Modelling Situation-Aware Formalism Using BDI Reasoning Agents

Kiran Saleem[1] and Hafiz Mahfooz Ul Haque[2](✉)

[1] Department of Computer Science, University of Lahore, Lahore, Pakistan
kiransaleem26@gmail.com

[2] Department of Software Engineering, University of Lahore, Lahore, Pakistan
mahfoozul.haque@se.uol.edu.pk

Abstract. Natural or man-made disasters are unavoidable situations that can occur anytime and anywhere. Timely disaster response plays a vital role in reducing its after-effects and can save countless lives. Over the years, people have been developing the guidelines and processes to cope up with such kinds of hazardous situations. In recent years, situation-awareness has been considered to be the most fascinating approach for the situation assessment and provides decision support accordingly. Situation-aware systems observe/perceive dynamic changes in the environment, understand/comprehend the situation, and perform actions according to the environment. Although state-of-the-art formalisms have been developed to handle such kinds of hazardous situations intelligently and rescue the victims. However, there are still many uncontrolled challenging issues. In this paper, we present a Belief-Desire-Intention (BDI) based multi-agent formalism to model the context-aware decision support system dynamically in order to achieve the desired goals. To illustrate the use of the proposed formalism, we develop a simple case study in which BDI agents are modeled and simulated to present results in terms of agents' reasoning processes. The behavior of the system has been tested using the NetLogo simulation environment to rigorously evaluate the validity of the system.

Keywords: Situation-awareness · BDI · Multi-agent system · Netlogo

1 Introduction

A natural disaster is an inevitable situation that can occur at any time and anywhere. It has varied forms such as earthquakes, floods, hurricane, wildfire, etc., and different level of occurrences has been recorded from mild to an intense level. A severe level of disasters usually deprives everything including people from their homes and even from their relatives. Numerous affected people become helpless and they wait for getting help to escape themselves from such kinds of vilest situations. Earthquake is considered to be the most dangerous natural disaster in the world and much effort has been made to predict the earthquake but unable

P. C. Vinh and A. Rakib (Eds.): ICCASA 2020/ICTCC 2020, LNICST 343, pp. 169–181, 2021.
https://doi.org/10.1007/978-3-030-67101-3_14

to find prolific results so far. On the other hand, with the technological advancements and growing interest in smart and intelligent devices, significant efforts have been made to escape and/or avoid such kinds of hazardous situations using these tiny smart devices along with the intelligent decision support systems.

In recent years, with the advent of the pervasive computing environment, context-awareness has gained significant attention, as context-aware systems acquire/sense information from distant locations using smart mobile devices, perform reasoning, and adapt behavior to take the right decision at the right time and in right place accordingly. The recent notion of situation-awareness has emerged from context-awareness. Context-aware systems have a very broad domain of applications for complex problem-solvings in the pervasive environment. In contrast, situation-aware systems perceive the dynamic changes in the environment within a volume of time and space, comprehend the situation, and perform actions according to the situation [20,24]. In recent years, situation-awareness has gained much attention in safety-critical domains and in emergency situations such military command and control operation [3], health care [2], emergency situations [24], navigation [18], and aviation [7]. To model such situations, agent-based technology has emerged as a new paradigm for conceptualizing, designing, and implementing sophisticated software systems. An agent could be a software agent, that perceives its environment and takes actions to perform specifically assigned tasks. According to Chen [6], *"We humans are inherently context-aware agents"*. Context-aware agents sense the environment and perform reasoning in order to achieve the desired goal. Literature has witnessed various multi-agent based complex applications, as multi-agent systems use different reasoning techniques to handle the different nature of problems. Among others, BDI (Belief-Desire-Intention) based reasoning has been advocated as the most optimistic approach due to its simplistic nature and the usage of folk psychological notions which corresponds to the human behavior as humans think, plan and make an intention to meet their desired goals [10]. BDI architecture presents the artificial agents with human-like decision-making strategies, based on their belief, desire, and intention components. Combining these two techniques such as BDI agents and situation-awareness leads a better decision-making mechanism [4]. In the literature, Feng et al. [9] present a situation-awareness based context-aware decision support system using an agent-based approach. In this framework, they incorporated a shared situation-aware model that provides human objects with their customized views and services through collaborative agents' planning to reduce the cognitive load of human objects. They have applied rule-based reasoning to model context-aware decision support systems and simulated command and control domains. The entity agents perform event classification and action recommendations with a high level of accuracy. Valette et al. [22] proposed BDI agents based evacuation model to deal with the disaster situation like a fire. They have used the BDI paradigm to model the cognition of agents along with their communication plan to infer their behavior in terms of emotions. The authors have implemented the model in the simulation environment to analyze the variability of the results. In [1], authors proposed BDI based multi-agent formalism for ontology-driven health-care system. In this model, they have developed an intelligent decision support system using BDI reasoning agents to infer the desired goals dynamically. This work is an extension of the previous work [1] where BDI agents based intelligent decision support system has been developed for an ontology-driven health care system. However, the work in BDI agents based on intelligent decision support systems incorporating the notion of situation-awareness is still in its infancy stage. In this paper, we intend to provide customized services out of shared situation-awareness based on contextual knowledge. We propose BDI agents based context-aware decision support system to assist hazardous situations dynamically in the affected area due to earthquake hit. We use the NetLogo simulator for BDI agents modelling and simulation to monitor the overall behavior of the system and rigorously evaluate the validity of the system.

