VDM-SL-Based Model of Border Protection System using WSANs

Hamra Afzaal
Department of Computer Science
CIIT, Sahiwal
Pakistan
hamraafzaal@hotmail.com

Tahreem Saeed
Department of Computer Science
CIIT, Sahiwal
Pakistan
Tahreem_saeed93@yahoo.com

Zafar Iqbal
Department of Computer Science
CIIT, Sahiwal
Pakistan
roy_zafar@ciitsahiwal.edu.pk

Nazir Ahmad Zafar
Department of Computer Science
COMSATS, Sahiwal
Pakistan
nazafar@ciitsahiwal.edu.pk

***Abstract*—Border protection requires real-time and mission critical functions with high accuracy. Formal methods are mathematics-based techniques which are used for describing properties of software and hardware increasing quality of computer systems. Formal model of border protection using wireless sensor and actor networks (WSANs) is proposed in this work. WSANs work effectively for such applications to eliminate human intervention. Energy efficiency is a major issue that is why partitioning of WSAN into subnets is assumed which localize the problem increasing efficiency of the algorithm. It is assumed that subnets of WSAN are deployed under and above the ground, and remain continuously connected through radio communication links. Various sensors, actors and gateways are densely deployed in subnets for detecting, tracking and catching the intruders. The model of border protection is formally specified and implemented by Vienna Development Method-Specification Language (VDM-SL) which is used to examine the system at a detailed level and to minimize the defects in a system at early stages. The border protection model is validated and verified using VDM-SL toolbox.**

***Keywords*—Border protection, Graph theory, Wireless sensor and actor networks, Formal methods, VDM-SL**

I. Introduction

Border protection is a critical issue among all the countries due to few obvious and important factors particularly of security issues. The use of information technology is being practiced by most of the countries to take necessary measures to improve security across the border. It is a huge challenge to protect the long stretched border this is because it requires constant human involvement. The conventional border protection systems consist of border troops and checkpoints of security. Each border troop protects specific border section by patrolling according to the pre-specified time and interval. As border protection requires real-time and mission-critical functions with high accuracy, therefore, there is a need to eliminate human involvement by making use of the existing state-of-the-art technologies. Wireless sensor and actor networks (WSANs) are most effective in such applications.

Although many surveillance techniques are proposed for border protection in which human involvement is minimized [1, 2], but are not sufficient to eliminate the challenges faced by existing surveillance techniques. This paper presents a formal model of border protection using WSANs which can detect and track intruders and provide protection from intruders without any human involvement. In the proposed model, partitioning of WSANs into subnets is assumed similar to our previous works [3, 4, 5] which localizes the problem and increases efficiency of the algorithm. Therefore subnet-based graph model is proposed for deployment of WSAN reducing complexity of the model. WSANs are deployed in the form of connected subnets under and above the ground in which sensors are responsible for detection of information and actors are for taking appropriate actions. As border protection is a mission critical system, it requires dense deployment of nodes in all subnets. Dense deployment means the coverage radius of a node is within the next neighbor node [6] and failure of a node does not partitions the network into disjoint segments. The proposed model is presented in Figure 1 showing a logical relationship of the components of the network topology.

