingelhofer, H. G., Ries, R., & Schneider, U. (1997). Analysis of the Fire on April 11th, 1996; Recommendations and Consequences for Dusseldorf Rhein-Ruhr-Airport. *Staatskanzlei Nordrhein-Wstfalen, Mannesmannufer*, *1*.

Winerman, L. (2004). Fighting fire with psychology. *Monitor on Pscyhology*, *35*(8), 28. Retrieved from https://www.apa.org/monitor/sep04/fighting

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. West Sussex, United Kingdom: John Wiley & Sons Ltd.

Wooldridge, M., & Jennings, N. R. (1995). Agent theories, architectures and languages: a survey. In M. J. Wooldridge & N. R. Jennings (Eds.), *Intelligent Agents*. https://doi.org/10.1007/3-540-58855-8

Xie, H., Filippidis, L., Galea, E. R., Blackshields, D., & Lawrence, P. J. (2012). Experimental analysis of the effectiveness of emergency signage and its implementation in evacuation simulation. *Fire and Materials*, *36*(5–6), 367–382. https://doi.org/10.1002/fam.1095

Xu, G., Cao, Y., Ren, Y., Li, X., & Feng, Z. (2017). Network Security Situation Awareness Based on Semantic Ontology and User-Defined Rules for Internet of Things. *IEEE Access*, *5*, 21046–21056. https://doi.org/10.1109/ACCESS.2017.2734681

Yang, C., Wang, D., Zeng, Y., Yue, Y., & Siritanawan, P. (2019). Knowledge-based multimodal information fusion for role recognition and situation assessment by using mobile robot. *Information Fusion*, *50*, 126–138. https://doi.org/10.1016/j.inffus.2018.10.007

Yurkiewicz, I. R., & Tsao, J. W. (2012). Book Review: Why people get lost: the psychology and neuroscience of spatial cognition by Paul A. Dudchenko. *Journal of the Neurological Sciences*, *313*(1–2), 197–198.

Zsambok, C. E. (1997). Naturalistic decision making: where are we now? In C. E. Zsambok & G. Klein (Eds.), *Naturalistic decision making* (pp. 3–16). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Zsambok, C. E., & Klein, G. A. (Eds.). (1997). *Naturalistic Decision Making*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Appendix A

The GSPNRL model

GSPN stands for generalized stochastic Petri net. The RL stands for route learning and hence the acronym GSPNRL is a model of route learning that exploits the stochastic Petri nets for representation of the phenomenon a human being undergoes when there is a need to learn a new route in an environment.

The model has thirty places, ten stochastic transition, and twenty-three immediate transitions. The stochastic transitions are depicted in Figure 3.2 as simple rectangles, and solid rectangles show the immediate transitions. Circles show the places. A circle with one dot shows a single token on the place, and a circle with a number inside shows the number of tokens that are present in that place. The place A_{22} is of integer type. The type integer is referred to as `NUM` in the model. The place A_{21} and A_{23} are of type `NUMLEVELS`, where `NUMLEVELS` is a product type in which the first member is `NUM` and the second member is an enumerated datatype `D = {LOW, LOWEST, MEDIUM, HIGH, HIGHEST}`, which represents the difficulty levels associated with a landmark. The remaining places do not associate any type and, therefore, represent only simple tokens in the model.

The GSPNRL model is 1-bounded. Thus, if any of the transitions from t_{28}-t_{32} are enabled, the others will be disabled and cannot execute allowing only one landmark to be processed at a single time. This resembles a real-life situation where people avoid getting confused dealing with more than one landmarks at a time, rather every landmark is processed in a sequential way. The net N3 is the only net in the model that uses colored tokens — tokens with custom datatypes such as the type `NUMLEVELS`. Colored Petri-net allows the development of compact and parameterized models, which otherwise require a difficult to read and understand, and lengthy models.

The net N2 is the main component of the model. It integrates the information coming from N1 and N3 by using a semi-Markov process (Bause & Kritzinger, 1996) such that the firing rates of the stochastic transitions are kept higher for inputs with higher difficulty level landmarks. Table 3.1 describes the range of stochastic transition rates assigned to the GSPNRL model. The rate ranges in Table 3.1 are selected, so that: (i) at the lowest difficulty possible, the rate of forgetting is at the minimum, (ii) at the highest difficulty possible, the rate of remembering is at the minimum. The boundaries of the rate ranges are defined to produce results close to the empirical values. The stochastic transitions t_8, t_{10}, t_{12}, t_{14}, and t_{16} can be assigned randomly picked values from the ranges defined in Table 3.1. A particular assignment of stochastic transition rates is dependent on the application. If a rate λ is to be used, the average time to fire (execute) will be $1/\lambda$, because the model uses exponentially distributed firing delays. The distribution of firing time of transition t_i is given by the rule:

$$F(x) = 1 - e^{-\lambda_i x}.$$

Transitions t_8 and t_9 are conflicting transitions: if t_8 fires, then t_9 will become disabled and vice versa. Firing of t_8 means that the model will not retain the navigation command, whereas firing of t_9 will save the navigation command along with the landmark information. The pairs of stochastic transitions (t_8, t_9), (t_{10}, t_{11}), (t_{12}, t_{13}), (t_{14}, t_{15}), and (t_{16}, t_{17}) are developed so that the first transition in each pair, if fired, is responsible for showing the behavior of forgetting, say by not saving any of its input data. The second transition in each pair shows the remembering behavior by saving its inputs into the memory of an agent that uses the GSPNRL model.