The rest of the paper is structured as follows. Section 2 discusses the core notions of BDI and NetLogo simulation. Section 3 presents a formalism for BDI reasoning agents to model the context-aware decision support system. In Sect. 4, we discuss a case study of a smart rescue system. For this, we use the NetLogo simulator for the prototypal implementation using BDI agents' modelling and simulation to evaluate the validity of the system and finally conclude in Sect. 5.

2 Preliminaries

2.1 BDI (Belief, Desire, Intention)

BDI (Belief, Desire, Intention) is a classical paradigm to formalize the internal structure of cognitive agents for complex problem-solving in a real-time environment. The BDI paradigm attempts to capture the general understanding of how humans observe and gain knowledge from the environment, have a perception about it, and develop a set of beliefs on it. Based on the existing information and with a newly gained set of hypothetical beliefs, humans intend to take a set of actions in order to achieve the desired goals efficiently. The concept of BDI is based on human decision making strategy, BDI agents acquire information from the environment or other agents and have a set of beliefs on it, perform reasoning to take specifically assigned task, and then make individual or collaborative plan among multi-agent systems to achieve the desired goals [5,8,23]. The BDI reasoning agents consist of a set of predicates having a set of agent's own beliefs, desires, and intentions. A predicate may have a name along with the parameter containing the values according to the situation. For example; *isSmokeDetected('true, GPSLocation('a, 'b))* – meaning that Smoke is detected at location('a, 'b). The architecture of BDI agents has three core components:

- **Beliefs** (what agents think): It is a set of facts stored in the agent's memory or agent's internal knowledge. For example; whenever an agent derives a predicate *isSmokeDetected('true, GPSLocation('a, 'b))*, whatever the predicate that may either be true or false, it will be added to agent's belief set.

- **Desires** (Intending to do): The ultimate objective of an agent is to accomplish the goal, i.e., *isSmokeDetected('false, GPSLocation('a, 'b))* – The desire is to put water to remove smoke. However, the agent's desire may have priority to select new intentions based on the desire and the agent's belief.
- **Intentions** (Act of doingness): The plans or set of actions that agents intend to follow and peruse their desires. Intentions can also be prioritized based on the selected plan to fulfill the desire.

BDI agents have strong roots in philosophy and are extensively used for modelling and simulation of complex systems using intelligent agents. BDI reasoning paradigm allows the systems to design and model a multi-agent system to do complex reasoning.

