Complex Adaptive Systems Modeling

RESEARCH

Open Access

Formal model of earthquake disaster mitigation and management system

Nazir Ahmad Zafar* and Hamra Afzaal

*Correspondence:
nazafar@gmail.com
Department of Computer Science, COMSATS Institute of Information Technology, Sahiwal Campus, Sahiwal 57000, Pakistan

Abstract

Wireless sensor and actors networks (WSANs) have become an important research area due to its large number of applications in safety, security and mission-critical systems. Natural disasters such as earthquakes and floods have distressing effects on human lives, economy and environment particularly in the developing countries due to their high population and lack of infrastructure. Earthquake is one of the major such disasters which causes a huge loss in terms of deaths, environment damages and loss of property because of its unpredictable nature. There exists much work on earthquake prediction, disaster mitigation and management but mostly is based on simulation and testing techniques which have certain limitations. Formal methods are mathematical approaches which assure correctness of systems to overcome limitations of simulation and testing techniques. That is why a formal system of earthquake disaster mitigation and management using formal methods and WSANs is proposed. Sensors and actors are deployed in the earthquakes vulnerable areas in the form of subnets which increase energy efficiency of the network as the processing becomes localized at a subnet level. Firstly, graph theory is used to represent subnet-based model which is then transformed into a formal model. Vienna Development Method-Specification Language (VDM-SL) is used to describe and prove correctness of the formal specification. The developed specification is then validated and verified through VDM-SL Toolbox facilities by analyzing the pre/post conditions and invariants over the formal system.

Keywords: Earthquake mitigation and management, Wireless sensor and actor networks, Graph theory, VDM-SL, Validation and verification

Background

Wireless sensor and actor networks (WSANs) consist of sensors and actors connected through wireless medium. In the network, sensors are used to sense complex events in an environment and actors take intelligent decisions and perform actions as required. Sensors are cheap devices which have low power batteries, slow processing capabilities and short communication range as compared to actors which are more expensive and powerful in terms of resources. It is noted that actors are capable of sensing the environment but sensors do not have ability to perform any action (Akyildiz and Kasimoglu 2004). As an example from military defence system, sensors detect enemy troops and report to actors which destroy the troops by coordinating with each other. For an effective monitoring and decision making, there must be a continuous sensor–sensor, sensor–actor and actor–actor communication to perform an appropriate action whenever

Springer Open

it is required. WSANs have become an important area of research due to their large number of applications in safety and mission-critical systems in harsh and inhospitable environment. Earthquakes, volcanoes, floods, storms and droughts are the examples of natural disasters that have devastating effects on human, animal lives, economy and environment. The developing countries are greatly affected by the natural disasters due to high population, poverty, lack of infrastructure and development. Further, the natural disasters are increasing due to unstable landforms and climate.

Earthquake is one of the major natural disasters caused due to sudden movements of tectonic plates in the earth crust. Mostly, the earthquakes are unpredictable and can occur at any terrestrial location. Earthquakes can reach to several miles, beyond the countries and continents in few seconds causing extreme destruction. Mainly, there are two important issues, that is, earthquake prevention and detection, and disaster mitigation and management. Disaster mitigation and management is an important area of research as earthquake prevention is not under very much control. For example, there are earthquake early warning systems under practice in several countries which use science and technology for monitoring systems to alert the devices and people. However, it is possible only for seconds to minutes to alert the people and systems in advance to take actions.

There exists much work on disaster mitigation and management (Tran and Saito 2016; Solmaz and Turgut 2017; Erdelj et al. 2017) but mostly it is based on simulation and testing techniques which have certain limitations. For example, simulation techniques do not assure about complete correctness of a system. Moreover, the number of inputs for a complex system may increase exponentially making impossible to evaluate an entire system. Formal methods (FMs) are effective tools to provide assurance about correctness to overcome limitations of testing and simulation techniques. Very little work is found on modelling of earthquake management systems (EMS) using formal methods (Sarmad et al. 2010), which is taken as a starting point in this research. In this work, the authors have proposed a framework of agent-based EMS using Pi-calculus and Pi-ADL formal specification languages which are abstract in nature and cannot be used for visualization of results. Formal methods are mathematical notations which are used for modelling, specifying and analyzing properties of safety, mission-critical and complex systems. Further, FMs are proved successful to develop precise, unambiguous, structured and correct specification at an appropriate level of abstraction (Zhozhikashvili 2014). As there is a need to mitigate the damages caused by the earthquakes, therefore an effective model of earthquake disaster mitigation and management using formal methods and WSANs is proposed in this work. WSANs are complex adaptive systems (CASs) because of their ability to work in complex adaptive and dynamic environment (Altamimi and Ramadan 2016; John 2006). Further, WSANs involve many interacting components such as, sensors, actors, gateways that behave as dynamic components that interact with the environment. Developing and modelling of algorithms using WSANs have raised various research questions. For example:

- How to deploy resource-constrained sensors/actors to achieve energy efficiency of the network?
- How to make a system failure free to ensure its correct functioning?

- How to minimize the consumption of resources and processing cost in WSAN?
- How to develop communication and coordination mechanisms to guarantee timely execution of actions?
- How to ensure the security of the information communicated among nodes in the network?

This paper is mainly focussed on the first two research questions by developing a localized and energy efficient algorithm for earthquake disaster mitigation and management system. The network topology is described using graph theory. This is because graphs and networks are similar in nature as vertices in a graph represent sensors, actors or gateways in a network and edges in the graph represent communication links between the network's nodes. That is why graph based model is developed to represent, store, process and transmit the earthquake information in a network. Further, it is easier to transform the graph-based model into a formal model by developing a mapping between both of these approaches. Graph-based model is defined below to describe the network topology as shown in Fig. 1.

This work is continuation of our earlier work in this area (Afzaal and Zafar 2016a, b, c). In this paper, an algorithm is proposed which assumes that sensors and actors are deployed in the form of subnets in the areas which are more affected by the earthquakes.

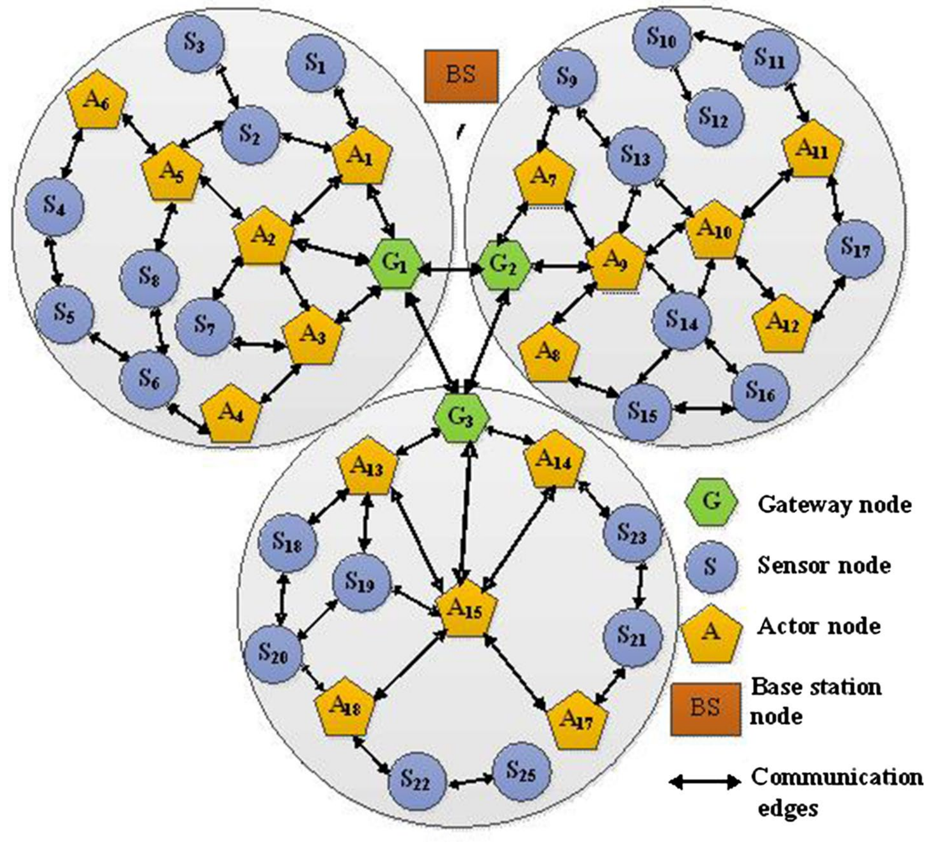


Fig. 1 Representation of subnets-based WSANs

The subnet-based deployment makes the model more energy efficient as the problem becomes localized in a subnet. A subnet employs large number of sensors and a less number of actors with a specified gateway node. The subnets are connected through gateway nodes which are connected with a base station for disseminating the information. The proposed model is a combination of centralized and distributed approach. The approach is centralized as the whole information of a subnet is transmitted towards the gateway node which issues order to an actor to perform an action. The centralized approach is more suitable in static environment but it increases computational cost in terms of energy and time. In the distributed approach, the control is distributed among sensors and actors for sensing and performing the actions. The distributed approach is useful in a dynamic and unpredictable environment however it is not efficient as it may have a lack of complete information. In the proposed model, sensors are used to predict the earthquakes and actors are used to take measures to minimize the damages. The gateway node is assumed to communicate earthquake information to other gateway nodes and to decide about the required actions by the actors in a subnet.

Formal model is described using Vienna Development Methods-Specification Language (VDM-SL) which has a detailed level of description with computer tools support (SCSK Corporation 2013). In the model, topology of the network is specified as dynamic graph due to mobile nodes which may change their position frequently. The nodes are specified as composite objects in VDM-SL. For describing safe behaviour, several invariants are specified over the composite object types. The state space of the system is defined which involves definitions of state, functions and operations to detect the earthquake and mitigate its damages. For verification of consistent and correct behaviour, pre and post conditions are defined over the functions and operations. The developed formal specification is analysed through various techniques supported by the VDM-SL toolbox (SCSK Corporation 2013) to increase the confidence of correctness. Rest of the paper is organized is as follows: The next section illustrates an introduction to complex adaptive systems. System model, problem statement and the proposed algorithm are presented in "Methods" section. Formal specification and its analysis are presented in "Results and discussion" section. In "Related work" section, the related work is discussed critically. Conclusion and future work are given in "Conclusions" section.

Complex adaptive systems

Complex adaptive systems (CASs) are dynamic networks consisting of large number of agents, species, individuals, firms, nations and even cells that may act in parallel or act and react constantly in response to each other. Artificial neural networks, artificial intelligence systems and evolutionary programs are few examples of CASs. The control of CASs is required to be coherent as it is greatly centralized and decentralized, and its agents make their own decisions. The agents influence and communicate with each other to change their behaviour based on experience learned from the environment for achieving a required goal.