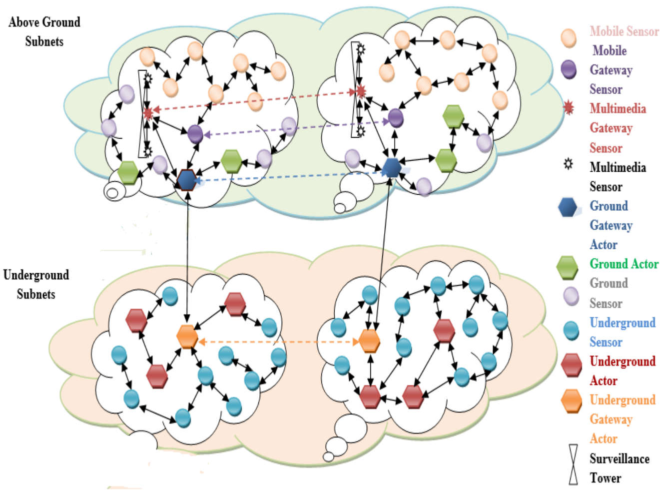

Figure 1: Under and above the ground connected subnets

Graph theory is used to represent topological requirements of a system which are actually informal. Formal methods are used to design formal system by converting requirements of a system to well defined formal specification. Formal methods are mathematical notations used to analyze properties of hardware and software systems [7]. Some researchers have focused on informal techniques, for example, simulations for evaluation of border surveillance techniques [2]. Simulations help to analyze critical system through a quantitative validation of certain properties, but all the properties cannot be validated as it needs extreme knowledge of the system. Formal methods serve as an effective way to analyze systems in detail as well as at an abstract level. Therefore formal methods are more useful for modeling and analysis of mission-critical systems. That is why we have used VDM-SL for the formal specification of the proposed border protection system. In this work, static modeling of border protection is done by defining composite object types by instantiating invariants over it. VDM-SL toolbox confirms the validation and verification of the proposed model. To the best of our knowledge, this is the first work that presents model of dense subnet-based WSANs for border protection system using formal techniques.

Rest of the paper is organized as follows: Section II discusses some related work critically. In Section III, problem statement, system model of the border protection and its algorithm is presented. Section IV presents formal specification of model. Finally conclusion of the paper is given in Section VI.

II. Related work

The conventional border protection systems require constant human involvement. To minimize the cost of increasing personnel on the border unmanned aerial vehicles (UAVs) are introduced for aerial surveillance and for detection of illegal crossing of border [8]. However, UAVs require valuable human resources for its monitoring which should make intelligent decisions and cannot protect whole border all the time like conventional border protection systems. Several techniques for border protection have been proposed by researchers with the objective of minimizing the human involvement. For example, the border patrol system proposed in [1] uses WSNs to minimize extreme human involvement and improves the intruder detection accuracy. Unlike this, we use WSANs with VDM-SL to eliminate human involvement providing more accuracy in monitoring. Many applications are developed based on WSNs for the border surveillance. For example, nodes of the network are deployed as line-sensors in [9] where every movement is detected that goes over sensors barriers. It is claimed that the sensor nodes deployment assures barrier coverage. The barrier coverage requires less number of nodes as compared to full coverage but it may suffer disconnection of radio links due to failure of sensor nodes. To attain continuous barrier coverage for the whole area, the segmentation technique is presented in [10].

Most of the research in this area is based on informal techniques. For example, simulation of border surveillance model using wireless sensor networks is done in [2]. Some other researchers have focused on the simulation of proposed algorithms using wireless sensor and actor networks [11, 12]. Formal methods are mathematical procedures used for the validation and verification of safety-critical [13] and mission-critical systems. Formal methods provide defects free systems and are more effective than simulations. It is noted that simulations results can be improved if the proposed model is formally verified at first. In [14-18], the authors focus on both formal and informal techniques for the verification and validation of the proposed algorithms for WSANs.

