*Safety and Reliability of Complex Engineered Systems – Podofillini et al. (Eds)*


# Accident analysis by logic programming technique

Z.I. Awal & K. Hasegawa
*Department of Naval Architecture & Ocean Engineering, Graduate School of Engineering, Osaka University, Osaka, Japan*

ABSTRACT: This paper proposes a new accident analysis method based on Logic Programming Technique (LPT). In this study, an accident is considered as an unintentional event that occurs due to one or more root cause(s) that can be logically linked to the accident event. It is discussed in this research that various root causes of an accident can be identified using different accident theories and can be formed into logic predicates. These logic predicates may describe human perception, action and/or any natural phenomena that can be utilized in logic programming domain (e.g. Prolog) and result in simulation of different accident scenarios. Such a model, however, fundamentally conducts a simulation of logical description of the world. This description exposes the accident causes and the occurrence of events leading to the accident. The study reveals that LPT has the ability to find out the hidden accident causes for different scenarios. The advantages of logic programming over other techniques for accident analysis are also exhibited. Some recent maritime accidents are utilized to explain the technique. The future prospects and challenges of this technique are discussed briefly.

## 1 INTRODUCTION

Accidents may take place almost anywhere and anytime, for example—underwater accidents of submarine, ship collision on the water surface, air plane accidents in the sky, road accidents on the land, mining accidents under the ground, or accidents in space. Most accidents remain unpredictable until the accident itself becomes unavoidable.

The domain of accident analysis is comparatively young considering other disciplines of science and engineering. During the past one hundred years or so researchers have become interested in accident modelling and it is fundamentally because of the enormous loss of lives and resources due to accidents. The rapid changes in the society have complicated the societal interactions among 'man and machine' and accidents are now developing from these complex interactions.

Interestingly, the discipline of accident analysis has been enriched by the contributions of the researchers from diverse disciplines like industrial engineering, public health studies, control engineering and so on. This fact, however, points to another reality that in order to solve the accident problem, it is essential to have a wide range of knowledge about the system of which accident analysis is performed. It is, therefore, essential to define what accident is (as a basis) and what the scope of this study deals with. Therefore, the following two sections discuss these issues briefly.

### 1.1 *Accident, accident analysis and accident model*

One of the earliest definition of accident was given by Heierich in 1931 and it has been referenced by Ward (2012). The definition is "An accident is an unplanned and uncontrolled event in which the action or reaction of an object, substance, person, or radiation results in personal injury or the probability thereof". However, one may derive a simpler definition out of it—an accident is an unforeseen and unplanned event or circumstance that causes damage and/or injury.

According to Stringfellow (2010) accident analysis is the process by which the reasons for the occurrence of an accident are uncovered. Information and lessons learnt from accident analysis are used to re-engineer the same or other systems so that future accidents (which may or may not be the form) do not occur.

Typically, an accident model provides a conceptualization of the characteristics of the accident that normally shows the relation between causes and effects (Qureshi 2008). Since, an accident event is the result of some cause or causes, therefore, the challenge for accident analyst is to identify the rela-

Table 1. Complexity in sociotechnical systems (Perrow 1984, Qureshi 2008).

| Problem | Example/Description |
|---|---|
| Large problem space | The problem space contains multiple objectives and constraints. The domain of knowledge for solving problems is large and diversified. Therefore, the problem space becomes large. |
| Heterogeneous perspective | The actors in a socio technical system have heterogeneous perspective, which makes solving problems more complex. |
| Dynamic | The systems are always changing its state with respect to time. |
| Tight coupling | The components of sociotechnical systems are often tightly coupled and room for margin of error is very little. |
| Complex interaction | The interaction of an actor with the system may reveal hidden effects, which cause accident. |
| Disturbances | Sociotechnical systems have to accept disturbance from variety of sources. |

tionship between these causes and effects within the system.

An accident model or accident theory provides a hypothesis of accident causation and attempts to validate the hypothesis. In order to validate the theories, there are several tools for accident analysis that essentially does not propose any hypothesis rather provide theoretical instruments for analysing accidents. For example, Fault Tree Analysis (FTA) (Vesely et al. 1981), AcciMap (Rasmussen & Svedung 2000), and Coloured Petri nets (Vernez et al. 2003) and others can be used to analyse accidents.