2.2 NetLogo

In the literature, a significant effort has been done for modelling and simulations using multi-agent systems such as Netlogo [4,14,17], Jason [13], JADE [11], GAMA [22], etc. Among other simulation tools, we opt the NetLogo simulation tool because it is a multi-agent based programming platform to model and simulate natural or unusual phenomena. The NetLogo platform was introduced in 1999 by Uri Wilensky. It is open-source software that includes a graphical development interface, a set of commands to create agent-based models, and a suite of tools to assist the development of the model, and the analysis of simulation results [19,21]. NetLogo can be used to design complex and reactive multi-agent systems by modelling a great number of intelligent agents, operating independently as well as collaboratively. It also powers HubNet participatory simulations [17]. NetLogo is used in different kinds of scenarios for instance in gaming, military purpose and in disaster management scenarios, etc. [4,14]. There are four types of agents in the NetLogo named as turtle, links, patch, and observer. Turtle agents can be used to model the agents' behavior, Link agents establish the connection between agents, patches are stationary agents with a set of predefined activities, and the observer oversees the set of activities.

NetLogo also supports BDI agents modelling and simulation. Authors in [15] introduced additional libraries for BDI agents' reasoning and communication mechanism. The BDI library can be imported in the NetLogo Simulation environment to implement the belief-desire-intention based multi-agent system. We have chosen BDI agents to model the system in NetLogo because it has the capability to capture the complex adaptive behavior using the pro-active goal-seeking approach.

3 BDI Agents Based Situation-Aware Formalism

In this section, we propose a BDI reasoning agents based formalism to provide customized services out of shared situation-awareness based on contextual knowledge. The primary roles of agents are to observe/acquire the environment situation, perform BDI reasoning to model a context-aware decision support system in a command and control setting. The system consists of a set of BDI agents where each agent in the system has a plan consisting of a set of beliefs along with a set of actions. In this formalism, the BDI agents have context-aware capabilities to identify the nature of the problem from the environment, classify the contextual knowledge based on the agent's belief set, recommend suitable action, and select the decision mode. As the system is designed for complex and safety-critical problems in nature, human assistance and support play vital roles to cope up with hazardous situations. However, the system is autonomous in nature and has an intelligent decision-making capability with minimal human involvement. The system is classified into an automated mode and semi-automated mode. In automated mode, group agents plan collaboratively with the set of specified actions, communicate themselves, and select the optimal goals according to the situation. Whereas agents in the semi-automated mode need human assistance due to the high consequences of the situation, as agents do not have the capability to take appropriate action in a rapidly changing environment. For example; severe damages of the earthquake and its after-effects. In such cases, human assistance is continuously required.

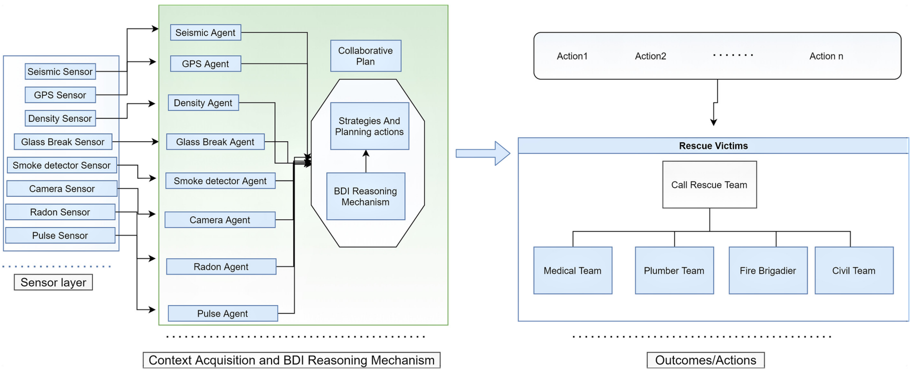

Fig. 1. Layered architecture of the proposed formalism

As shown in Fig. 1, the proposed framework consists of three layers, namely: (a) context acquisition, (b) BDI agents reasoning, and (c) agents modelling and simulation. In the context acquisition layer, we assume that BDI agents acquire the contextual information from sensors that can be installed to monitor the hazardous situation. In this model, every sensor is connected to its corresponding BDI based agent where an agent in the system can have a belief based on the existing knowledge or newly acquired knowledge from other agents/sensors.