Complex adaptive systems can be analysed through theoretical, experimental and applied methods, for example, computer-based simulations and mathematical modelling. Due to a complex nature of CASs, it is required to analyse it through rigorous tools. Mostly, CASs are analysed through computer simulation tools using agent based methodologies (ABMs) and complex networks (CNs). In the artificial world, ABMs are used for detailed simulation of CASs to describe communications and actions of the agents. Agent-based models use recursive mathematical functions and computer code to apply for definite inputs. However, mathematical and statistical analysis techniques are still important in developing and testing the ABMs. CASs, such as, neural networks, biological systems and chemical systems are effectively defined through CNs. Previously, graphs are used to define these systems where agents are denoted as nodes and communications between agents are represented through edges. Mainly, CN is used to define structural properties of a CAS to govern behaviour of the agents. How to develop model of a large network, which have complex topology, arises a serious research question in applications of CNs. Most of the researchers have focussed on simulations and testing techniques, i.e., ABMs and CNs for modelling and verification of CASs. These techniques lack in guaranteeing the correctness of systems as mentioned above. Further, if there is a need of enhancement in the system then regression testing is needed to be performed in which it is required that a complete set of simulations must be re-performed. Formal methods are focussed on proving correctness of models and are effective to overcome the drawbacks of simulations and testing techniques (Shah et al. 2015). Moreover, formal methods are effective in providing better understanding of a system therefore simulations can be performed after developing formal specification. Z, B and RAISE are well known formal notations used at an abstract level of specification. Z notation is used for specifying formal models for WSANs and MAHSNs in mission critical systems (Imran et al. 2016; Alnuem et al. 2014). Statecharts and Petri nets are modelling languages which are powerful for describing behaviour but lack in providing proof concepts for complex and critical systems. Behavioural changes of component-based systems are studied to elaborate a coordination protocol to maintain a consistency by implementing distributed mechanism (Hadj et al. 2009). In this approach, correct coordination is focussed rather than considering dependency relations among the multiple adaptations. The study is motivated by the potential benefits of using formalisms using Coloured Petri Nets with sufficient abstraction.

The VDM-SL is applied in this research as it is a matured formal specification language which has been used successfully for various industrial projects in a cost-effective manner. The VDM-SL has a rigorous computer tool support to construct and animate the computer models providing its detailed level examination. The IFAD VDM Toolbox, VDM-Tools, Crescendo, Overture and Symphony are few important tools being used for validating and verifying the VDM specification.

Methods

Problem statement and system model

It is impossible to predict an accurate time and location of an earthquake due to their unpredictable nature. However, using an alert system, it is possible to provide the earthquake information and precautionary measures to the people to save their lives. WSANs can be used for predicting earthquakes to minimize the after consequences. An algorithm for Earthquake Disaster Mitigation and Management (EDMM) is developed assuming a planned deployment of sensors, actors and gateways in the form of connected subnets in the areas which are greatly affected by the earthquakes. The benefit of partitioning the network into subnets is that the computation/processing localizes to large extent in the subnet which increases the energy efficiency of the network. A subnet employs large number of sensors, small number of actors and a gateway node. Actors are more powerful in terms of battery, processing power and communication range. The gateway node is assumed as most powerful actor node in the subnet. For detection of an earthquake, three types of sensors are employed (Rahman et al. 2016), i.e., animal, water pressure and radon sensors. Animal sensors are deployed on bodies of animals to measure their body temperature and detect their behavior. It is assumed that an animal sensor becomes active if the current body temperature of an animal becomes less than or greater than the normal body temperature, otherwise sensor's mode remains in sleeping state. Water pressure sensors are deployed in underground wells and water channels and on the ground at different places where the ground water flow is constant. It is assumed that a water pressure sensor becomes active when it detects that the current water pressure crosses its threshold limit, otherwise it remains in sleeping mode. Radon sensors are deployed on rocks, soil and underground water to detect the emission of radon. It is assumed that a radon sensor becomes active if it detects the emission of radon, otherwise remains in sleeping mode. When a sensor becomes active then it reports to the nearby actors about the detected information and then the actors report to a gateway node of the same subnet. After predicting earthquake the gateway node communicates with the base station. Through the base station, the information is disseminated towards people using information and communication technologies such as radio broadcasting, electronic media and social media. The gateway node can communicate with other gateways about the earthquake information and issues command to the actor nodes for necessary actions, e.g., dislocate alive people inside the camps, dislocate alive animals outside the camps, provide first aid to people, search nearest hospital, move severely injured people to hospital, move dead people to dead house. The system model is presented in the form of a flow chart as in Fig. 2.

Algorithm

A high level pseudo code of the EDMM algorithm is proposed as in Fig. 3. Different types of nodes are deployed after pre-planning in the form of subnets (line 1) then a gateway node is selected in a subnet (line 2). If an animal sensor detects earthquake then mode of the sensor become active, sleep otherwise (lines 3–7). If a water pressure sensor detects earthquake then it become active, sleep otherwise (lines 8–10). If a radon sensor detects earthquake then it becomes active, otherwise remains in the sleeping mode (lines 11–13). If mode of a sensor becomes active in a subnet then earthquake is predicted (lines 14–15). The sensors of a subnet report to its actors which report to the gateway for communication with the neighbour gateways (lines 16–17). The earthquake information is circulated among people and then the gateway node issues order to actors of the subnet (lines 18–20). Alive people are dislocated by the actors to save their lives (lines 21–23). If a person receives normal injury then actors provide first aid otherwise the person is moved to the nearest hospital (lines 24–27). In case of death, a nearest hospital is searched and the person is moved to its dead house (28–29). Similarly, animals are dislocated to save the lives (lines 30–32). The functions and notations used in the algorithm are explained below:

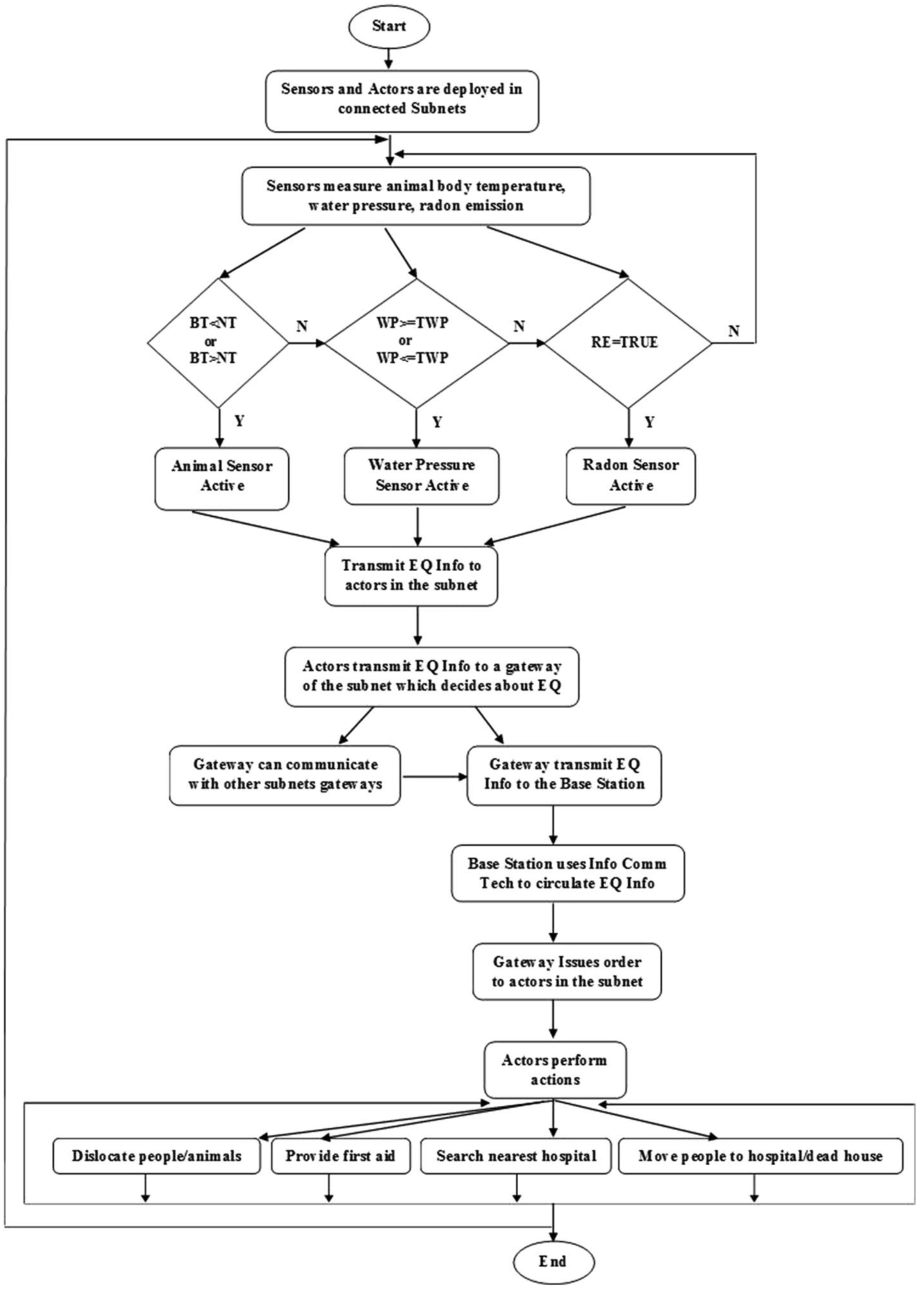
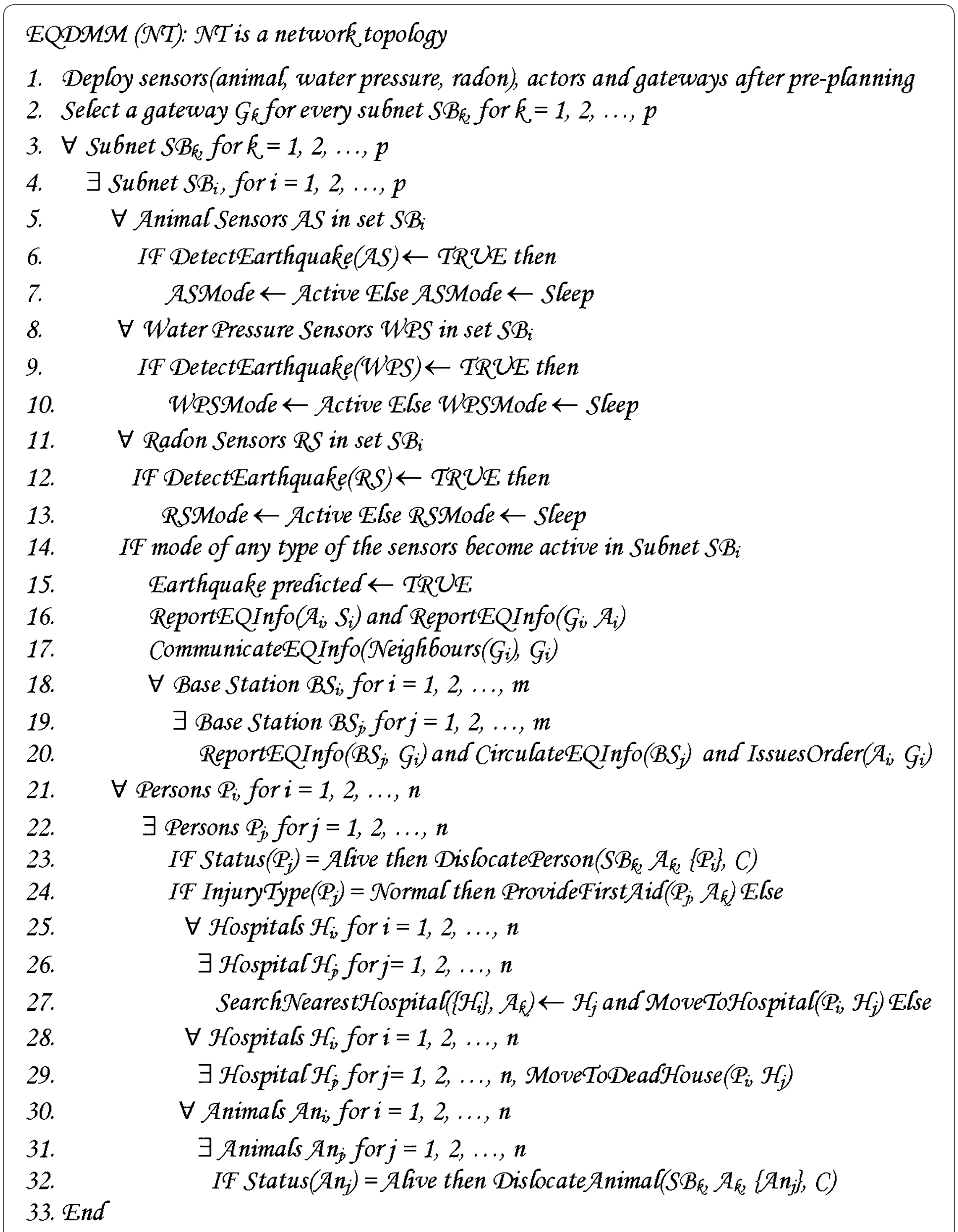