### 1.2 *Accident modelling problem*

The problem of identifying accident causes is quite complex and diverse. Most of the modern day accident models adopt the fact that accident takes place in a complex sociotechnical system in order to combine the social and technical attributes in the analysis (Qureshi 2008 and Khanzode et al. 2012). Some key attributes of sociotechnical system are large problem space, heterogeneous perspective, tight coupling, complex interaction, and others. Table 1 elaborates these the key complexities.

In order to deal with the problems mentioned above, this study conducted a literature review on the major accident models. The following section discusses the literature review briefly.

## 2 LITERATURE REVIEW

The literature review on accident modelling suggest that different branches of knowledge (such as ergonomics and human factors, organization theory, industrial psychology, medicine, environmental sciences, law etc.) are interrelated with accident causation. From the accident causation perspective, these fields are overlapping and originate complexities. Therefore, the classifications of accident analysis techniques vary widely. Khanzode et al. (2012) and Qureshi (2008) reviewed accident/injury theories and made respective classifications. For example, Khanzode et al. (2012) classified the accident theories according to chronology of generation:

– 1st Generation: Accident proneness based
– 2nd Generation: Domino theory based
– 3rd Generation: Injury epidemiology based
– 4th Generation: System based

The study by Qureshi (2007) reveals another type of classification of accident models. Such as:

– Traditional approaches to accident modelling (sequential models)
– Epidemiological/Organizational models of accident causation
– Systemic accident models

Based on the literature review a chronological order of major accident models can be constructed which is shown in Figure 1.

A comparison between Khanzode et al. (2012) and Qureshi (2008) suggest that both classification are similar in a sense that Khanzode's 1st and 2nd generation of accidents can be considered as sequential accident models by Qureshi (2008). The 3rd and 4th generation accident models can be considered as epidemiological and systemic model of accidents respectively by Qureshi (2008).

Nevertheless, it is important to comprehend that the development of accident theories can be related to the change in sociotechnical context over the years. It is evident in Figure 1 that in recent times more system theoretic models are proposed compared to the earlier times when sequential models were proposed. The rapid industrialization, change in interaction between men and machine is giving birth to new types of accidents. Therefore, new generation of accident analysis techniques are fundamentally essential.

The literature review also suggest that traditional approach towards accident analysis is using statistical tools (for example Awal 2007, Awal et al. 2010). So far very little computational techniques have been developed that can efficiently analyse accident in an established programming domain. Recent studies by Awal & Hasegawa (2014a, b)

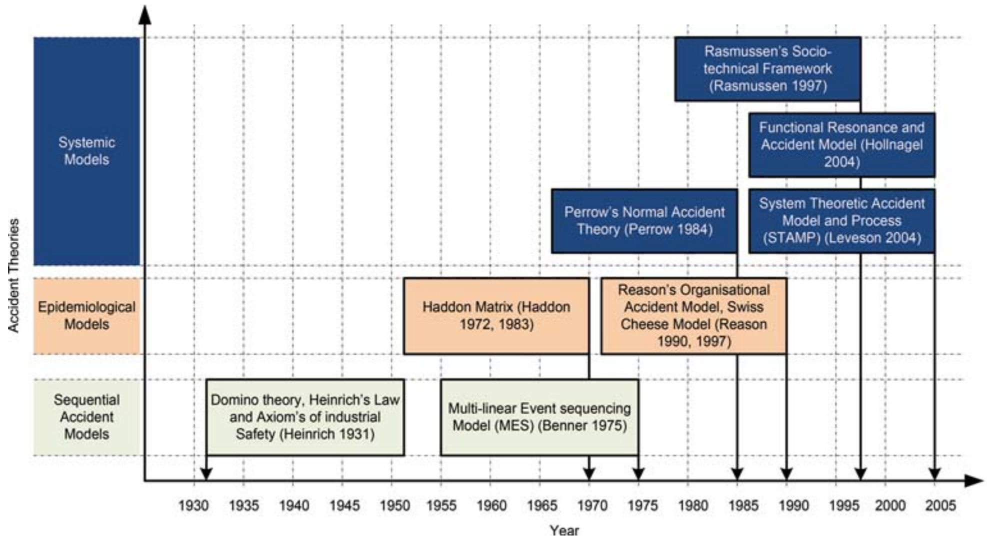


Figure 1. Major accident theories and their chronological appearance.

and Hasegawa & Awal (2013) show the need and progress of such an approach. These research works reveal the potentials of utilizing logic programming technique in accident analysis. This research work is an extension of the author's ongoing research work. In this study, this paper attempts to highlight the following:

– Briefly describes the utilization of computational technique for accident analysis using intelligent agents.
– Incorporate or unify the analysis of human action/perception, natural phenomena, engineering and technological aspects in the same programming platform.