BDI Agent's activity is driven by its belief set, intentions, and desires; given below:

- Perception of elements in the current situation: Agent's activity is driven by its belief set, however, the agent's belief set can be revised as the situation varies in the dynamic environment.

– Comprehension of the current situation: Each agent in the system observes the situation in the environment after certain intervals of time, revise its belief set, select the execution plan to attain its goal.
– Projection of future status: After selecting the execution plan, agents trigger the actions and act accordingly in order to achieve its desire objective.

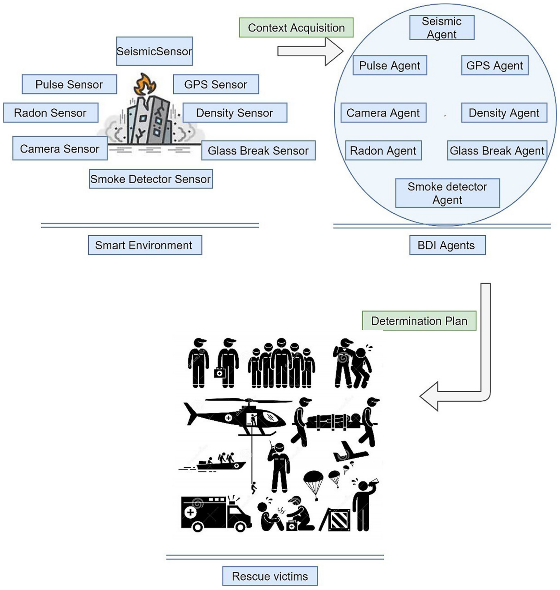

Fig. 2. System flow: BDI agents modelling and simulation

In the BDI agent's reasoning layer, each agent in the system performs the specifically assigned task to achieve the desired goals individually or collaboratively. For this, each agent has a plan consisting of a set of predicates having its own set of beliefs, desires, and intentions for each action. As BDI agents perform goal-directed tasks based on the set of beliefs, where the belief of the agents can be revised dynamically whenever the situation changes. The main inspiration is that each agent updates its belief every time whenever the situation changes keeping the most recent values in the memory. To model the communication among different agents, an agent can communicate with other agents only if it needs to have additional information in its belief set to produce the desired result or to achieve the goal state in order to fulfill the agent's intentions and desires.

4 BDI Agents Modelling and Simulation Using NetLogo

To illustrate the use of the proposed formalism, we present a smart rescue system to observe the after-effects of earthquakes and rescue the victims. For this, we develop a context-aware decision support system for modelling BDI agents and simulate the validity of the results to see whether the system would be able to achieve the desired goals effectively or not? NetLogo has additional libraries for BDI agents' reasoning and communication mechanism that can be imported in the NetLogo simulation environment to implement the belief-desire-intention based multi-agent system [15,16].