Start
Sensors and Actors are deployed in connected Subnets
Sensors measure animal body temperature, water pressure, radon emission
BT<NT or BT>NT
N
WP>=TWP or WP<=TWP
N
RE=TRUE
N
Y
Y
Y
Animal Sensor Active
Water Pressure Sensor Active
Radon Sensor Active
Transmit EQ Info to actors in the subnet
Actors transmit EQ Info to a gateway of the subnet which decides about EQ
Gateway can communicate with other subnets gateways
Gateway transmit EQ Info to the Base Station
Base Station uses Info Comm Tech to circulate EQ Info
Gateway Issues order to actors in the subnet
Actors perform actions
Dislocate people/animals
Provide first aid
Search nearest hospital
Move people to hospital/dead house
End

Fig. 2 Flowchart of the proposed model

- NT = Network Topology(S, A, G, BS, SB, L)
- S = Set of all sensors = $\{S_1, S_2, \ldots, S_x\}$
- A = Set of all actors = $\{A_1, A_2, \ldots, A_y\}$
- G = Set of all gateways = $\{G_1, G_2, \ldots, G_p\} \subseteq \{A_1, A_2, \ldots, A_y\}$
- BS = Set of all base stations = $\{BS_1, BS_2, \ldots, BS_z\}$
- Maximum size of a subnet = s
- Number of subnets = p = (No. of sensors + No. of actors)/s
- SB = Set of all subnets = $\{SB_1, SB_2, \ldots, SB_p\}$



```
EQDMM (NT): NT is a network topology
Deploy sensors(animal, water pressure, radon), actors and gateways after pre-planning
Select a gateway $G_k$ for every subnet $SB_k$, for $k = 1, 2, \ldots, p$
∀ Subnet $SB_k$, for $k = 1, 2, \ldots, p$
  ∃ Subnet $SB_i$, for $i = 1, 2, \ldots, p$
    ∀ Animal Sensors AS in set $SB_i$
      IF DetectEarthquake(AS) ← TRUE then
        ASMode ← Active Else ASMode ← Sleep
    ∀ Water Pressure Sensors WPS in set $SB_i$
      IF DetectEarthquake(WPS) ← TRUE then
        WPSMode ← Active Else WPSMode ← Sleep
    ∀ Radon Sensors RS in set $SB_i$
      IF DetectEarthquake(RS) ← TRUE then
        RSMode ← Active Else RSMode ← Sleep
    IF mode of any type of the sensors become active in Subnet $SB_i$
      Earthquake predicted ← TRUE
      ReportEQInfo($A_i$, $S_i$) and ReportEQInfo($G_i$, $A_i$)
      CommunicateEQInfo(Neighbours($G_i$), $G_i$)
      ∀ Base Station $BS_i$, for $i = 1, 2, \ldots, m$
        ∃ Base Station $BS_j$, for $j = 1, 2, \ldots, m$
          ReportEQInfo($BS_j$, $G_i$) and CirculateEQInfo($BS_j$) and IssuesOrder($A_i$, $G_i$)
    ∀ Persons $P_i$, for $i = 1, 2, \ldots, n$
      ∃ Persons $P_j$, for $j = 1, 2, \ldots, n$
        IF Status($P_j$) = Alive then DislocatePerson($SB_k$, $A_k$, {$P_i$}, C)
        IF InjuryType($P_j$) = Normal then ProvideFirstAid($P_j$, $A_k$) Else
          ∀ Hospitals $H_i$, for $i = 1, 2, \ldots, n$
            ∃ Hospital $H_j$, for $j= 1, 2, \ldots, n$
              SearchNearestHospital({$H_i$}, $A_k$) ← $H_j$ and MoveToHospital($P_i$, $H_j$) Else
          ∀ Hospitals $H_i$, for $i = 1, 2, \ldots, n$
            ∃ Hospital $H_j$, for $j= 1, 2, \ldots, n$, MoveToDeadHouse($P_i$, $H_j$)
          ∀ Animals $An_i$, for $i = 1, 2, \ldots, n$
            ∃ Animals $An_j$, for $j = 1, 2, \ldots, n$
              IF Status($An_j$) = Alive then DislocateAnimal($SB_k$, $A_k$, {$An_j$}, C)
End
```

Fig. 3 High level pseudo code of the algorithm

- L = Set of all possible links among sensors, actors, gateways and base stations
- ReportEQInfo(N_1, N_2) == N_2 reports earthquake information to N_1
- Neighbors(N) == returns set of all neighbors of N
- CommunicateEQInfo(N_1, N_2) == N_1 and N_2 share information with each other
- IssuesOrder(N_1, N_2) == N_2 issues order to N_1
- DislocatePersons(SB, A, {P}, C) == return true if the person is dislocated in a camp
- InjuryType(P) == returns the type of injury
- ProvideFirstAid(P, A) == An actor A provides first aid to a person P
- SearchNearestHospital(H, A) == return the nearest hospital from the set of all hospitals

- MoveToHospital(P, H) == Person P is moved to the hospital H
- MoveToDeadHouse(P, H) == Person P is moved to a dead house in the hospital H

Results and discussion

Formal specification using VDM-SL

Formal specification of the proposed model is described using VDM-SL in this section. Several constructs, for example, sets, composite objects, invariants, pre/post conditions are used for developing the formal specification. The static and dynamic models are defined for the proposed model. The static model includes the definition of data types while the dynamic model specifies the state and operations.

Static model

The static model of the algorithm using WSANs is specified by employing sensors and actors having some common fields. The common information is defined by the composite object *Node* which has seven fields, i.e., *nid*, *type*, *mode*, *pwr*, *connected*, *position* and *eqinfo*. The description of these fields is given in Table 1.

```
types
EQData = token;
Position :: xco : int
            yco : int;
EQInfo :: eqdata : set of EQData
          sensed : bool
          transmitted : bool
          received : bool;
ID = token; Mode = <<SLEEP>> | <<ACTIVE>>; Pwr = <<HIGH>> |<<LOW>>;
Type = <<SENSOR>> | <<ACTOR>> | <<GATEWAY>> | <<BSTATION>>;
Node :: nid : ID
        type : Type
        mode : Mode
        pwr : Pwr
        connected : bool
        position : Position
        eqinfo : EQInfo;
```

Table 1 Fields of node

#	Field name	Field description
1.	*Nid*	Every node has a unique identifier
2.	*Type*	A node may be sensor, actor, gateway or a base station
3.	*Mode*	A node may be in sleeping or active mode
4.	*Pwr*	A node may have some power either high or low
5.	*Connected*	It describes the connectivity of a node
6.	*Position*	It defines the position of a node in terms of x and y-coordinates
7.	*Eqinfo*	Represents earthquake data which may be sensed, transmitted or received

Any two nodes in the network are to be connected by an edge which depends upon type of the nodes. The connectivity of the whole network is described by the edges relation.

```
EType = <<SS>> | <<SA>> | <<AA>> | <<AG>> | <<GG>> | <<GB>>;
Edge :: n1 : Node
        n2 : Node
        etype : EType
inv mk_Edge(n1, n2, -) == n1<>n2;
Edges = set of Edge
inv edges ==
forall eg in set edges &
 eg.n1.type=<<SENSOR>> and eg.n2.type = <<SENSOR>> => eg.etype=<<SS>> and
 eg.n1.type=<<SENSOR>> and eg.n2.type = <<ACTOR>> => eg.etype = <<SA>> and
 eg.n1.type=<<ACTOR>> and eg.n2.type = <<ACTOR>> => eg.etype = <<AA>> and
 eg.n1.type=<<ACTOR>> and eg.n2.type = <<GATEWAY>> =>eg.etype = <<AG>> and
 eg.n1.type=<<GATEWAY>> and eg.n2.type=<<GATEWAY>> => eg.etype = <<GG>> and
 eg.n1.type=<<GATEWAY>> and eg.n2.type=<<BSTATION>> => eg.etype = <<GB>>;
```

Invariants (1) An edge employs two distinct nodes, that is, there is no loop in the network. (2) The edge type *SS* describes that the nodes are sensor type. (3) The edge type *SA* defines that one is a sensor and the other is actor node. (4) The edge type *AA* shows that two actor nodes are connected by an edge. (5) The edge type *AG* represents that an actor node communicates with a gateway node through an edge. (6) The edge type *GG* illustrates that the edge employs two gateway nodes. (7) The edge type *GB* shows the connectivity of a gateway node with the base station.