The following section describes the methodology for logic programming technique briefly.

## 3 METHODOLOGY

### 3.1 *Hypothesis*

The hypothesis adopted in this study is that Logic Programming Technique (LPT) can be used to analyse and deduce the perception/action of human agents using deductive logic along with simulation of the concerned system in order to find out the unknown causes of a particular type of accident.

### 3.2 *Definition of logic*

Logic may be defined as the science of reasoning (Hardegree 1999). Reasoning is a special mental activity called inferring, what can also be called making (or performing) inferences. A useful and simple definition of the word 'infer' may be given as 'To infer is to draw conclusions from premises' (Hardegree 1999). In order to simplify the understanding of reasoning, logic treats both premises and conclusions in a single term called 'statements'. Logic correspondingly treats inferences in terms of collections of statements, which are also called 'arguments'. The definition of 'argument' that is relevant to logic is given as—'an argument is a collection of statements, one of which is designated as the conclusion, and the remainder of which are designated as the premises' (Hardegree 1999). Therefore, the reasoning process may be thought of as beginning with input (premises, data, etc.) and producing output (conclusions).

### 3.3 *Example of usage*

An example of the usage of deductive logic in the perception-action of ship crew is given in Figure 2. Based on the accident of MV Costa Concordia in 2012 (Ministry of Infrastructures and Transports, 2013) an analysis is conducted by Lieto (2012) which reveals the errors made by the crew of the ship prior to the accident. According to Lieto (2012) this accident could be an organizational accident where the root causes of the accident demonstrate the characteristics of Swiss cheese model (Reason 1990, 1997).

Based on the analysis by Lieto (2012) the first error made by the ship crew was the change in

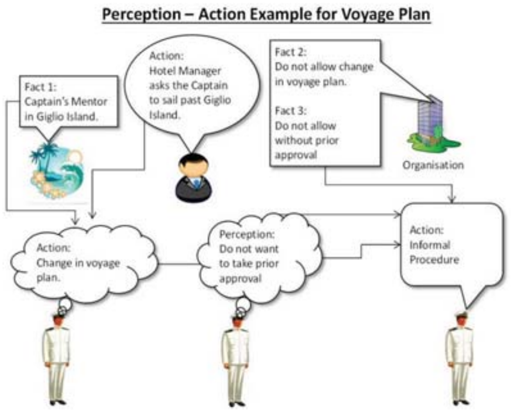


Figure 2. Action perception sequence of ship crew for the 1st error in the Costa Concordia accident case.

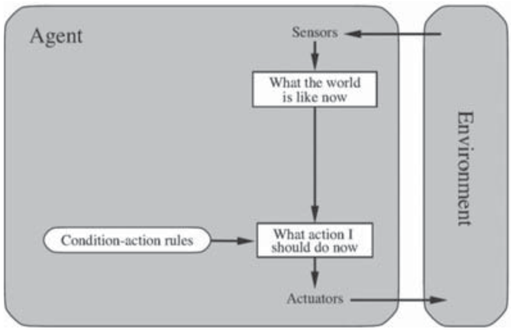


Figure 3. A schematic diagram of simple reflex agent (Russel & Norvig 2010).

voyage plan. The way the decision was made to change the voyage plan is shown in Figure 2. Under the circumstances the decision made at that instance did not appear harmful rather a logical choice consistent with the sociotechnical scenario. It is interesting to note that the decisions, which are considered as errors, taken by the crew at different instants seemed logically correct in the context. Yet the accident took place and appeared to match with Swiss cheese analogy of accident causation. Therefore, in this research it is discussed that logical deductions of crew decisions in terms of perception and action may help to identify accident causes and the accident itself.