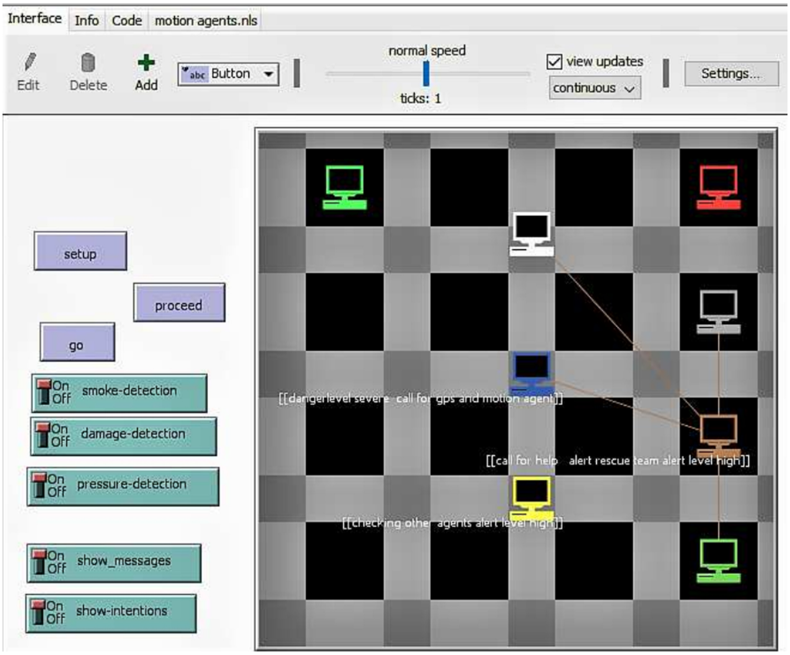

Fig. 3. BDI agents' modeling

We develop a case study of a smart rescue system to illustrate the use of the proposed formalism. A severe earthquake is considered to be the most dangerous disaster that can have a deteriorate after-effects such as fire, building damages if the situation would not handle in time [12]. We consider the earthquake evacuation system to handle the disaster situation. In the system design, we consider a number of intelligent BDI reasoning agents to monitor the current status of the affected area. For example, a number of smart and intelligent devices/sensors are considered to monitor a current situation, which updates their status whenever the situation changes. The Fig. 2 depicts the system flow using BDI agents modelling whereas the Fig. 3 shows the execution process in the NetLogo simulation environment. The description of the BDI agent's task are listed below:

– Seismic Agent: When the earthquake hits, the seismic agent acquires the data from the sensor with earthquake magnitude value and save newly acquired data in its belief set. Then it takes a set of actions to pursue the desire. The seismic agent generates alerts based on the magnitude level which is forwarded to the concerned agents.

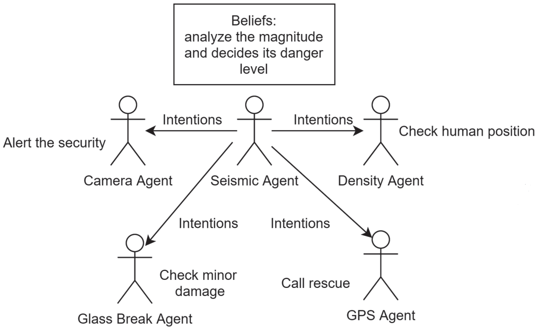

Fig. 4. Seismic agent's design

- Density Agent: The density agents acquire the data (total count of the people nearby the affected area) from a density sensor. The density agent is activated whenever it receives earthquake alerts from the seismic agent.
- Glass Break Agent: This agent ensures the safety of all glass windows and its effects on the people. It notifies RescueAgent to take appropriate action.
- Pulse Agent: Pulse sensor detects the severe building damages by a ratio called inter-story drift. As the earthquake hits, Pulse Agent acquires pulses after certain intervals of time and sends updates to the consequent agents to take suitable action in case of abnormalities.
- Smoke Detector Agent: This agent may alarm in case if smoke or fire is detected. As this information arrives in the agent's belief set, the information is forwarded to the density agent to check the existence and counts of the people and notifies RescueAgent to rescue the people.
- Radon Agent: When the earthquake hits, the density agent alerts the radon agent to detect the gas leakages, thus if the detection would be true then the density agent checks human counts near it. The radon agent performs reasoning on the acquired data by its existing facts, thus plan the actions to achieve its desires according to the situation.
- Camera Agent: The level disaster or severity of the damage can be observed through a Camera agent.
- GPS Agent: It is activated immediately after receiving GPS location requests from other agents. The system detects the exact location of the notified area and tags itself with other agents to call RescueAgents.