Sensors are used to detect the earthquake where a sensor is defined by the composite object *Sensor* which has four fields, i.e., *snode*, *stype*, *status* and *snbrs*. The fields are described in Table 2.

```
SType = <<ASENSOR>> | <<WPSENSOR>> | <<RADSENSOR>>;
Anomly = <<DETECTED>> | <<NOTDETECTED>> ;
Sensor :: snode : Node
          stype : SType
          status : Anomly
          snbrs : set of Node
inv mk_Sensor(snode, stype, status, snbrs)==
snode.pwr = <<LOW>> and stype = <<ASENSOR>> or
stype = <<WPSENSOR>> or
stype = <<RADSENSOR>> and status = <<DETECTED>> <=>
 snode.mode = <<ACTIVE>> and status = <<NOTDETECTED>> <=>
 snode.mode = <<SLEEP>> and snode.connected = true <=> snbrs <> {} and
snode.connected = false <=> snbrs = {};
Action = <<IDLE>> | <<SAVELIVES>> | <<PROVIDE_FIRST_AID>> |
<<SEARCH_APLACE>> | <<SEARCH_HOSPITAL>>;
```

Table 2 Fields of sensor

#	Field name	Field description
1.	*Snode*	A sensor is a node in the network
2.	*Stype*	The sensor type may be animal sensor, water pressure sensor or a radon sensor
3.	*Status*	This field is used to record the anomaly status
4.	*Snbrs*	This field describes the set of neighbours of a sensor node

Invariants (1) A sensor node has low power. (2) There are three types of sensor nodes, i.e., animal, water pressure or radon sensor. (3) The status of a sensor node is detected if it is in active mode and it is not detected if it remains in sleeping mode. (4) A sensor node is connected if it is connected with neighbour nodes.

Three types of sensors are used for detecting earthquake, i.e., animal, water pressure and radon sensors. Firstly, formal specification of animal sensor is described which is a composite object *ASensor* having four fields, i.e., *asnode*, *animals*, *temp* and *abehavior*. The description of these fields is provided in Table 3.

```
ANStatus = <<ALIVE>> | <<DEAD>>;
Animal :: id : ID
          position : Position
          anstatus : ANStatus;
Temp :: btemp : int
        nbtemp : int;
ABehavior = <<NORMAL>> | <<ANXIETY>> | <<FEAR>> | <<ESCAPE>>;
ASensor :: asnode : Node
           animals : set of Animal
           temp : Temp
           abehavior : ABehavior
inv mk_ASensor(asnode, animals, temp, abehavior)==
forall an in set animals &
 an.id = asnode.nid and
 asnode.mode = <<ACTIVE>> <=> temp.btemp < temp.nbtemp or
  temp.btemp > temp.nbtemp => abehavior = <<ANXIETY>> or
  abehavior = <<FEAR>> and abehavior = <<ESCAPE>> and
  asnode.mode = <<SLEEP>> <=> temp.btemp = temp.nbtemp =>
  abehavior = <<NORMAL>>;
```

Invariants (1) A unique animal sensor node is deployed on the body of an animal. (2) The mode of animal sensor node is active if and only if current body's temperature of the animal is at least normal body's temperature. The animal sensor detects a fear or anxiety

Table 3 Fields of animal sensor

#	Field name	Field description
1.	*Asnode*	The animal sensor is a node in the network
2.	*Animals*	It is set is of animals
3.	*Temp*	This field records body temperature of an animal
4.	*Abehavior*	The animal sensor detects a change in the behaviour of an animal

in the animal and it escapes from the current place. If the above conditions are not satisfied, then mode of the sensor remains *sleep*.

The water pressure sensor is used to detect level of the water pressure. It is defined as a composite object *WPSensor* which consists of four fields, i.e., *wpsnode*, *wpsdeployed*, *wpslocation* and *wpressure*. The detail of the fields is given in Table 4.

```
UGround = token;Ground = token;WPSLocation = <<UGROUND>> | <<GROUND>>;
WPressure :: cwp : int
             thwp : int;
WPSensor :: wpsnode : Node
            wpsdeployed : map ID to Position
            wpslocation : WPSLocation
            wpressure : WPressure
inv mk_WPSensor(wpsnode, wpsdeployed, wpslocation, wpressure)==
forall n in set dom wpsdeployed & n = wpsnode.nid and
 wpslocation = <<UGROUND>> or wpslocation = <<GROUND>> and
wpsnode.mode = <<ACTIVE>> <=> wpressure.cwp >= wpressure.thwp and
wpsnode.mode = <<SLEEP>> <=> wpressure.cwp < wpressure.thwp;
```

Invariants (1) Sensors are deployed under/on ground locations to measure water pressure which are uniquely identified. (2) The water pressure sensor is active if and only if current water pressure becomes greater than or equal to the threshold limit, otherwise, the sensor remains in sleeping mode.

The radon sensor is defined as a composite object *RadSensor* which consists of four fields, i.e., *rsnode*, *raddeployed*, *rslocation* and *rademission*. The detail description of the fields is given in Table 5.

```
RSLocation = <<ROCKS>> | <<SOIL>> | <<UGWATER>>;
RadSensor :: rsnode : Node
             raddeployed : map ID to Position
             rslocation : RSLocation
             rademission : bool
inv mk_RadSensor(rsnode, raddeployed,rslocation, rademission)==
forall n in set dom raddeployed &
 n = rsnode.nid and rslocation = <<ROCKS>> or
rslocation = <<SOIL>> or rslocation = <<UGWATER>> and
rsnode.mode = <<ACTIVE>> <=>  rademission = true and
rsnode.mode = <<SLEEP>> <=>  rademission = false;
```

Invariants (1) Unique radon sensors are deployed on rocks, soil and underground water for detecting radon emission. (2) A radon sensor becomes active if it detects emission otherwise it remains in sleeping mode.

Table 4 Description fields of water pressure sensor

#	Field name	Field description
1.	*Wpsnode*	It represents identifier of the water pressure sensor which is a node in the network
2.	*Wpsdeployed*	It is a unique water pressure sensor node which is deployed on every position
3.	*Wpslocation*	The location of the water pressure sensor which might be on ground or under the ground
4.	*Wpressure*	It records the water pressure

Table 5 Fields of radon sensor

#	Field name	Field description
1.	*Rsnode*	The radon sensor is a node in the network
2.	*Raddeployed*	A unique radon sensor node is deployed at every position
3.	*Rslocation*	Radon sensors are located at rocks, soil or underground water to radon emission
4.	*Rademission*	This field is used to check the emission of the radon

Sensors detect the earthquake information and report to actors. An actor is defined as a composite object *Actor* consisting of four fields which are explained in Table 6.

```
Action = <<IDLE>> | <<SAVELIVES>> | <<PROVIDE_FIRST_AID>> |
<<SEARCH_APLACE>> | <<SEARCH_HOSPITAL>>;
Actor :: acnode : Node
         sense : Sensor
         action : Action
         anbrs : set of Node
inv mk_Actor(acnode, sense, action, anbrs)==
acnode.pwr = <<HIGH>> and sense.status = <<DETECTED>> =>
 action = <<SAVELIVES>> and action = <<PROVIDE_FIRST_AID>> and
 action = <<SEARCH_APLACE>> and action = <<SEARCH_HOSPITAL>> and
sense.status = <<NOTDETECTED>> => action = <<IDLE>> and
acnode.connected = true <=> anbrs <> {} and
acnode.connected = false <=> anbrs = {};
```

Invariants (1) An actor node is of high power as compared to a sensor. (2) If the earthquake is detected then actor performs actions, i.e., saves lives, provides first aid, searches places for allocation and searches hospital to save the lives. (3) If the earthquake is not detected then actor remains idle. (4) An actor node is connected if and only if its neighbour set is non-empty, not connected otherwise.

Table 6 Fields of actor

#	Field name	Field description
1.	*Acnode*	Used to represent actor node in the network
2.	*Sense*	An actor has ability for sensing information
3.	*Action*	Actors are able for performing actions
4.	*Anbrs*	This field describes set of the neighbours of an actor node

The gateway node is the most powerful actor node specified as *Gateway* having four fields, i.e., *ganode*, *acnode*, *issueorder* and *gnbrs*. The fields of a gateway node are explained in Table 7.

```
Gateway :: ganode : Node
           acnode : Actor
           issueorder : bool
           gnbrs : set of Node
inv mk_Gateway(ganode, acnode, -, gnbrs) ==
ganode.pwr = <<HIGH>> and ganode.eqinfo.received = true <=>
 acnode.sense.status = <<DETECTED>> and
ganode.eqinfo.received = false <=>acnode.sense.status = <<NOTDETECTED>> and
ganode.connected = true <=> gnbrs <> {} and
ganode.connected = false <=> gnbrs = {};
```

Invariants (1) The gateway node is of high power. (2) It receives earthquake information if and only if an actor detects it. (3) It does not receive any information if an actor does not detect it. (4) The connectivity of a gateway node is assured if and only if it has neighbour nodes.

A gateway node communicates with the base station to disseminate earthquake information. The fields of a base station are presented in Table 8.

```
CTechnologies = <<ELECTRONIC_MEDIA>> | <<SOCIALMEDIA>> | <<RADIO>>;
BStation :: bnode : Node
            ceqinfo : CTechnologies
            bnbrs : set of Node
inv mk_BStation(bnode, ceqinfo, bnbrs)==
bnode.pwr = <<HIGH>> and
bnode.eqinfo.received = true => ceqinfo = <<ELECTRONIC_MEDIA>> and
ceqinfo = <<SOCIALMEDIA>> and ceqinfo = <<RADIO>> and
bnode.connected = true <=> bnbrs <> {} and
bnode.connected = false <=> bnbrs = {};
```

Invariants (1) A base station node should be of high power. (2) If it receives information about the earthquake then it uses communication technologies to disseminate the information. (3) The base station node is assumed as connected if it has neighbour nodes.