III. System Model and Algorithm

Border protection is a real-time and mission-critical issue. The constant human involvement cannot provide more accuracy that neccesisates the use of wireless sensor and actor networks which are more suitable for such applications. In this work, we have used WSANs in the form of connected subnets which are deployed under and above the ground. The partitioning of network into subnets localizes the problem and increases efficiency of the algorithm. The nodes in subnets are densly deployed which means the coverage radius of a node is up to the next neighbor node [6]. In dense deployment, failure of a node does not partition the subnet into disjoint segments. The subnets consist of sensors, gateways, actors, multimedia sensors, multimedia gateway sensors, mobile sensors and mobile gateway sensors [1]. The subnets remain continuously connected as gateway nodes use radios for continuous communication. The underground subnet nodes have low cost and short range radio frequency for intrusion detection while above ground nodes have high cost and long range radio frequency. The actors are used for providing protection from intruders by taking appropriate decisions.

```
BorderProtection()
1.  Deployment of WSAN Subnets
        Underground subnets
        Aboveground subnets
2.  Vibration caused by intrusion is detected by underground
    sensors
3.  Detected vibration is communicated with underground
    gateway actor
4.  Underground gateway actor activates underground actors
    and also communicate with ground gateway actor to
    confirm detected vibration
5.  Ground gateway actor communicate with ground sensors
    to confirm vibration
6.  Ground sensors confirm the detection of vibration and
    send information to ground gateway actor
7.  Ground gateway actor communicate the detected
    information with multimedia gateway sensor
8.  Multimedia gateway sensor communicate with nearby
    multimedia sensors to take image or video of intruder
9.  Multimedia sensors take image or video of intruder and
    send it to multimedia gateway sensor
10. Multimedia gateway sensor communicate the information
    with mobile gateway sensor which communicates with
    nearby mobile sensors to track intruder
11. Mobile sensors after tracking intruder send information to
    mobile gateway sensor
12. Mobile gateway sensor send the tracking information to
    ground gateway actor
13. Ground gateway actor activates ground actors to catch the
    intruder and fight with it
```

Figure 2: Pseudo code of the algorithm

The underground sensors detect vibration for possible intrusion and send it to underground gateways. The gateway activates underground actors for the required action and send the information to above the ground gateway which communicates with ground sensors to verify the intrusion. The ground sensors confirm detection of vibration and communicate information with ground gateways. The gateway shares this information with the multimedia gateway sensors deployed on surveillance tower, which sends information to the nearby multimedia sensors for taking image or video of intruders. The multimedia sensors send this information to multimedia gateway sensor which communicates with the mobile gateway sensor which shares this information with the nearby mobile sensors to track the intruders. The mobile sensors send tracked information to mobile gateway sensors which communicate intruder's information with ground gateway actor that are actually mobile robots for catching the intruder and fighting with it. Pseudo code of the algorithm is given in Figure 2.

IV. FORMAL SPECIFICATION USING VDM-SL

In this section, formal specification of the model is provided using VDM-SL. Communication links exists between all nodes which show the network is connected. Communication links and coverage are specified by composite objects, *Link* and *Coverage* respectively. WSAN is presented as a composite object, *DenseGraph*, which consists of three fields. The first and second fields, *nodes* and *commlinks* represent nodes and communication links. The third field, *node_coverage*, is used to represent dense of the graph.

```
types
Node = token;
Link:: nd1:Node
  nd2: Node
inv mk_Link (nd1, nd2) == nd1 <> nd2;
Coverage:: min_range: nat
  max_range: nat;
DenseGraph:: nodes: set of Node
  commlinks: set of Link
  node_coverage: set of Coverage
inv mk_DenseGraph (nodes, commlinks,node_coverage) ==
forall cl in set commlinks & cl.nd1 in set nodes and
cl.nd2 in set nodes and
forall n in set nodes & (exists cl in set commlinks &
  (n=cl.nd1 and n=cl.nd2)) and
forall n1,n2 in set node_coverage &
(n1.max_range>=n2.min_range);
```

Invariants: (1) Every communication link consists of two nodes. (2) For all the nodes in a dense graph, there is a communication link which consists of two nodes that must be unique. (3) The both nodes in the communication link should be within the maximum range.