As the framework is based on a multi-agent system, agents in the system perform BDI reasoning in order to monitor the hazardous situations, take decisions intelligently based on the existing set of beliefs, and infer the preventive plan to rescue the victims from the disaster situation. As shown in Table 1, the system needs a set of BDI predicates to infer the desired goal. The BDI predicate of each agent has a belief set, intentions, and desires to fulfill agents' desired objectives. NetLogo provides additional library support to design BDI agents. Agents are modeled with their belief sets, intentions, and desires as shown in

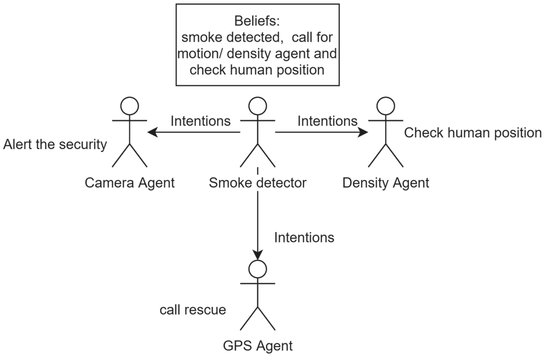

Fig. 5. Smoke agent's design

Fig. 4 and Fig. 5, all other agents are designed in the same pattern as per their set of beliefs and communication plan.

```
  if dangerlevel >= 7 [ set agentbelief create-belief "danger level" "severe"
    add-belief create-belief "danger level" "severe"
    print "***********Belief Of Seismic Agent*************************"
    show agentbelief
     print "-----------------Invoke Intention------------------"
    call-rescue]
  ; intentions
if dangerlevel >= 7 [
  add-intention "dangerlevel severe " "call for gps and motion agent"
  add-intention "call-to-agents" "true"]
   ;plan
  broadcast-to gpsAgents add-content (list "rescue need as soon as possible for seismic agent" )
  create-message "Earthquake"
  broadcast-to camAgents add-content (list "alert the security" )
  create-message "earthquake"
  broadcast-to motionAgents add-content (list "check for positions via Density/ Motion Agent" )
  create-message "acquire data about humans"
  broadcast-to GlassBreakAgents add-content (list "check the severity on building damage" )
  create-message "check minor damage, earthquake severe"
```

Fig. 6. Seismic agent code

For modelling multi-agent systems in NetLogo, we have created 8 files(nls) of the mentioned agents. Initially, we model all agents with their belief set along with their execution and communication plan in the NetLogo simulator with their specified actions. At the initial stage of the simulation environment, all agents in the system have predefined belief sets and execution plans. As the system proceeds with the dynamic changes, agents revise their belief sets according to the situation, select the best available plan, and execute. So when we simulate the smart rescue system, the Seismic agent observes the situation, earthquake hits with the magnitude value +7. As this magnitude value is high, the system immediately activated and generate alerts to all agents in the system to perform their corresponding tasks. The seismic agent updates its beliefs about magnitude value and communicates with the corresponding agents such as the density agent, GPS agent, glass break agent, camera agent to revise their belief set and act accordingly. The Fig. 6 shows a fragment of the Seismic agent's Netlogo code. As the density agent gets an earthquake alert, it exchanges information with the radon and smoke agents to observe the human presence at the smoked and gas leakage area. The radon agent checks gas mains breakage or pressure detection (site 1), the smoke detector agent activates in case of fire or smoke detected (site 2), the camera agent sends an alert message to the security, the density agent checks human counts, and notify rescue agent. As the radon agent detects the gas mains are broken or any leakage at site 1, the radon agent notifies the density agent. The density agent observes the human presence at site 1, updates its belief set, sends a notification to the GPS agent, and camera agent. In case of fire or smoke detected at site 2, the smoke agent notifies the density agent. Then the density agent counts the human presence at site 2, updates its belief set. The radon agent shares the GPS location of the affected site to rescue the victims. As the Seismic agent shares the magnitude value with the glass break agent, the glass break agent revises its belief and notifies rescue agents. Based on the situation, the rescue agent notifies the rescue team to rescue the victims. This the severity level, rescue action needs to be taken on a priority basis. Figure 7 depicts a few agent's execution processes along with their beliefs, intentions, and executions. We have tried to simulate the system with the maximum possible execution steps in order to test and simulate the designed model of the system. We analyze the behavior of the system and rigorously evaluate the validity of the system.