### 3.4 *Agent: Definition and types*

An agent can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators (Russel & Norvig 2010). For example, a software agent receives keystrokes, file contents and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets. In general, for an agent, choice of action at any given instant may depend on the entire percept sequence observed to date but not on anything that it has not perceived. Mathematically, an agent's behaviour is described by the agent function that maps and given percept sequence to an action.

According to Russel & Norvig (2010) there are several types of agents with different characteristics:

– Simple reflex agent
– Model-based reflex agent
– Goal-based agent
– Utility-based agent
– Learning agent

In this study, simple reflex agents are considered for discussing the logic programming technique.

### 3.5 *Design of Simple Reflex agent*

The characteristic of a simple reflex agent is that such an agent selects action(s) based on the current percept, ignoring the rest of the percept history. The agent uses the condition-action rule or situation-action rule. The simple reflex agent needs to have a library of rules so that if a certain situation should arise and it is in the set of condition-action rules the agent will know how to react with minimal reasoning. A schematic diagram of simple reflex agent is shown in Figure 3. An example of simple reflex agent could be the reaction of a person to fire. A person pulls his or her hand away without thinking about any possibility that there could be danger in the path of his/her arm. This is called reflex action. Similar to a person's reaction to fire, a simple reflex agent behaves relative to the situation and does not consider previous percept.

An initial yet most significant step for agent design is to specify the task environment as fully as possible. Task environments are essentially the 'problems' to which the rational agents are the 'solutions' (Russel & Norvig 2010). It is a general practice to define or describe PEAS (Performance, Environment, Actuators and Sensors) as fully as possible for designing agents. In this study, several agents are considered based on the maritime context. Table 2 depicts a description of the agents in terms of PEAS. In this table, six simple reflex agents are considered, as an example, including the ship itself and five ship crewmembers, such as a Captain, a Senior Officer of the Watch (SOOW), a Junior Officer of the Watch (JOOW), a Helms-

Table 2. Example of PEAS definition of different agents.

| Name of Agent | Performance | Environment | Actuator | Sensor |
|---|---|---|---|---|
| Ship | Calculate ship position and heading, evaluate status (sailing, grounded, etc.) | Coastal water Underwater rocks | Rudder angle and speed | Rudder command and Speed command |
| Captain | Visual observation inside and outside the ship, listen to ship crew, Command to ship crew | Bridge deck | Verbal command and manual operation | Vision and hearing |
| SOOW | Visual observation inside and outside the ship, communicate with ship crew. | Bridge deck | Verbal command and manual operation | Vision and hearing |
| JOOW | Visual observation inside and outside the ship, communicate with ship crew and monitor route. | Bridge deck | Exchange information and manual operation | Vision and hearing |
| Helmsman | Visual observation inside and outside the ship, communicate with ship crew and execute command from Captain at the helm. | Bridge deck | Exchange information and manual operation | Vision and hearing |
| Chief Engineer | Visual observation inside the engine room, communicate with ship crew and command engine room crew. | Engine Room | Verbal command and manual operation | Vision and hearing |

man and a Chief Engineer. The following sections briefly describe the properties of these agents.

### 3.5.1 *Ship agent*

A ship agent is a mathematical model of ship manoeuvring. In this study ship is considered as a simple reflex agent because the ship behaves according to its given commands and does not behave based on its behaviour history. For example, the ship receives the rudder command given by helmsman and using this rudder command the ship agent computes its next position in the water, considering the speed, heading and turning rates are initially given. The ship will always compute its next position based on the given inputs and will not consider the new position based on old input values. Thereby the ship agent behaves like a simple reflex agent. Figure 4 shows the definition of ship agent.

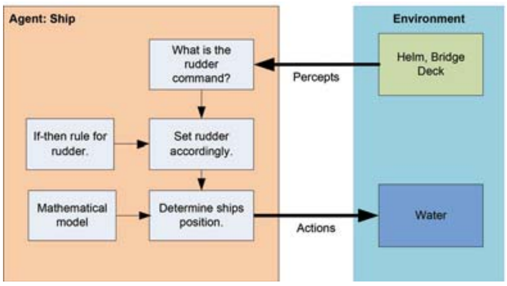


Figure 4. Definition of ship agent.