Two types of subnets, i.e., under and above the ground are used in the network. The underground subnet is specified by a composite object, *Underground_Subnet* which employs six fields. The first field *subnets* is used to describe underground subnet as a dense graph. The second and third fields, *nodes* and *commlinks* are used for nodes and communication links. The fourth, fifth and sixth fields, i.e., *ugs*, *ugga* and *uga* describe collection of underground sensors, gateways and actors.

```
Uderground_Subnet:: subnets: set of DenseGraph
  nodes: set of Node
  commlinks: set of Link
  ugs: set of UnderGroundSensor
  ugga: set of UnderGroundGatewayActor
  uga: set of UnderGroundActor
inv mk_Uderground_Subnet(subnets, nodes, commlinks,ugs,
ugga,uga) ==
forall sbn in set subnets & card sbn.nodes >= 3 and
forall node in set nodes & exists us in set ugs &
  exists ug in set ugga & exists ua in set uga &
  ({node} = us.nodes and {node} = ug.nodes and
  {node} = ua.nodes) and
forall us1,us2 in set ugs & (exists cl in set commlinks &
  (us1.ugs_idt=cl.nd1 and  us2.ugs_idt=cl.nd2)) and
forall us in set ugs & (forall ug in set ugga &
  (exists cl in set commlinks & (us.ugs_idt=cl.nd1 and
ug.ugga_idt=cl.nd2))) and forall ug1,ug2 in set ugga &
  (exists cl in set commlinks & (ug1.ugga_idt=cl.nd1 and
ug2.ugga_idt=cl.nd2)) and forall ugg in set ugga &
  (forall ug in set uga & (exists cl in set commlinks &
  (ugg.ugga_idt = cl.nd1 and ug.uga_idt=cl.nd2))) and
forall ug1,ug2 in set uga & (exists cl in set commlinks &
  (ug1.uga_idt = cl.nd1 and ug2.uga_idt = cl.nd2));
```

Invariants: (1) An underground subnet consists of at least three nodes. (2) An underground subnet employs sensors, gateways and actors. (3) There exists a communication link between the sensors. (4) There exists communication link between sensors and gateway actors. (5) Underground gateway actors communicate with each other in different subnets so there must be a communication link between them. (6) The underground gateways and actors communicate with each other so there is a communication link between them.

As all the nodes transmit and receive information so it is specified by a composite object *Communicate*. The nodes communicate intruder information so it is described by a composite object *CommIntruderInfo*.

```
Communicate::
  transmit: Transmit
  receive: Receive;
CommIntruderInfo::
  transmit: Transmit
  receive: Receive;
```

The underground sensors are represented as composite object *UndergroundSensor*. The first field in the object describes that all the nodes have unique identifiers. The second field *nodes*, is used as these are assumed as nodes. The third and fourth fields, *comm* and *commII* are used for communication with each other and with the intruder's information. The fifth field *cost* expresses that the nodes have certain cost. The sixth field *radiofrequency* specifies that these nodes communicate through certain radio frequency. The seventh field *visibility* represents the visibility status and the eighth one *vibration* describes that these nodes are responsible for detecting and communicating vibration sensed under the ground. Similarly the underground gateways and actors can be defined.

```
Vibration = <YES>|<NO>; Cost=<HIGH>|<LOW>;
Range = <LONG>|<SHORT>;
UnderGroundSensor::  ugs_idt: Node
  nodes: set of Node
  comm: Communicate
  commII:CommIntruderInfo
```

cost:*Cost*
radiofrequency:*Range*
visibility:**bool**
vibration:*Vibration*
inv mk_*UnderGroundSensor* (*ugs_idt*, *nodes*,-,-, *cost*, *radiofrequency*, *visibility*,-) == {*ugs_idt*} = *nodes* **and** *cost* = <LOW> **and** *radiofrequency* =<SHORT> **and** *visibility* = **false**;

Invariants: (1) All the underground sensors are nodes in the subnet. (2) The nodes have low cost, communicate using short range radio frequency and are not visible to the intruders.

The above ground subnet is described by a composite object *AboveGround_Subnet* which consists of ten fields. The first field *subnets* describes that it is the dense graph. The second and third fields, *nodes* and *commlinks*, explain that subnet nodes exist and there exists communication links between them above the ground. The remaining seven fields, i.e., *gga*, *ga*, *gs*, *mmgs*, *mms*, *mgs* and *ms* illustrate set of gateways, set of actors, set of sensors, multimedia gateway sensors, multimedia sensors, mobile gateway sensors and mobile sensors.