Table 1. BDI reasoning agent's data model.

BDI agents	Context acquisition	Belief set	Intention	Desire (Goal)
Seismic Agent	Get earthquake magnitude value	- Earthquake magnitude value	- Check human position - Call rescue	- Call rescue agent to help the victims
Density Agent	- Seismic agent alert - Human counts	- Earthquake hit - Building damage detected	- Check for human counts - Call GPS agent - Alert Camera	- Call rescue team - Help victims
Glass Break Agent	- demage level - Minor damage (frequency detection)	- Earthquake hit	- Check damage severity	- Check severity of the building damage
Smoke Detector Agent	- Smoke alert	- Earthquake hit - Smoke detected	- Alert Density agent - Alert GPS agent - Alert Camera agent	- Call rescue team - Check human count - Alert the security - Help victims
Radon Agent	- Gas leakage	- Earthquake hit - Gas leakage detected	- Alert Density agent - Alert GPS agent - Alert Camera agent	- Call rescue team - Check human count - Alert the security - Help victims

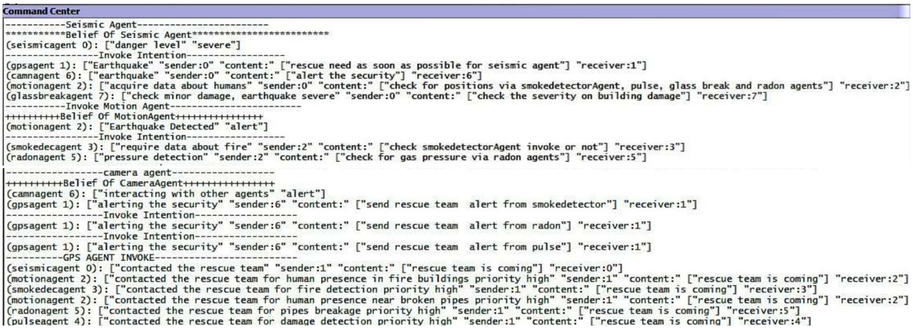

Fig. 7. BDI agents execution process

5 Conclusion

In this paper, we proposed BDI agents based formalism to model the context-aware decision support system to suitably manage the disaster management system. To evaluate the scalability and validity of the system, we tested the proposed system in the NetLogo simulation environment using BDI agents modelling, reasoning, and simulation. Although it is the implementation of a simple case study and early stage in the development process, however, this work has shown promising results using BDI agents. In future work, we will implement the system physically using a comprehensive case study. For this, we will use NetLogo as a middle-ware platform, as it allows the API facility and interoperability with other platforms such as Python, Matlab, Jason.

References

1. Akhtar, S.M., Nazir, M., Saleem, K., Haque, H.M.U., Hussain, I.: An ontology-driven IoT based healthcare formalism. Int. J. Adv. Comput. Sci. Appl **11**(2), 479–486 (2020)
2. Alkhomsan, M.N., Hossain, M.A., Rahman, S.M.M., Masud, M.: Situation awareness in ambient assisted living for smart healthcare. IEEE Access **5**, 20716–20725 (2017)
3. Boril, J., Smrz, V., Mach, O.: Development of experimental methods for testing of human performance in the framework of future millitary pilot's preparation. In: 2017 International Conference on Military Technologies (ICMT), pp. 548–552. IEEE (2017)
4. Buettner, R., Baumgartl, H.: A highly effective deep learning based escape route recognition module for autonomous robots in crisis and emergency situations. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)
5. Caillou, P., Gaudou, B., Grignard, A., Truong, C.Q., Taillandier, P.: A simple-to-use BDI architecture for agent-based modeling and simulation. In: Jager, W., Verbrugge, R., Flache, A., de Roo, G., Hoogduin, L., Hemelrijk, C. (eds.) Advances in Social Simulation 2015. AISC, vol. 528, pp. 15–28. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-47253-9_2