Table 7 Fields of gateway

#	Field name	Field description
1.	*Ganode*	The gateway is a node in the network
2.	*Acnode*	The gateway node has the features of an actor node
3.	*Issueorder*	The gateway node issues order to the actor nodes
4.	*Gnbrs*	It defines the set of neighbours of a gateway node

Table 8 Fields of base station

#	Field name	Field description
1.	*Bnode*	A base station is assumed as a node in the network
2.	*Ceqinfo*	This field is used for communication of earthquake information
3.	*Bnbrs*	It describes the neighbors of a base station node

A planned deployment of the network topology is assumed in the form of subnets and is specified as *Subnet* having six fields, i.e., *nodes*, *sensors*, *actors*, *gateway*, *edges* and *position* as in Table 9.

```
Subnet :: nodes : set of Node
          sensors : set of Sensor
          actors : set of Actor
          gateway : Gateway
          edges : Edges
          position : Position
inv mk_Subnet(nodes, sensors, actors, gateway, edges, -)==
forall nd1, nd2 in set nodes & exists eg in set edges &
 nd1 = eg.n1 and nd2 = eg.n2 and
forall eg in set edges & eg.n1 in set nodes and
 eg.n2 in set nodes and
 exists s in set sensors & exists a in set actors &
 nd1 = s.snode or nd1 = a.acnode or nd1 = gateway.ganode or
 nd2 = s.snode or nd2 = a.acnode or
 nd2 = gateway.ganode and
forall nd1, nd2 in set nodes &
 nd1.eqinfo.sensed <> nd2.eqinfo.sensed and
 exists d1 in set nd1.eqinfo.eqdata &
 exists d2 in set nd2.eqinfo.eqdata & d1 <> d2 and
forall nd1, nd2 in set nodes &
 nd1.eqinfo.transmitted <> nd2.eqinfo.transmitted and
 exists d1 in set nd1.eqinfo.eqdata &
 exists d2 in set nd2.eqinfo.eqdata & d1 <> d2 and
forall sn1, sn2, rn in set nodes &
 exists d1 in set sn1.eqinfo.eqdata &
 exists d2 in set sn2.eqinfo.eqdata &
  exists d in set rn.eqinfo.eqdata & d = d1 and d = d2 and
  rn.eqinfo.received = true <=> d1 <> d2 and
  sn1.eqinfo.transmitted = true and sn2.eqinfo.transmitted = true;
```

Invariants (1) Any two nodes in a subnet are connected through an edge. (2) Every edge employs two nodes which may be sensor, actor or a gateway. (3) Any two nodes cannot sense the same information and both the nodes have different data about the earthquake. (4) Any two nodes cannot transmit the same earthquake information which shows that both have different data. (5) The earthquake data is received by a node from any two nodes if and only if the data is different.

Table 9 Fields of subnet

#	Field name	Field description
1.	*Nodes*	A subnet consists of set of nodes
2.	*Sensors, actors, gateway*	Set of sensors, actors and a gateway exist in a subnet
3.	*Edges*	Edges are used to describe communication links between the nodes
4.	*Position*	It records the position of a subnet

The network is deployed in the form of subnets and is defined as a composite object *Network* having three fields, i.e., *subnets*, *edges* and *bstations* which are described in Table 10.

```
Network :: subnets : set of Subnet
           edges : Edges
           bstations : set of BStation
inv mk_Network(subnets, edges, bstations)==
forall s1, s2 in set subnets & s1.nodes union s2.nodes <> {} and
 s1.nodes inter s2.nodes = {} and
forall s1 in set subnets & exists s2 in set subnets &
 exists eg in set edges & eg.etype = <<GG>> and
 eg.n1 = s1.gateway.ganode and
 eg.n2 = s2.gateway.ganode and
forall eg in set edges & eg.etype = <<GG>> and
 exists s1,s2 in set subnets & eg.n1 = s1.gateway.ganode and
 eg.n2 = s2.gateway.ganode and
exists s in set subnets & exists bs in set bstations &
 exists beg in set edges & beg.etype = <<GB>> and
 beg.n1 = s.gateway.ganode and
 beg.n2 = bs.bnode;
```

Invariants (1) The union of nodes of any two subnets should not be empty. (2) The intersection of any two subnet nodes should be empty because a subnet must employ unique nodes. (3) The subnets are connected through edges and type of an edge is gateway–gateway link. It represents that the edge employs two gateway nodes. (4) There must exist a subnet which is connected with a base station. It means that there must be an edge between a gateway and a base station node.

The person is specified as a composite object *Person* having six fields, i.e., *id*, *position*, *address*, *mobile*, *pstatus* and *itype*. The fields are described in Table 11.

Table 10 Fields of network

#	Field name	Field description
1.	*Subnets*	It represents set of subnets in the network
2.	*Edges*	It represents set of all edges in the network
3.	*Bstations*	This field describes that the network is connected with a base station

Table 11 Fields of person

#	Field name	Field description
1.	*Id*	It represents that every person has a unique identifier
2.	*Position*	Every person exists on a certain position
3.	*Address*	A person has an address
4.	*Mobile*	A person may have a mobile phone
5.	*Pstatus*	It describes the status of a person which may be alive or dead
6.	*Itype*	It represents that the injury may be severe or normal

```
PStatus = <<ALIVE>> | <<DEAD>>; Address = seq of char; Mobile = token;
IType = <<SEVERE_INJURY>> | <<NORMAL_INJURY>>;
Person :: id : ID
          position : Position
          address : Address
          mobile : Mobile
          pstatus : PStatus
          itype : IType;
```

If a person suffers from an injury then there is a need to move him/her to a hospital or even if a person dies then he needs to be moved to a dead house in a hospital. Therefore hospital \is defined as a composite object *Hospital* having three fields, i.e., *id*, *position* and *hlocation*. Table 12 explains the fields.

```
HLocation = <<DEAD_HOUSE>> | <<ROOM>>;
Hospital :: id : ID
            position : Position
            hlocation : HLocation;
```

The earthquakes may destruct houses of the people and then people have to shift to some other places and camps can be arranged there which have basic necessities of life. That is why camp is specified having five fields, i.e., *id*, *position*, *facilities*, *capacity* and *clocation* described in Table 13.

```
CLocation = <<INSIDE>> | <<OUTSIDE>>; Facilities = token;
Camp :: id : ID
        position : Position
        facilities : Facilities
        capacity : nat
        clocation : CLocation;
```

Table 12 Fields of hospital

#	Field name	Field description
1.	*Id*	It states that every hospital is unique
2.	*Position*	It is located on a certain position
3.	*Hlocation*	It describes that the locations in a hospital are dead house or treatment rooms

Table 13 Fields of camp

#	Field name	Field description
1.	*Id*	It describes that every camp is unique
2.	*Position*	It is located on a certain position
3.	*Facilities*	It contains the basic facilities of life
4.	*Capacity*	A camp has a certain capacity
5.	*Clocation*	A camp has a space inside and outside

Dynamic model

The dynamic model is described in the form of a state, functions and operations. The state of the EDMM is described as *EarthQuakeDMManagement*. The attributes that define the state are already described, which are initialized using an *init* function.

```
state EarthQuakeDMManagement of
 network : [Network]
 subnets : set of Subnet
 nodes : set of Node
 asensors : set of ASensor
 wpsensors : set of WPSensor
 radsensors : set of RadSensor
 actors : set of Actor
 gateways : set of Gateway
 bstations : set of BStation
 edges : Edges
 persons : set of Person
 camps : set of Camp
 animals : set of Animal
 hospitals : set of Hospital
inv mk_EarthQuakeManagement(network, -, nodes, -, -, -, -, -, -, edges, -,
-, -, -) ==
forall nd1, nd2 in set nodes & exists eg in set edges &
 nd1 = eg.n1 and nd2 = eg.n2 and
forall eg in set edges & eg.n1 in set nodes and eg.n2 in set nodes and
forall s in set network.subnets &
 forall s1,s2 in set s.sensors & exists eg in set s.edges &
 eg.n1 = s1.snode and eg.n2 = s2.snode and
 forall a1,a2 in set s.actors & exists eg in set s.edges &
 eg.n1 = a1.acnode and eg.n2 = a2.acnode and
 forall s in set network.subnets & exists a in set s.actors &
 (exists g : Gateway & g = s.gateway and
  exists eg in set s.edges & eg.n1 = g.ganode and eg.n2 = a.acnode) and
forall s1 in set network.subnets & exists s2 in set network.subnets &
 exists eg in set network.edges &
 eg.n1 = s1.gateway.ganode and eg.n2 = s2.gateway.ganode and
exists s in set network.subnets &
 exists bs in set network.bstations &
 exists g : Gateway & g = s.gateway and
 exists eg in set network.edges &
  eg.n1 = g.ganode and eg.n2 = bs.bnode
init eqm == eqm = mk_EarthQuakeManagement(nil, {}, {}, {}, {}, {}, {}, {},
{}, {}, {}, {}, {}, {})
end
```

Invariants (1) Any two nodes in a network are connected through an edge. (2) Every edge must employ two nodes. (3) In a subnet, there exist edges, i.e., sensor–sensor, sensor–actor, actor–actor and actor–gateway. (4) Any two subnets are connected through gateway–gateway edges. (5) There must exist a gateway node in a subnet which is connected with a base station.

The operations are defined which change state space of the system. Firstly, formal specification of detection of earthquake process is described as an operation *EQDetected*. It takes a subnet as input and returns true as output if the earthquake is detected. The animal, water pressure and radon sensors are read in the external clause.

```
EQDetected(subnet : Subnet)detected : bool
ext rd network : [Network]
 rd asensors : set of ASensor
 rd wpsensors : set of WPSensor
 rd radsensors : set of RadSensor
pre true
post detected <=>
exists asn in set asensors &
 asn.asnode.mode = <<ACTIVE>> => asn.temp.btemp < asn.temp.nbtemp or
 asn.temp.btemp > asn.temp.nbtemp and
 asn.abehavior = <<ANXIETY>> or asn.abehavior = <<FEAR>> and
 asn.abehavior = <<ESCAPE>> or exists wps in set wpsensors &
 wps.wpsnode.mode = <<ACTIVE>> =>wps.wpressure.cwp >= wps.wpressure.thwp or
exists rds in set radsensors &
rds.rsnode.mode = <<ACTIVE>> => rds.rademission = true;
```

Pre/post-conditions (1) In post-condition, it is defined that there exist animal sensors which if become active then it means that they detect that body temperature of an animal is less than or greater than the normal body temperature. The animal sensors detect anxiety and fear in the body of the animal. (2) There exists water pressure sensors which if become active, they detect that water pressure crosses certain threshold water pressure limit. (3) There exist radon sensors which if become active then it shows that they detect emission of radon gas.

After detection of earthquake, it is reported, which is specified as an operation *EQReported*. The operation takes *detected* variable as input for the verification of detection of earthquake and returns true as output if it is reported. The state component, network, is read in the external clause and is not modified.

```
EQReported(detected : bool)reported : bool
ext rd network : [Network]
pre detected = true
post reported <=>
exists s in set network.subnets & exists sen in set s.sensors &
 exists a in set s.actors & exists g : Gateway & g = s.gateway and
 exists b in set network.bstations &
 a.acnode.eqinfo.received = sen.snode.eqinfo.transmitted and
 g.ganode.eqinfo.received = a.acnode.eqinfo.transmitted and
 b.bnode.eqinfo.received = g.ganode.eqinfo.transmitted;
```

Pre/post-conditions (1) In the pre-condition, it is verified that the earthquake is detected. (2) The reporting process of earthquake in the network is described as: (i) There exists a subnet in the network in which a sensor transmits the earthquake information to an actor. (ii) The actor transmits the received information to the gateway node in a subnet. (iii) The gateway node transmits the received information to the base station.