AboveGround_Subnet:: *subnets*: **set of** *DenseGraph*
nodes: **set of** *Node*
commlinks: **set of** *Link*
gga: **set of** *GroundGatewayActor*
ga: **set of** *GroundActor*
gs: **set of** *GroundSensor*
mmgs: **set of** *MultimediaGatewaySensor*
mms: **set of** *MultimediaSensor*
mgs: **set of** *MobileGatewaySensor*
ms: **set of** *MobileSensor*
inv mk_*AboveGround_Subnet*(*subnets*, *nodes*, *commlinks*, *gga*,*ga*,*gs*,*mmgs*,*mms*,*mgs*,*ms*)== **forall** *sbn* **in set** *subnets* &
(**card** *sbn*.*nodes*>=7) **and forall** *node* **in set** *nodes* &
(**exists** *ag* **in set** *gga* & (**exists** *aa* **in set** *ga* &
(**exists** *ass* **in set** *gs* & (**exists** *ammg* **in set** *mmgs* &
(**exists** *ams* **in set** *mms* & (**exists** *amgs* **in set** *mgs* &
(**exists** *amos* **in set** *ms* & ({*node*}=*ag*.*nodes* **and**
{*node*}=*aa*.*nodes* **and**{*node*}=*ass*.*nodes* **and**
{*node*}=*ammg*.*nodes* **and**{*node*}=*ams*.*nodes* **and**
{*node*}=*amgs*.*nodes* **and**{*node*}=*amos*.*nodes*)))))))) **and**
forall *ag1*,*ag2* **in set** *gga* & (**exists** *cl* **in set** *commlinks* &
(*ag1*.*gga_idt*= *cl*.*nd1* **and** *ag2*.*gga_idt* =*cl*.*nd2*)) **and**
forall *ag* **in set** *gga* & (**exists** *ga1* **in set** *ga* &
(**exists** *cl* **in set** *commlinks* & (*ag*.*gga_idt*= *cl*.*nd1* **and**
ga1.*ga_idt* =*cl*.*nd2*))) **and forall** *ag1*,*ag2* **in set** *ga* &
(**exists** *cl* **in set** *commlinks* & (*ag1*.*ga_idt*= *cl*.*nd1* **and**
ag2.*ga_idt* =*cl*.*nd2*)) **and forall** *ag* **in set** *ga* &
(**exists** *gs1* **in set** *gs* &(**exists** *cl* **in set** *commlinks* &
(*ag*.*ga_idt*= *cl*.*nd1* **and** *gs1*.*gs_idt* =*cl*.*nd2*))) **and**
forall *gs1*,*gs2* **in set** *gs* & (**exists** *cl* **in set** *commlinks* &
(*gs1*.*gs_idt*= *cl*.*nd1* **and** *gs2*.*gs_idt* =*cl*.*nd2*)) **and**
forall *aga* **in set** *gga* & (**exists** *mgs1* **in set** *mmgs* &
(**exists** *cl* **in set** *commlinks* & (*aga*.*gga_idt*= *cl*.*nd1* **and**
mgs1.*mmgs_idt* =*cl*.*nd2*))) **and forall** *mmgs1*,*mmgs2* **in set**
mmgs & (**exists** *cl* **in set** *commlinks* & (*mmgs1*.*mmgs_idt*=
cl.*nd1* **and** *mmgs2*.*mmgs_idt* =*cl*.*nd2*)) **and**
forall *amgs* **in set** *mmgs* & (**exists** *mgs* **in set** *mgs* &
(**exists** *cl* **in set** *commlinks* & (*amgs*.*mmgs_idt*= *cl*.*nd1* **and**
mgs.*mgs_idt* =*cl*.*nd2*))) **and forall** *mgs1*,*mgs2* **in set** *mgs* &
(**exists** *cl* **in set** *commlinks* & (*mgs1*.*mgs_idt*= *cl*.*nd1* **and**
mgs2.*mgs_idt* =*cl*.*nd2*)) **and forall** *amgs* **in set** *mgs* &
(**exists** *ams* **in set** *ms* &(**exists** *cl* **in set** *commlinks* &
(*amgs*.*mgs_idt*= *cl*.*nd1* **and** *ams*.*ms_idt* =*cl*.*nd2*))) **and**
forall *ms1*,*ms2* **in set** *ms* & (**exists** *cl* **in set** *commlinks* &
(*ms1*.*ms_idt*= *cl*.*nd1* **and** *ms2*.*ms_idt* =*cl*.*nd2*)) **and**
forall *amgs* **in set** *mgs* & (**exists** *aga* **in set** *gga* &
(**exists** *cl* **in set** *commlinks* & (*amgs*.*mgs_idt*= *cl*.*nd1*
and *aga*.*gga_idt* =*cl*.*nd2*)));