Appendix B

B.1 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 Figure B.1.

B.2 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 (Klein, 1998). In RPDM, a decision-maker should pick one option and evaluate if this will work in the current scenario.

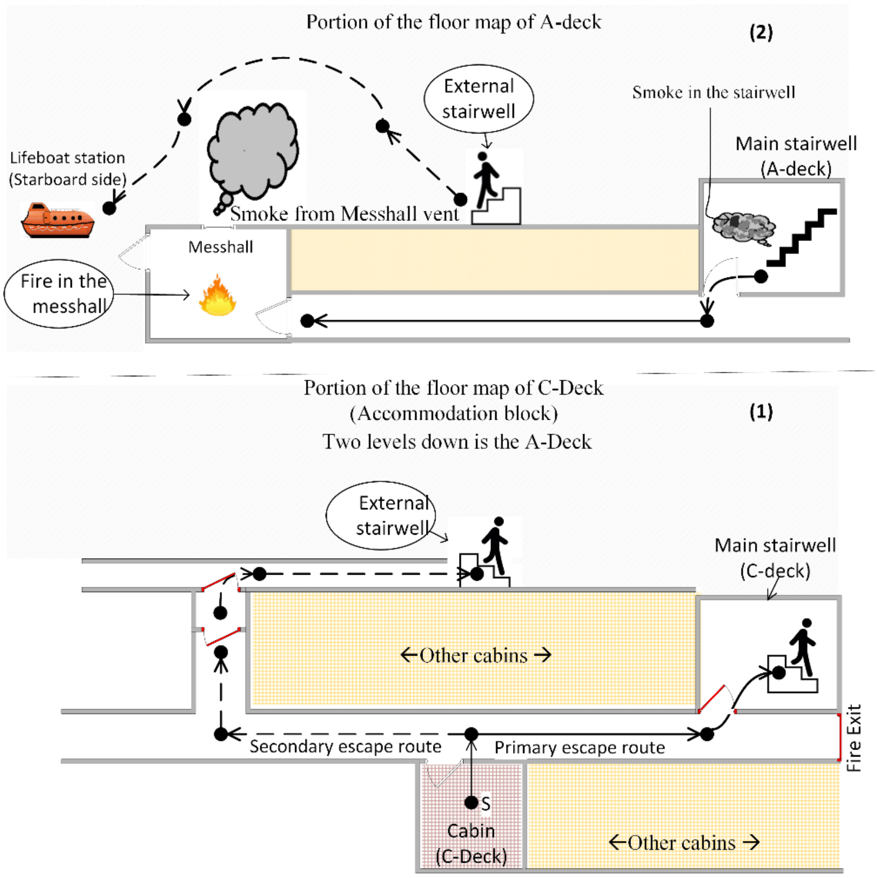


Figure B.1. A portion of the floor map of A-deck, and C-deck (accommodation block). The primary escape route is shown with arrows pointing towards the main stairwell from the cabin. The participant has to go down two levels from the cabin to A-deck, where the muster stations are located. Hazards at different locations are shown to illustrate an emergency.

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 (Klein, 1998, 2004) for details about these variations. Figure 5.1 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" (Klein, 1998, pp. 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.

Appendix C

Assumptions for knowledge elicitation

Because each participant has performed the scenarios as explained in Section 4.2, the following assumptions seems reasonable about the participants of Smith's (2015) 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 D

D.1 An implementation of Algorithm 5.1 in OO2APL based BDI agent.

Let p1, p2 are probabilities for two competing situations (`FIRE` & `EVACUATE`) during time interval *T*.

Let p3, 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 p4, 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`.

```
if(p1>=alpha1&&p2<=alpha2)
        exp.arrDecision.add(DecisionalVal.MoveToPMS);
else if(p2>=alpha1&&p1<=alpha2)
    exp.arrDecision.add(DecisionalVal.MoveToSMS);
else{
        if(p3>=alpha1&&p4<=alpha2)
            exp.arrDecision.add(DecisionalVal.MoveToPMS);
        else if(p4>=alpha1)&&(p3<=alpha2)
            exp.arrDecision.add(DecisionalVal.MoveToSMS);
    else
            exp.arrDecision.add(DecisionalVal.Diagnose);
    }
```

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 cannot 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.

D.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 E

Actions evaluation and modifications

Let $x[1], x[2], \ldots, x[n]$ 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]) //a goal, move to x[n] from the current position
      FollowPath                  //subgoal of MoveTo
         TraverseEdge(x[1])    //x[1] is 1st waypoint, a subgoal of FollowPath
         TraverseEdge(x[2])    //x[2] is the 2nd waypoint along the route.
         ⋮
         TraverseEdge(x[i])    //move down to the ith waypoint
         AssessEdgeForBlockage  //subgoal of TraverseEdge; an edge is in the
                                //navigation graph
         if(route is blocked at x[i])
            1. "Estimate how many waypoints beyond x[i] (including x[i]) be
               avoided let this be k."
            2. "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]) //a steering behavior, see (Buckland, 2004).
            else
               Seek(x[i])   //a steering behavior.
         ⋮
         TraverseEdge(w[n])
         stop
```