### 3.5.2 *Captain agent*

The captain of a ship is responsible for every action and its consequences that occur on-board. The Captain must control all the crew and the ship itself. In this study, the captain agent perceives the actions of ship crew and the action of the ship agent itself. Based on this perceptions and simple if-then rules the captain agent takes actions. Actions usually involve giving commands to other crew and manual operations such as controlling the engine rpm. The captain agent necessarily requires to have a set of situation-action rules based on which the agent can perceive and take action. These rules may be derived from the existing regulations and practices. Figure 5 defines the captain agent.

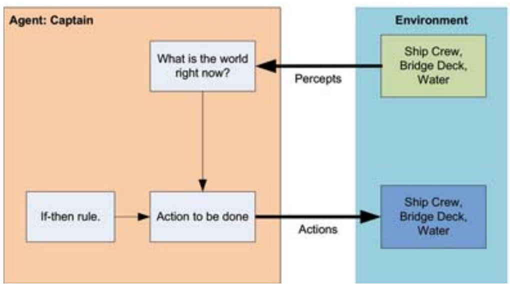


Figure 5. Definition of captain agent.

### 3.5.3 *SOOW agent*

In a ship, the senior officer of the watch needs to follow the tasks assigned by the Captain. For example, in the case of MV Costa Concordia, the SOOW was assigned to conduct ship manoeuvring

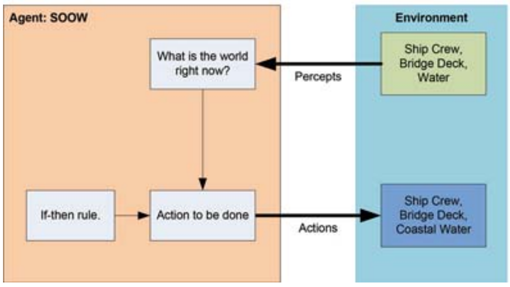


Figure 6. Definition of SOOW agent.

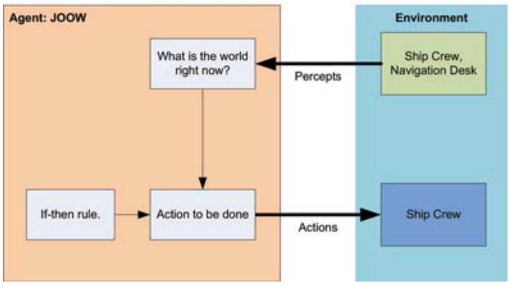


Figure 7. Definition of JOOW agent.

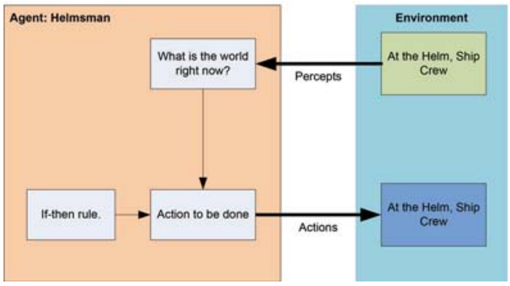


Figure 8. Definition of helmsman agent.

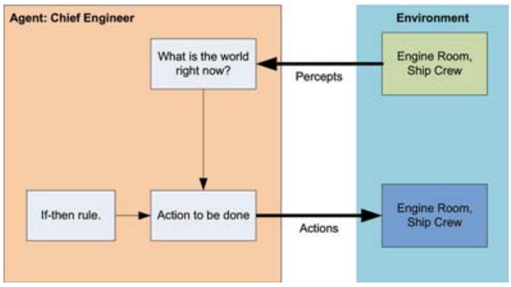


Figure 9. Definition of chief engineer agent.

and route monitoring at different times during its voyage. In this study, the SOOW agent works under the captain and his working environment is inside the bridge deck. The agent perceives from the actions of other ship crew and visual observation from bridge deck gadgets. He may order the JOOW or Helmsman and conduct manual operations (e.g. route planning). Figure 6 defines SOOW agent.

#### 3.5.4 *JOOW agent*
In a ship, the Junior Officer of the Watch (JOOW) usually works under the Captain and the SOOW and executes the orders of his or her superiors. For example, the JOOW may conduct route monitoring on the paper chart during a voyage or may execute any other command given by the Captain. In this study, the JOOW agent can perceive from the orders and actions from the ship crew. His own actions will be executing the orders from his superiors and ordering to his juniors. He may perceive from the surrounding world as well. Figure 7 defines the JOOW agent.