All kinds of sensors, actors and gateways are connected and have a path through communication links with each other.

The communication under and above ground subnets is expressed by *Under_Above_Ground_Subnet_Comm* defined by four fields. The first field *nodes* shows the communication between subnets through nodes. The second field *commlinks* expresses that there exists communication links between the nodes. The third and fourth fields, i.e., *ugga* and *gga* are the collection of under and above the ground gateway actors. The underground gateway actor exists in underground subnet and ground gateway actor exists above the ground subnet having some information of the subnet.

Under_Above_Ground_Subnet_Comm::
nodes: **set of** *Node*
commlinks: **set of** *Link*
ugga: **set of***UnderGroundGatewayActor*
gga: **set of** *GroundGatewayActor*
inv mk_*Under_Above_Ground_Subnet_Comm* (-,*commlinks*, *ugga*, *gga*) == **forall** *u* **in set** *ugga* & (**exists** *a* **in set** *gga* &
(**exists** *cl* **in set** *commlinks* & (*u*.*ugga_idt*= *cl*.*nd1* **and** *a*.*gga_idt* =*cl*.*nd2*)));

The ground sensor is specified by *GroundSensor* consisting of six fields. The first field *gs_idt* represents that ground sensors are unique. The one, *nodes*, explains that the nodes are in the subnet. The third one *cost* illustrates that the nodes have certain cost. The *radiofrequency* shows that the nodes communicate using their radio frequency. The *visibility* is used to check the visibility status and *vibration* is used to detect the vibration.

GroundSensor::
gs_idt: *Node*
nodes: **set of** *Node*
cost: *Cost*
radiofrequency: *Range*
visibility: **bool**
vibration: *Vibration*
inv mk_*GroundSensor*(*gs_idt*, *nodes*,*cost*, *radiofrequency*, *visibility*,-)== {*gs_idt*} = *nodes* **and** *cost*=<LOW> **and** *radiofrequency* = <SHORT> **and** *visibility* = **true**;

The gateway and actor are specified as composite objects, i.e., *GroundGatewayActor* and *Ground Actor* having nine fields. The first field is used to illustrate that these actors have unique identifiers. The *nodes* describes nodes above the ground subnet. The *mobilerobots* represents that these actors are the mobile robots. The fourth and fifth fields, *comm* and *commII* explain that the nodes communicate with each other. The sixth field *cost* represents that nodes have certain cost. The seventh field *radiofrequency* shows that the actors use radio frequency for communication. The eighth field *visibility* is used to check the visibility status and *vibration* is used to detect vibration.