When the earthquake information is reported, then actions are taken which is defined as an operation *ActionPerformed* which takes *reported* variable as input for verification of reporting of the earthquake. The state components, *network*, *persons*, *camps*, *animals* and *hospitals* are read in the external clause.

```
ActionPerformed(reported : bool)
ext rd network : [Network]
 rd persons : set of Person
 rd camps : set of Camp
 rd animals : set of Animal
 rd hospitals : set of Hospital
pre reported = true
post exists s in set network.subnets & forall a in set s.actors &
 exists g : Gateway & g = s.gateway and
 a.acnode in set g.gnbrs and
 g.issueorder = true and
 a.action = <<SAVELIVES>> and
 forall p in set persons & p.pstatus = <<ALIVE>> =>
   DislocatePersons(s, s.actors, persons, camps) and
 p.pstatus = <<ALIVE>> and p.itype = <<NORMAL_INJURY>> =>
   a.action = <<PROVIDE_FIRST_AID>> and
 p.pstatus = <<ALIVE>> and p.itype = <<SEVERE_INJURY>> =>
   exists h in set hospitals & SearchNHospital(hospitals, a) = h &
   MoveToHospital(p, h)and
 p.pstatus = <<DEAD>> => exists h in set hospitals &
   SearchNHospital(hospitals, a) = h & MoveToDeadHouse(p, h) and
 forall ani in set animals & ani.anstatus = <<ALIVE>> =>
   DislocateAnimals(s, s.actors, animals, camps);
```

Pre/post-conditions (1) In the pre-condition, it is assured that the earthquake is reported. (2) Following actions are performed when the earthquake is reported. (i) There exists a gateway node in a subnet which issues order to its neighbour actors to save the precious lives. (ii) For the persons that are alive, actors take action to dislocate them in a camp. (iii) The persons that are alive and have normal injury, actors provide them first aid. (iv) The persons that are alive and receive severe injury, actors search nearest suitable hospital for them and move them to the hospital. (v) The actors transfer the dead bodies of the persons to the nearest hospital dead house. (vi) The animals that are alive are also dislocated by the actors to the location of camps.

The dislocation of persons is specified as a function *DislocatePersons* which takes subnet, set of sensors, actors, people and camps as input and returns true as output if persons are dislocated.

```
values
LIMIT : nat = 5;
DislocatePersons(subnet : Subnet, actors : set of Actor, persons : set of
Person, camps : set of Camp) pdislocated : bool
pre exists c in set camps & c.capacity < LIMIT and
 exists a in set actors & a in set subnet.actors and
 a.acnode.position = subnet.position
post pdislocated <=> exists p in set persons &
 exists c in set camps & p.position = c.position and
 c.clocation = <<INSIDE>> and
 exists a in set actors & a in set subnet.actors and
 a.acnode.position = c.position and a in set actors and
 MoveBack(subnet, actors);
```

Pre/post-conditions (1) In the pre-condition, it is assured that there must exist a camp whose capacity is less than the limit. (2) It is also specified in the pre-condition that there must exist actors in the subnet in which the earthquake is detected. (3) In the post-condition, it is verified that the persons are dislocated if and only if actors locate them inside the camps and after this the actors move back in the subnet.

The *MoveBack* function is specified having two inputs, i.e., *subnet* and *actors* and returns Boolean function true as output if the actors move back in the subnet.

```
MoveBack(subnet : Subnet, actors : set of Actor) mb : bool
pre true
post mb <=> forall a in set actors &
 a.acnode.position = subnet.position;
```

Pre/post-condition The actors of the subnet move back in the subnet if and only if position of the actors becomes equal to the position of the subnet.

The searching of nearest hospital is specified as a function *SearchNHospital* which takes two inputs, i.e., *hospitals* and *actor* and returns the nearest hospital. In the post condition, it is stated that there exists a hospital which is closest to the actor and is selected as the nearest hospital.

```
SearchNHospital(hospitals:set of Hospital, actor:Actor) hospital:Hospital
pre true
post forall h in set hospitals & exists h1 in set hospitals &
 abs actor.acnode.position.xco - abs h1.position.xco <=
 actor.acnode.position.xco - abs h.position.xco and
 abs actor.acnode.position.yco - abs h1.position.yco <=
 actor.acnode.position.yco - abs h.position.yco => hospital = h1;
```

The person is moved to the nearest hospital which is specified as a function *MoveToHospital.* This function takes a person and hospital as input and returns true as output if the person is moved to the hospital. In the post-condition, it is assured that the person is moved to a proper location in the hospital.

```
MoveToHospital(person : Person, hospital : Hospital) moved : bool
pre true
post moved <=> person.position = hospital.position and
hospital.hlocation = <<ROOM>>;
```

If the person dies then he should be moved to the dead house in the hospital. This is specified as a function which takes two inputs, i.e., person and hospital and returns true as output if the person is moved.

```
MoveToDeadHouse(person : Person, hospital : Hospital) moved : bool
pre true
post moved <=> person.position = hospital.position and
hospital.hlocation = <<DEAD_HOUSE>>;
```

The animals that are alive are dislocated to outside of the camps. This is specified as a function which takes four inputs, i.e., subnet, actors, animals and camps and returns Boolean output as true.

```
DislocateAnimals(subnet : Subnet, actors : set of Actor, animals : set of
Animal, camps : set of Camp) adislocated : bool
pre exists a in set actors &
 a in set subnet.actors and a.acnode.position = subnet.position
post adislocated <=> exists a in set animals & exists c in set camps &
 a.position = c.position and c.clocation = <<OUTSIDE>> and
 exists a in set actors & a in set subnet.actors and
 a.acnode.position = c.position and a in set actors and
 MoveBack(subnet, actors);
```

Pre/post-conditions (1) In the pre-condition, it is defined that initially there must exists actors in the subnet in which the earthquake is detected. (2) In the post-condition, it is assured that the animals are dislocated if and only if actors locate them outside the camps and after this the actors move back to the subnet.

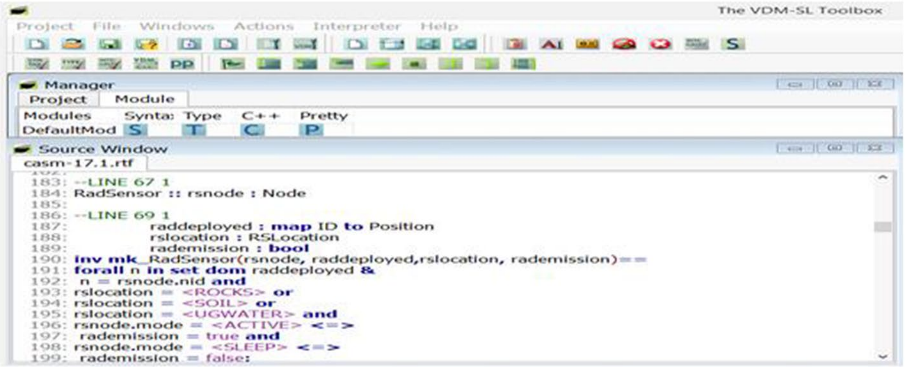
Model analysis

We know that no computer tool can guarantee about 100% correctness of a computer model. The art of writing a formal specification does not guarantee about complete correctness of a model. However, if analysis of the formal specification is done through rigorous computer tools then potential errors can be identified at earlier stages of development which increases a confidence of correctness. Formal specification of the proposed algorithm is defined using VDM-SL which is used to describe complex models both at abstract and detailed level. A model development using VDM-SL provides better understanding and helps in stabilizing the requirements. In the development process of a system, two principles are mainly focussed, i.e., validation and verification. Validation assures that the developed system is according to user requirements while verification guarantees that the developed system in particular phase also includes requirements of the previous stage. VDM-SL helped to identify potential errors at earlier stages, thus generating defects free system.

The developed static and dynamic models are verified by syntax and type checkers in VDM-SL Toolbox. The syntax checker analyses the syntax of the formal specification according to VDM-SL language. The type checker checks misused operators and values which can show run-time errors. Consistency of the formal specification is evaluated by Pretty Printer. The model analysis is provided in Fig. 4. Some errors may remain unidentified therefore dynamic checking was enabled to identify run time errors as shown in Fig. 5. Invariants and pre/post-conditions are defined for the validation of the specification. The analysis of the developed formal specification through integrity examiner is shown in Fig. 6. Integrity examiner analyses the formal specification to generate integrity properties of invariants and pre-post conditions. These properties are then analysed for evaluation of correctness. Validation, animation and testing of the model are done through interpreter and debugger, which increased the confidence of correctness and exhibited that the formal specification reflected the informal requirements.

Related work

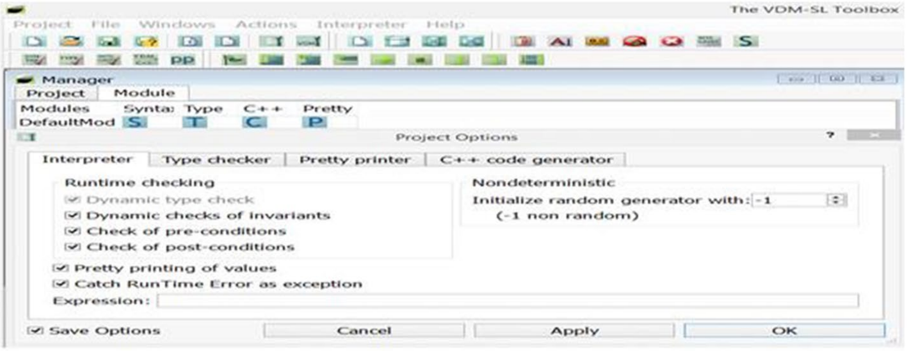
Due to increased damages caused by the natural disasters, focus of researchers is moved towards controlling such damages. An algorithm is developed on the basis of actual seismic hazard information and geographical maps to minimize the total end-to-end disconnection probabilities under certain limitations (Tran and Saito 2016). A method is developed for tracking pedestrians by ad hoc communication techniques for disaster management. In this work, smart-phones are used to store and carry information to a limited number of mobile sources which are responsible for communication (Solmaz and



```
--LINE 67 1
RadSensor :: rsnode : Node

--LINE 69 1
             raddeployed : map ID to Position
             rslocation : RSLocation
             rademission : bool
inv mk_RadSensor(rsnode, raddeployed,rslocation, rademission)==
forall n in set dom raddeployed &
 n = rsnode.nid and
rslocation = <ROCKS> or
rslocation = <SOIL> or
rslocation = <UGWATER> and
rsnode.mode = <ACTIVE> <=>
 rademission = true and
rsnode.mode = <SLEEP> <=>
 rademission = false;
```