#### 3.5.5 *Helmsman agent*
The helmsman of a ship is the crewmember who executed the rudder command given by the ship and usually is stationed at the helm of the ship. The helmsman is often responsible for executing the engine rpm command depending on the circumstances and environment. In this study, the Helmsman agent only executes the rudder command given by the Captain. Figure 8 defines Helmsman agent.

#### 3.5.6 *Chief engineer agent*
In a ship, the chief engineer is the head of engine room crew. Generally, a ship engine is very complex device and requires several crewmembers for operation and maintenance. The chief engineer is responsible for using his engine room crew and overall operation. In this study, the chief engineer agent's function is simple and straightforward since no other engine crew is considered. The agent will perceive from bridge deck crew and the engine status. Based on the perceptions, the agent can take actions such as controlling the engine and communicating with the bridge deck crew. Figure 9 defines the chief engineer agent.

Based on the above-mentioned agents it is however, possible to conduct a logical deduction about the truth of occurrence of an accident. The following section briefly describes the logical deduction of accident by multiple agent perception-action.

Table 3. Example of perception-action predicates.

| No. | Crew | Predicates |
|---|---|---|
| 1. | Captain Perception | *Premise*: Engine RPM Low.<br>*Conclusion*: Transfer Control To Chief Engineer |
| 2. | SOOW Perception | *Premise*: Captain decided informal voyage.<br>*Conclusion*: No need to practice formal procedures. |
| 3. | Chief Engineer Action | *Premise*: Engine has history of starting problem.<br>*Conclusion*: During a voyage if the engine fails then request transfer of control from bridge deck to engine room. |

## 4 ANALYSIS AND DISCUSSION

The hypothesis of this research suggests that logical deductions of perception-action of the agents may result is the truth value of an accident. Therefore, this section discusses the perception-action of the agents using if-then rules (or knowledge base) in terms of logic predicates and attempts to deduce an outcome. Table 3 shows some example of simple if-then rules of different agents, which represent their perception and actions.

For instance, in Table 3, if the ship agent encounters an engine low rpm problem, the chief engineer will ask the captain agent to transfer the engine control from bridge to the engine room. In another instance, if the Captain adopts an informal voyage the SOOW will conduct informal procedures for voyage (such as not making a detailed voyage plan). These are the small if-then rules that construct the logic world and their agents. It is understandable that the more logic predicates are utilised the more complex the world would be and more realistic behaviour from the agents may be observed.

A knowledge base can be constructed which can be transformed into a Prolog computer program using the above mentioned predicates of different agents. These predicates may have a general structure as shown in Figure 10 (logic segment). In order to utilise the logic predicates a query is necessary to launch (Figure 10 see query segment). Whenever a query is executed, the program will search through the logics and return the truth values. If the search finds logics are true then the query will return the success in terms of conclusion as denoted by variable ‘C’in Figure 10.

Using such a framework it is possible to deduce some results. An example of logical deductions of the agents’ perceptions and actions is given in

```
logic(Conclusion, Premise1, Premise2):-

     Premise1 = _________,
     Premise2 = _________,

     Conclusion = _________.

query:-
        Logic(C, P1, P2).
```

Figure 10. Structure of a logic predicate and query in Prolog.

Table 4. Example of logical deductions base on MV Costa Concordia accident.

| Iteration | Deduction |
|---|---|
| 1. | *Captain perception*: Change voyage plan.<br>*Captain action*: Command SOOW to change voyage plan.<br>*Ship*: Idle |
| 2. | *SOOW perception*: Need to change the voyage plan.<br>*SOOW action*: Conduct route planning on large scale chat (large scale chart because of limited time)<br>*Ship*: Idle |
| 3. | *Captain perception*: Route planning is done.<br>*Captain action*: Start the voyage by commanding the helmsman.<br>*Ship*: Idle |
| 4. | *Helmsman perception*: Captains rudder and engine speed commands.<br>*Action*: Executing captains commands.<br>*Ship*: Sailing |
| 5. | *JOOW perception*: Voyage started.<br>*JOOW action*: Conduct route monitoring.<br>*Ship*: Sailing. |
| 6. | *Ship*: Grounded because the danger of rocks were not seen on large scale charts |