GroundGatewayActor::*gga_idt*:*Node*
nodes:**set of** *Node*
mobilerobot: *MobileRobot*
comm: *Communicate*
commII: *CommIntruderInfo*
cost:*Cost*
radiofrequency:*Range*
visibility:**bool**
vibration: *Vibration*
inv mk_*GroundGatewayActor* (*gga_idt*, *nodes*,-,-,-,*cost*, *radiofrequency*, *visibility*,-) == {*gga_idt*}=*nodes* **and** *cost*= <HIGH> **and** *radiofrequency* =<LONG> **and** *visibility*=**true**;
GroundActor:: *gga_idt*: *Node*
nodes:**set of** *Node*
mobilerobot: *MobileRobot*
comm: *Communicate*
commII: *CommIntruderInfo*
cost:*Cost*
radiofrequency:*Range*
visibility:**bool**
vibration: *Vibration*
inv mk_*GroundGatewayActor* (*ga_idt*, *nodes*,-,-,-,*cost*, *radiofrequency*, *visibility*,-) ==
{*ga_idt*}=*nodes* **and** *cost*= <HIGH> **and**
radiofrequency = <LONG> **and** *visibility*=**true**;

Invariants: (1) Ground actors are nodes in above the ground subnet. (2) Ground actors have high cost, long range radio frequency for communication and are visible to intruders.

The multimedia gateway and sensor are represented by *MultimediaGatewaySensor* and *MultimediaSensor* having nine fields. All the fields are described previously except eighth and ninth fields. The eighth field *capture* is required as sensors are used to capture image or video of an intruder. The ninth field *deployed* describes that nodes are deployed on surveillance tower which is represented *SurveillanceTower*. The towers are deployed in a way that these are within coverage of each other.

Signals = **token**;
Transmit = **token**;
Receive = **token**;
Capture=<IMAGE>|<VIDEO>;
SurveillanceTower::*signals*:*Signals*
transmit:*Transmit*
receive:*Receive*
tower_signals_coverage:**set of** *Coverage*
inv mk_*SurveillanceTower*(-,-,-, *tower_signals_coverage*) ==
forall *tsc1*,*tsc2* **in set** *tower_signals_coverage* &
tsc1.max_range> = *tsc2.min_range*;
MultimediaGatewaySensor::*mmgs_idt*:*Node*
nodes:**set of** *Node*
comm: *Communicate*
commII: *CommIntruderInfo*
cost:*Cost*
radiofrequency:*Range*
visibility:**bool**
capture:*Capture*
deployed:*SurveillanceTower*
inv mk_*MultimediaGatewaySensor*(*mmgs_idt*, *nodes*,-,-,*cost*, *radiofrequency*, *visibility*, *capture*,-)==
{*mmgs_idt*} = *nodes* **and** *cost*=<HIGH> **and**
radiofrequency = <LONG> **and** *visibility* = **true and**
capture = <IMAGE> **or** *capture* = <VIDEO>;

Invariants: (1) Multimedia gateway sensors are nodes in above ground subnet. (2) Multimedia gateway sensors have high cost, have long range radio frequency for communication, visible to intruder and are used to capture image or video.

MultimediaSensor::*mms_idt*:*Node*
nodes:**set of** *Node*
comm: *Communicate*
commII: *CommIntruderInfo*
cost: *Cost*
radiofrequency:*Range*
visibility: **bool**
capture:*Capture* *deployed*: *SurveillanceTower*
inv mk_*MultimediaSensor*(*mms_idt*, *nodes*,-,-,*cost*, *radiofrequency*, *visibility*, *capture*,-)==
{*mms_idt*}=*nodes* **and** *cost* = <HIGH> **and**
radiofrequency = <LONG> **and** *visibility* = **true and**
capture = <IMAGE> **or** *capture*=<VIDEO>;

Invariants: (1) Multimedia sensors are nodes in the above ground subnet. (2) Multimedia sensors have high cost, have long range radio frequency for communication, visible to intruder and are used to capture image or video.

The mobile gateway sensor and mobile sensor are specified by composite objects *MobileGatewaySensor* and *MobileSensor* and are defined by nine fields. All fields are described except *deploy* which shows the nodes are deployed in air.