6. Chen, H., Tolia, S.: Steps towards creating a context-aware software agent system. HP. Technical report HPL-2001-231 (2001)
7. Endsley, M.R., Garland, D.J.: Pilot situation awareness training in general aviation. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 44, pp. 357–360. SAGE Publications, Los Angeles (2000)
8. Evertsz, R., Thangarajah, J., Ly, T.: A BDI-based methodology for eliciting tactical decision-making expertise. In: Sarker, R., Abbass, H.A., Dunstall, S., Kilby, P., Davis, R., Young, L. (eds.) Data and Decision Sciences in Action. LNMIE, pp. 13–26. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-55914-8_2
9. Feng, Y.H., Teng, T.H., Tan, A.H.: Modelling situation awareness for context-aware decision support. Expert Syst. Appl. **36**(1), 455–463 (2009)
10. Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M.: The belief-desire-intention model of agency. In: Müller, J.P., Rao, A.S., Singh, M.P. (eds.) ATAL 1998. LNCS, vol. 1555, pp. 1–10. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49057-4_1
11. Junger, D., Guinelli, J., Pantoja, C.E.: An analysis of Javino middleware for robotic platforms using Jason and JADE frameworks. In: 10th Software Agents, Environments and Applications School (2016)
12. Little, B.: The deadliest earthquake ever recorded (2020). https://www.history.com/news/the-deadliest-earthquake-ever-recorded
13. Ramirez, W.A.L., Fasli, M.: Integrating NetLogo and Jason: a disaster-rescue simulation. In: 2017 9th Computer Science and Electronic Engineering (CEEC), pp. 213–218. IEEE (2017)
14. Luna Ramirez, W.A., Fasli, M.: Plan acquisition in a BDI agent framework through intentional learning. In: Berndt, J.O., Petta, P., Unland, R. (eds.) MATES 2017. LNCS (LNAI), vol. 10413, pp. 167–186. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64798-2_11
15. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Enhancing NetLogo to simulate BDI communicating agents. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 263–275. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87881-0_24
16. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Teaching intelligent agents using NetLogo. In: ACM-IFIP IEEIII, pp. 209–221 (2008)
17. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: MAS coursework design in NetLogo. In: Proceedings of the International Workshop on the Educational Uses of Multi-Agent Systems (EDUMAS 2009), pp. 47–54 (2009)
18. Saus, E.R., Johnsen, B.H., Eid, J., Thayer, J.F.: Who benefits from simulator training: personality and heart rate variability in relation to situation awareness during navigation training. Comput. Hum. Behav. **28**(4), 1262–1268 (2012)
19. Schermer, B.W.: Software agents, surveillance, and the right to privacy: a legislative framework for agent-enabled surveillance (2007)
20. Stanton, N.A., Chambers, P.R., Piggott, J.: Situational awareness and safety. Saf. Sci. **39**(3), 189–204 (2001)
21. Tisue, S., Wilensky, U.: NetLogo: a simple environment for modeling complexity, pp. 16–21, January 2004
22. Valette, M., Gaudou, B., Longin, D., Taillandier, P.: Modeling a real-case situation of egress using BDI agents with emotions and social skills. In: Miller, T., Oren, N., Sakurai, Y., Noda, I., Savarimuthu, B.T.R., Cao Son, T. (eds.) PRIMA 2018. LNCS (LNAI), vol. 11224, pp. 3–18. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03098-8_1

23. van Oijen, J., van Doesburg, W., Dignum, F.: Goal-based communication using BDI agents as virtual humans in training: an ontology driven dialogue system. In: Dignum, F. (ed.) AGS 2010. LNCS (LNAI), vol. 6525, pp. 38–52. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18181-8_3
24. Yau, S.S., Huang, D., Gong, H., Davulcu, H.: Situation-awareness for adaptive coordination in service-based systems. In: 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), vol. 1, pp. 107–112. IEEE (2005)