Table 4. In this table, it is shown that the captain decides to change the voyage plan prior to voyage and therefore orders his SOOW to take necessary actions. The SOOW understands that it is an informal voyage and there is insufficient time to adopt formal procedures for the voyage. Therefore, the SOOW conducts route planning on large-scale charts where the danger of charted rocks is not visible. As a result, at night when the JOOW was conducting the route monitoring, the JOOW did not see any danger and eventually the ship was grounded. This example however, does not exactly represent the Costa Concordia accident but resemble

Another example may be given based on the accident of MV Bright Field, which occurred in

Table 5. Example of logical deductions base on MV Bright Field accident.

| Iteration | Deduction |
|---|---|
| 1. | *Situation:* Engine power drop. *Captain perception*: Engine fault. |
| 2. | *Captain action:* Ask chief engineer to increase power. |
| 3. | *Chief engineer action*: Request engine control to be transferred to engine room from the bridge. |
| 4. | *Chief engineer perception*: System automatic re-start. |
| 5. | *Chief engineer action*: Wait until full engine restart. |
| 6. | *Situation*: Ship is in a river bend and there is limited time to take evasive action. |
| 7. | *Captain perception*: Common problem, engine will be restarted soon. |
| 8. | *Chief engineer action*: Transfer control back to bridge deck. |
| 9. | *Situation:* Too late to take evasive action so ship gets grounded. |

1996 (NTSB 1998, NASA 2010). The ship temporarily lost propulsion power as she was navigating outbound in the Lower Mississippi River at New Orleans, Louisiana. She struck a wharf adjacent to a populated commercial causing some serious injuries and a damage worth US$ 20 million. Based on this study a representative logical deduction of perception action can be made as shown in Table 5.

Thus it can be comprehended from the above analyses that logical deductions of the crew perceptions and actions may reveal the sequence of events leading to accidents. Such a technique may be utilized to uncover the hidden faults in a procedure where multiple people are involved and performing given tasks. It may be argued that the decision making process of human agents are difficult to model using mathematical equations. Under such circumstances LPT may be employed and practical solutions for accident prevention may be obtained.

## 5 CONCLUSIONS

This paper demonstrates a new concept for accident analysis technique. The study included a literature review, which demonstrated that accident theories are evolving with respect to sociotechnical context. The literature review also suggests that most accident theories only provide conceptual hypothesis and does not suggest any computational technique for analysis. Therefore, this study attempts to fill in this weakness by suggesting Logic Programming Technique (LPT) for accident analysis. Multiple agent's perception-action is proposed for utilisation in LPT. The technique is theoretically explained and examples are shown using famous maritime accidents. The study suggests that such a technique has the potential of bringing the human, social and technical factors in the same platform and provides the ability to analyse accidents in a single program. The future challenges include establishing and enriching the theoretical foundations and validation of the technique. The future of this research work seems very promising because of its practical applications and importance in saving lives and resources.

## REFERENCES


Awal, Z.I. & Hasegawa, K. 2014a. Analysis of marine accidents by logic programming technique, *Proceedings of the International Symposium on Marine Engineering (ISME), Harbin, 15–19 September 2014*. Paper-ISME127. Harbin: Harbin Engineering University (HEU).

Awal, Z.I. & Hasegawa, K. 2014b. Application of logic programming technique on maritime accident analysis, *Proceedings of the 3rd International Conference on Ship and Offshore Technology (ICSOT), Makassar, 4–5 November 2014*. 59–66. Makassar: Jurusan Teknik Perkapalan, University of Hasanuddin.

Awal, Z.I., 2007. A study on inland water transport accidents in Bangladesh: Experience of a decade (1995–2005), *International Journal for Small Craft Technology (IJSCT)* 149(B2): 35–42. London: The Royal Institution of Naval Architects (RINA).

Awal, Z.I., Islam, M.R. & Hoque, M.M. 2010. Collision of marine vehicles in Bangladesh: a study on accident characteristics, *Journal of Disaster Prevention and Management* 19(5): 582–595.