Fig. 4 Proof of correctness



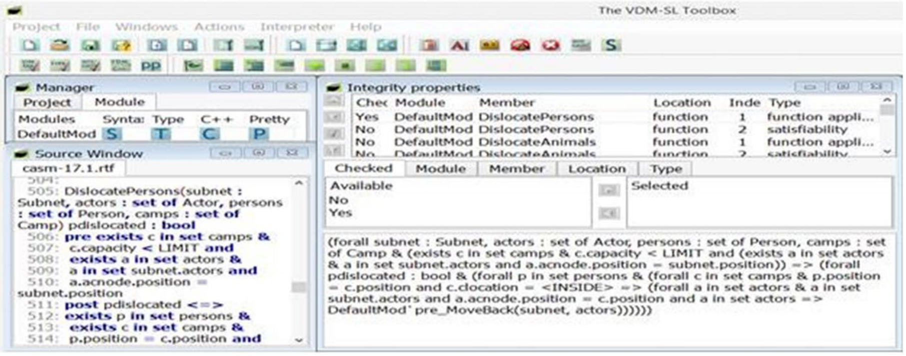
Runtime checking
☑ Dynamic type check
☑ Dynamic checks of invariants
☑ Check of pre-conditions
☑ Check of post-conditions

Nondeterministic
Initialize random generator with: -1
(-1 non random)

☑ Pretty printing of values
☑ Catch RunTime Error as exception
Expression:

☑ Save Options

Fig. 5 Dynamic checking



Modules	Synta:	Type	C++	Pretty
DefaultMod	S	T	C	P

```
DislocatePersons(subnet :
Subnet, actors : set of Actor, persons
: set of Person, camps : set of
Camp) pdislocated : bool
pre exists c in set camps &
 c.capacity < LIMIT and
 exists a in set actors &
 a in set subnet.actors and
 a.acnode.position =
subnet.position
post pdislocated <=>
exists p in set persons &
 exists c in set camps &
 p.position = c.position and
```

Chec	Module	Member	Location	Inde	Type
Yes	DefaultMod	DislocatePersons	function	1	function appli...
No	DefaultMod	DislocatePersons	function	2	satisfiability
No	DefaultMod	DislocateAnimals	function	1	function appli...
No	DefaultMod	DislocateAnimals	function	2	satisfiability

No
Yes

```
(forall subnet : Subnet, actors : set of Actor, persons : set of Person, camps : set
of Camp & (exists c in set camps & c.capacity < LIMIT and (exists a in set actors
& a in set subnet.actors and a.acnode.position = subnet.position)) => (forall
pdislocated : bool & (forall p in set persons & (forall c in set camps & p.position
= c.position and c.clocation = <INSIDE> => (forall a in set actors & a in set
subnet.actors and a.acnode.position = c.position and a in set actors =>
DefaultMod' pre_MoveBack(subnet, actors))))))
```

Fig. 6 Analysis of integrity properties

Turgut 2017). This technique has limited applications which need to be extended toward other similar environments. In another work, applications of WSN and Unmanned Aerial Vehicles (UAV) are realized in the context of natural disaster management (Erdelj et al. 2017). In the same research, a review of most related work is provided addressing the challenges which needs to be addressed. A framework for data delivery in a large-scale network for disaster management is proposed for energy efficiency using Internet of Things (IoT) (Al-Turjman 2017). In this work, it is assumed that resources are limited in terms of hop count and energy levels. In (Kahraman et al. 2016), a local self similarity descriptor is proposed for change detection and deformations in buildings against local and global variations. In this work, the proposed work is evaluated by the 2010 Haiti Earthquake and Typhoon Haiyan 2013 Philippines for detection of collapsed buildings. A recovery process is developed to examine the 2008 Wenchuan earthquake in (Yang and Qi 2017). Change vector analysis technique was applied to measure the post-seismic vegetation recovery conditions. The analysis indicated that the change trend of post-seismic vegetation conditions was grouped into three classes namely recovering, fluctuating, and deteriorating. Internet of Things in terms of web enabling framework and message queue telemetry transport techniques are used to focus earthquakes analysis taking seismic data for an Early Warning System capable of anticipating up to 12 s (Rahman et al. 2016). An analysis of pictures through computer vision is made as application in the area of smart cities and smart homes using the idea of Internet of Things to detect the people (García et al. 2017). In case of a disaster, the base stations may become unreachable, therefore to solve this problem Ad hoc Relay Stations (ARS) are used (Wu et al. 2001). The performance of the proposed algorithm is compared with respect to these three types of cells and is also compared with a hierarchical protocol, LEACH. It is noted that WSNDM protocol performs better than LEACH protocol. DistressNet is an ad hoc wireless architecture proposed for disaster response (George et al. 2010). In this distributed collaborative sensing, multichannel, topology-aware routing protocol is used for accurate localization of resources. To provide collision free transmission of data in an emergency scenario, wireless sensor nodes are integrated with ultrasonic sensor nodes (Erd et al. 2016). The developed architecture is tested for confirming reliability through an experiment by explosion in a building. An overview of disaster management projects using WSN to gather information in disaster areas is provided in (Benkhelifa et al. 2014). For real time volcanic detection, a novel qualities driven approach is presented in (Tan et al. 2010). This approach is based on collaborative algorithms for minimizing consumption of sensor energy. For monitoring earthquake at high precision an earthquake monitoring system using WSN is presented in (Suzuki et al. 2007). In this system, a hard real time Pavenet OS is developed for sensor nodes and acceleration of sensor board. In the project of detection of earthquake, community seismic network (CSN) (Faulkner et al. 2011), android cell phones and a cloud fusion centre are used. The cloud fusion centre helps participants to form "mock earthquakes". When such an event is detected, the cloud fusion centre sends real-time alerts to the cell phones so that people can take precautionary measures. An energy efficient architecture using ad hoc wireless sensor network for locating thousands of people in earthquakes is presented in (Ahmad et al. 2011). Domain ontology is defined for the earthquake scenario by adding domain related concepts to IoT ontology (Spalazzi et al. 2014). Further the authors compare this work with other works in which ontology is used to formally describe things. A system for earthquake automation is proposed in (Aldasouqi and Shaout 2009) based on Ranger Seismometer sensor. For high precision monitoring, Labview application is developed. During earthquakes this system measures vibrations of structures at less cost and higher node density compared to other systems. Sensor Network for Disaster Relief Operations Management (SENDROM) (Cayirci and Coplu 2007) is proposed for the earthquakes detection in Turkey. In this system, sensors are deployed before disaster and central nodes behave as strategic centres and are linked to the database of SENDROM system. Some work exist on modelling of CAS using formal methods however it requires further investigation together with complex networks to model CAS (Batool and Niazi 2017; Niazi 2016).

A framework is formalized unifying behavioral adaptation and structural reconfiguration of components for static detection and reconfiguration of a system, and illustrated by a simple but realistic set of examples (Carlos et al. 2012). A model is developed to support the dynamic reconfiguration of software architectures by taking the advantages of aspect oriented approach for autonomous reconfiguration at run time (Costa 2011). This approach is developed by creating a new kind of aspect in PRISMA in order to provide dynamic reconfiguration of services to complex components of software systems. Policy-based self-adaptive formal model is proposed to develop self-adaptive evolving system using a collection of autonomous actors and managers considering governing and adaptation policies. A combination of two formalisms namely, algebraic and an actor-based model, is used to specify the behaviour (Khakpour et al. 2012). A formal-based method is proposed to model self-organizing systems to meet challenges of existing methods of engineering for modelling reliable complex systems (Bahareh and Saeed 2016). The several challenges include modularity, robustness, decentralized control, flexible and adaptive control mechanism, separation of adaptation, business logic and safe adaptation.

An early earthquake prediction system based on WSNs is designed using information and communication technologies for disseminating earthquake information (Rahman et al. 2016). Unlike this work, our work uses WSANs to predict earthquakes and minimize damages caused by earthquakes. Further, most of the previous work on WSANs is simulations based which have some drawbacks and to overcome these limitations we have used formal methods. Formal model of WSNs with agent-based simulation model is proposed in (Niazi and Hussain 2011), however, this work lacks in describing mathematical model therefore, it is extended through Gaussian function to describe sensing of emergent behaviour in CAS (Chaudhry 2015). Complex network representations of visual-agent and its formal model are developed for cognitive evolution in the form of a temporal cognitive level networks (Hussain and Niazi 2014). Validation of this approach is done through historic data of citations. Some other related work is listed in (Carreras et al. 2017; Lee et al. 2016; Avallone and Banchs 2016; Cheng and Li 2017).

Conclusions

Complex adaptive systems (CASs) are described as dynamic networks which have many agents, acting and reacting in response to each other. The control in CAS is decentralized, therefore it should be coherent and achieved by several decisions accomplished by a large number of agents in competition with each other. In most of the previous works, computer based simulation techniques, i.e., agent-based methodologies and complex networks are used for testing of CAS. Simulation techniques lack in proving complete correctness of a model because the number of test cases increases exponentially to gain a required level of confidence. Formal methods are used to overcome such limitations because these are effective to increase a confidence of correctness of the models. That is why formal specification language, i.e., VDM-SL is used in this work to develop the proposed model. Its correctness is proved by analysing the specification through VDM-SL Toolbox.

WSANs are complex adaptive systems and are modelled as dynamic undirected graphs. Nodes represent sensors, actors and gateways while edges represent communication between them. As the nodes are mobile therefore these are modelled through dynamic graphs. The communication between any two nodes in the network is bidirectional therefore undirected graphs are used because it is easier and economical in terms of time complexity to model undirected graphs. Moreover graph based models are effective to store and process information of any kind of networks. In this work, we have used subnet-based topology in WSAN to localize the problem at subnet level which makes the model energy efficient and reduces the computation at subnet level.

Earthquake management is a big issue across the world and WSANs can be used to mitigate the damages caused by the earthquakes because of their complex adaptive nature. Therefore, Earthquake Disaster Mitigation and Management (EDMM) algorithm using subnet-based WSAN was proposed in this work. In a subnet, sensors detect earthquake and report to actors. Actors report the earthquake information to a gateway in a subnet which decides about the actions to be taken by actors in a subnet. The gateway transmits the earthquake information to a base station through which people were informed about the earthquake using information and communication technologies. Actors in a subnet perform actions like, dislocate people inside the camps dislocate animals outside the camps, provide first aid to people, dislocate severely injured people in hospitals and dead people in dead house. This model was transformed into a formal model using VDM-SL which is used because of its descriptive power and effectiveness for validation, verification, testing and visualization of the specification. The formal specification was described in terms of static and dynamic models. In static model, data types were defined while in dynamic model the whole state space of the system was defined. Invariants were specified on objects to describe their safe behaviour while pre/post conditions were defined in functions and operations to restrict the system for their correct execution. The developed formal specification is analysed through various existing facilities in the VDM-SL Toolbox. Finally, it is stated that, in our knowledge, there does not exist any real work on modelling of earthquake disaster management system using formal methods which shows another contribution in this area.