Benner, L. 1975. Accident investigations: Multilinear events sequencing methods, *Journal of Safety Research* 7(2): 67–73.

Haddon, W.Jr. 1972. Reducing highway losses: A logical framework for categorizing highway safety phenomena and activity, *Journal of Trauma* 12: 193–207.

Haddon, W.Jr. 1983. Approaches to prevention of injuries, *Presentation at the American Medical Association Conference on Prevention of Disabling Injuries*, Miami, 20 May 1983.

Hardegree, G. M. 1999. *Symbolic logic: a first course*. Mcgraw-Hill College.

Hasegawa, K. & Awal, Z.I. 2013. A concept for expert system based accident prediction technique for ship manoeuvring, *Proceedings of 5th International Maritime Conference on Design for Safety (IDFS)*, Shanghai, 25–27 November 2013. DFS-2013-044. Shanghai: Shanghai Jiao Tong University.

Heinrich, H.W. 1931. Industrial accident prevention: a scientific approach. McGraw-Hill. Quoted in Hollnagel, E. 2009. *Safer Complex Industrial Environments: A Human Factors Approach*. ISBN 1-4200-9248-0. Taylor & Francis.

Heinrich, H.W. 1941. *Industrial Accident Prevention (2nd Edition)*, New York: McGraw Hill.

Hollnagel, E. 2004. *Barriers and Accident Prevention*. Hampshire: Ashgate.

Khanzode, V.V., Maiti, J. & Ray, P.K. 2012. Occupational injury and accident research: A comprehensive review, *Safety Science* 50: 1355–1367.

Leveson, N. 2004. A new accident model for engineering safer systems, *Journal of Safety Science* 42: 237–270.

Lieto, A.D. 2012. Costa Concordia anatomy of an organisational accident, Web address: http://www.enav-international.com/wosmedia/273/costaconcordiaanatomyofanorganisationalaccident.pdf

Ministry of Infrastructures and Transports, 2013. Cruise ship Costa Concordia marine casualty on January 13, 2012—report on the safety technical investigation, Marine Casualties Investigative Body: Italy.

National Aeronautics and Space Administration (NASA) 2010. Brace for Impact, *System Failure Case Studies*, 4(10).

National Transportation Safety Board (NTSB), 1998. Marine Accident Report, PB98-916401, NTSB/MAR-98/01.

Perrow, C. 1984. *Normal Accidents Living with High-Risk Technologies*, USA: Basic Books.

Qureshi, Z.H. 2007. A review of accident modelling approaches for complex socio-technical systems, *Proceedings of the 12th Australian Conference on Safety-Related Programmable Systems*, Adelaide 30–31 August 2007, 47–59.

Qureshi, Z.H., 2008. *A Review of Accident Modelling Approaches for Complex Critical Sociotechnical Systems*, Adelaide: University of South Australia.

Rasmussen, J. & Svedung, I. 2000. *Proactive Risk Management in a Dynamic Society*, Karlstad: Swedish Rescue Services Agency.

Rasmussen, J. 1997. Risk management in a dynamic society: a modelling problem, *Safety Science* 27(2/3): 183–213.

Reason, J. 1990. The contribution of latent human failures to the breakdown of complex systems, *Philosophical Transactions of the Royal Society of London*, B327:475–484.

Reason, J. 1997. *Managing the Risks of Organizational Accidents*. England: Ashgate.

Russel, S. & Norvig, P. 2010. *Artificial Intelligence A Modern Approach (3rd Edition)*, New Jersey: Prentice Hall.

Stringfellow, M.V. 2010. *Accident Analysis and Hazard Analysis for Human and Organizational Factors*, Massachusetts: Massachusetts Institute of Technology (MIT).

Vernez, D., Buchs, D. & Pierrehumbert, G. 2003. Perspectives in the use of coloured Petri nets for risk analysis and accident modelling, *Safety Science* 41: 445–463.

Vesely, W.E., Goldberg, F.F., Roberts, N.H., and Haasl, D.F. 1981. *Fault Tree Handbook*, Washington DC: US Nuclear Regulatory Commission.

Ward, R.B. 2012. Revisiting Heinrich's law, Chemeca 2012: Quality of Life through Chemical Engineering, Wellington 23–26 September 2012, 1179–1187.