Abbreviations

WSANs: wireless sensor and actor networks; CAS: complex adaptive systems; VDM-SL: Vienna Development Method-Specification Language; ABM: agent-based methodologies; CN: complex networks; EDMMS: Earthquake Disaster Mitigation and Management; ARS: Ad hoc Relay Stations; LEACH: Low-Energy Adaptive Clustering Hierarchy; QCN: Quake-Catcher Network; MEMS: Micro Electro Mechanical Systems; CSN: community seismic network; IoT: Internet of Things; SENDROM: Sensor Network for Disaster Relief Operations Management.

Authors' contributions

HA and NAZ have proposed the localized and energy-efficient Earthquake Disaster Mitigation and Management (EDMM) model using WSANs. The authors have proposed the approach of subnets in WSAN which is energy efficient as compared to the clustering technique. Graph theory is used to model WSANs topology to use the graph based structures for efficient storage and for processing the network by graph based algorithms. To overcome the disadvantages of simulations authors have transformed the algorithm into an equivalent formal specification using Vienna Development Method Specification Language (VDM-SL) to prove its correctness. They have analysed, validated and verified the developed formal specification through VDM-SL Toolbox. Both authors read and approved the final manuscript.

Authors' information

Ms. Hamra Afzaal is Lecturer at Computer Science Department, COMSATS Sahiwal, Pakistan. Her research interests are formal methods, wireless sensor and actor networks, integration of approaches, etc.

Nazir A. Zafar was born in 1969 in Pakistan. He received his M.Sc. (Math. in 1991), M. Phil (Math. in 1993), and M.Sc. (Nucl. Engg. in 1994) from Quaid-i-Azam University, Pakistan. He earned his PhD degree in computer science from Kyushu University, Japan in 2004. He has served at various universities and well-reputed scientific organizations. For example, he has worked (2010–2014) as Associate Professor at the College of Computer Sciences and Information Technology (CCSIT), King Faisal University (KFU), Al Ahsa, Saudi Arabia. He has also worked (2007–2010) as Dean/Professor of Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan. Currently, he is working as a Professor & Head Academics, COMSATS Sahiwal, Pakistan. His research interest includes modelling of systems using formal approaches, integration of approaches, safety critical systems. He is an active member of Pakistan Mathematical Society. He has contributed for scientific and technical committees including organizing conferences and curriculum development in the capacity of a member as well as chairman.

Competing interests

The authors declare that they have no competing interests.

Funding

The study was not funded.

Publisher's Note

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

Received: 18 July 2017 Accepted: 29 August 2017
Published online: 13 September 2017

References

Afzaal H, Zafar NA (2016a) Towards formalism of earthquake detection and disaster reduction using WSANs. In: 14th IEEE international conference on frontiers of information technology (FIT), pp 147–152

Afzaal H, Zafar NA (2016b) Formal analysis of subnet-based failure recovery algorithm in wireless sensor and actor and network. Complex Adapt Syst Model 4(1):27

Afzaal H, Zafar NA (2016c) Robot-based forest fire detection and extinguishing model. In: 2nd IEEE international conference on robotics and artificial intelligence (ICRAI), pp 112–117

Ahmad N, Riaz N, Hussain M (2011) Ad hoc wirelesses sensor network architecture for disaster survivor detection. Int J Adv Sci Technol 34:16

Akyildiz IF, Kasimoglu IH (2004) Wireless sensor and actor networks: research challenges. Ad Hoc Netw 2(4):351–367

Aldasouqi I, Shaout A (2009) Earthquake monitoring system using ranger seismometer sensor. Int J Geol 3:55–59

Alnuem M, Zafar NA, Imran M, Ullah S, Fayed M (2014) Formal specification and validation of a localized algorithm for segregation of critical/noncritical nodes in MAHSNs. Int J Distrib Sens Netw 10:140973

Altamimi AB, Ramadan RA (2016) Towards internet of things modeling: a gateway approach. Complex Adapt Syst Model 4(1):1–25

Al-Turjman F (2017) Cognitive routing protocol for disaster-inspired internet of things. Future Gener Comput Syst. doi:10.1016/j.future.2017.03.014

Avallone S, Banchs A (2016) A channel assignment and routing algorithm for energy harvesting multiradio wireless mesh networks. IEEE J Sel Areas Commun 34(5):1463–1476

Bahareh A, Saeed J (2016) Towards modelling and runtime verification of self-organizing systems. Expert Syst Appl 44:230–244

Batool K, Niazi MA (2017) Modeling the internet of things: a hybrid modeling approach using complex networks and agent-based models. Complex Adapt Syst Model 5(1):1–4

Benkhelifa I, Nouali-Taboudjemat N, Moussaoui S (2014) Disaster management projects using wireless sensor networks: an overview. In: 28th international conference on advanced information networking and applications workshops (WAINA), pp 605–610

Carlos C, Javier C, Gwen S (2012) Structural reconfiguration of systems under behavioral adaptation. Sci Comput Program 78:46–84

Carreras N, Moure D, Gomáriz S, Mihai D, Mànuel A, Ortiz R (2017) Design of a smart and wireless seismometer for volcanology monitoring. Measurement 97:174–185

Cayirci E, Coplu T (2007) SENDROM: sensor networks for disaster relief operations management. Wireless Netw 13(3):409–423

Chaudhry QA (2015) A Gaussian function model for simulation of complex environmental sensing. Complex Adapt Syst Model 3(1):1–4

Cheng CF, Li LH (2017) Data gathering problem with the data importance consideration in Underwater Wireless Sensor Networks. J Net Comput Appl 78:300–312

Costa SC (2011) Dynamic evolution and reconfiguration of software architectures through aspects. Department of Information System and Computation, University of Politecnica DeValencia, Valencia

Erd M, Schaeffer F, Kostic M, Reindl LM (2016) Event monitoring in emergency scenarios using energy efficient wireless sensor nodes for the disaster information management. Int J Disaster Risk Reduct 16:33–42

Erdelj M, Król M, Natalizio E (2017) Wireless sensor networks and multi-UAV systems for natural disaster management. Comput Netw 124:72–76

Faulkner M, Olson M, Chandy R, Krause J, Chandy KM, Krause A (2011) The next big one: detecting earthquakes and other rare events from community-based sensors. In: 10th international conference on information processing in sensor networks (IPSN), pp 13–24

García CG, Meana-Llorián D, G-Bustelo BCP, Lovelle JMC, Garcia-Fernandez N (2017) Midgar: detection of people through computer vision in the internet of things scenarios to improve the security in smart cities, smart towns, and smart homes. Future Gener Comput Syst 76:301–313

George SM, Zhou W, Chenji H, Won M, Lee Y, Pazarloglou A, Stoleru R, Barooah P (2010) DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response. IEEE Commun Mag 48(3):128–136

Hadj-Kacem N, Kacem AH, Drira K (2009) A formal model of a multi-step coordination protocol for self-adaptive software using coloured petri nets. IJCIS 7(1):25–39

Hussain A, Niazi M (2014) Toward a formal, visual framework of emergent cognitive development of scholars. Cogn Comput 6(1):113–124

Imran M, Zafar NA, Alnuem MA, Aksoy MS, Vasilakos AV (2016) Formal verification and validation of a movement control actor relocation algorithm for safety—critical applications. Wireless Netw 22:247–265

John HH (2006) Studying complex adaptive systems. J Syst Sci Complex 19(1):1–8

Kahraman F, Imamoglu M, Ates HF (2016) Disaster damage assessment of buildings using adaptive self-similarity descriptor. IEEE Geosci Remote Sens Lett 13(8):1188–1192

Khakpour N, Jalili S, Talcott C, Sirjani M, Mousavi M (2012) Formal modelling of evolving self-adaptive systems. Sci Comput Program 78:3–26

Lee EK, Viswanathan H, Pompili D (2016) RescueNet: reinforcement-learning-based communication framework for emergency networking. Comput Netw 98:14–28

Niazi MA (2016) Introduction to the modeling and analysis of complex systems: a review. Complex Adapt Syst Model 4(1):1–3

Niazi M, Hussain A (2011) A novel agent-based simulation framework for sensing in complex adaptive environments. IEEE Sens J 11(2):404–412

Rahman M, Rahman S, Mansoor S, Deep V, Aashkaar M (2016) Implementation of ICT and wireless sensor networks for earthquake alert and disaster management in earthquake prone areas. Procedia Comput Sci 85:92–99

Sarmad S, Alade R, Arshad A, Hafiz FA, Hiroki S (2010) Modelling high assurance agent-based earthquake management system using formal techniques. J Supercomput 52(2):97–118

SCSK Corporation (2013) VDM Tools, User Manual, Version 9.0.2

SCSK Corporation (2013) VDM Tools, Language Manual, Version 9.0.2

Shah MA, Abbas G, Dogar AB, Halim Z (2015) Scaling hierarchical clustering and energy aware routing for sensor networks. Complex Adapt Syst Model (CASM) 3(1):1–23

Solmaz G, Turgut D (2017) Tracking pedestrians and emergent events in disaster areas. J Netw Comput Appl 84:55–67

Spalazzi L, Taccari G, Bernardini A (2014) An internet of things ontology for earthquake emergency evaluation and response. In: IEEE international conference on collaboration technologies and systems, pp 528–534

Suzuki M, Saruwatari S, Kurata N, Morikawa H (2007) A high density earthquake monitoring system using wireless sensor networks. In: International conference on embedded network sensor system

Tan R, Xing G, Chen J, Song WZ, Huang R (2010) Quality driven volcanic earthquake detection using wireless sensor network. In: Proceedings on real time system symposium

Tran PN, Saito H (2016) Enhancing physical network robustness against earthquake disasters with additional links. J Lightwave Technol 34(22):5226–5238

Wu H, Qiao C, De S, Tonguz O (2001) Integrated cellular and ad hoc relaying systems: icar. IEEE J Sel Areas Commun 19(10):2105–2115

Yang W, Qi W (2017) Spatial-temporal dynamic monitoring of vegetation recovery after the wenchuan earthquake. IEEE J Sel Topics Appl Earth Observ Remote Sens 10(3):868–876

Zhozhikashvili AV (2014) Monads for the formalization of a pattern matching procedure. Progr Comput Softw 40(3):117–127