Air = **token**;
MobileGatewaySensor:: *mgs_idt*:*Node*
nodes:**set of** *Node* *comm*: *Communicate*
commII: *CommIntruderInfo* *cost*: *Cost*
radiofrequency: *Range*
visibility: **bool**
track: *Intruder; deploy*: *Air*
inv mk_*MobileGatewaySensor* (*mgs_idt*, *nodes*,-,-,*cost*, *radiofrequency*, *visibility*,-,-) ==
{*mgs_idt*} = *nodes* **and** *cost* = <HIGH> **and**
radiofrequency = <LONG> **and** *visibility* = **true**;

Invariants: (1) Mobile gateway sensors are nodes in above the ground subnet. (2) Mobile gateway sensors have high cost, long range radio frequency and are visible to intruder.

MobileSensor:: *ms_idt*: *Node*
nodes:**set of** *Node*
comm: *Communicate*
commII: *CommIntruderInfo*
cost: *Cost*
radiofrequency: *Range*
visibility: **bool**
track: *Intruder; deploy*: *Air*
inv mk_*MobileSensor* (*ms_idt*, *nodes*,-,-,*cost*, *radiofrequency*, *visibility*,-,-) == {*ms_idt*}= *nodes* **and** *cost* = <HIGH> **and**
radiofrequency = <LONG> **and** *visibility* = **true**;

As mobile robot takes suitable action to protect the border, it is defined by a composite object *MobileRobot* having four fields. The first field *name* describes name of the mobile robot. The second one *operation* represents that the mobile robots either work or in idle state. The *location* exhibits intrusion detection location and *action* describes an action of the mobile robot.

Operation = <WORK>|<IDLE>;
Location =
<INTRUSION_DETECTED>|<INTRUSION_NOT_DETECTED>;
Action = <CATCH_INTRUDER>|<FIGHT>;
MobileRobot:: *name*: *String*
operation:*Operation*
location: *Location*
action:*Action*
inv mk_*MobileRobot*(-,*operation*,*location*,*action*) ==
operation=<WORK> =>*location* =<INTRUSION_DETECTED>**and**
action=<<CATCH_INTRUDER>> **and** *action* =<FIGHT>
and *operation*=<IDLE> => *location*=
<INTRUSION_NOT_DETECTED>;

Invariants: (1) Robots work in location where the intrusion is detected. The robots catch the intruder and fight with it. (2) The robot will be idle if the intrusion is not detected.

The intruder is defined by a composite object *Intruder* with fields, i.e., *name*, *destroy* and *info*. The first field describes that the intruder has certain name. The second field explains the intruder can destroy multimedia and ground sensor. The third field is used for information related to the intruder.

String = **seq of char**;
Destroy::*ms*: **set of** *MultimediaSensor*
gs: **set of** *GroundSensor*;
Intruder:: *name*: *String*
destroy: *Destroy*
info:*Communicate*;

V. CONCLUSION

Border protection is highly a critical issue for all the countries. That is why a model of border protection using wireless sensor and actor networks is presented in this paper to detect, track and catch the intruders, terrorists, or any other doubtful activities on the border. There exists some work on surveillance techniques to minimize the human intervention using wireless sensor networks but still it needs to explore this research area as this is an open research problem. In this research article, WSANs are integrated with formal techniques to eliminate human intervention for achieving high accuracy in this real-time and mission-critical system. WSANs are deployed under and above the ground in the form of subnets increasing efficiency of the algorithm because of localizing the problem. The subnets are modeled using graph theory which gives semi-formal representation of the system. Formal specification language, VDM-SL, was used to transform the model to formal system providing a detailed specification. It is mentioned that VDM-SL was used because of its easier implementation to any of the programming languages and having various tools for analysis of the model. Further, it was investigated that there is a good relationship between graph theory and VDM-SL which was realized in terms of defining rules among both of the approaches. VDM-SL toolbox was used to validate and verify the border protection model. The specification of the model is analyzed using toolbox which showed that there is no syntax or type error in the specification. Further, no warning was found which gives proof of correctness of the specification.

REFERENCES

[1] Z. Sun, P. Wang, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan, and I. F. Akyildiz, "BorderSense: Border patrol through advanced wireless sensor networks," Ad Hoc Networks, 9(3), 468-477, 2011.

[2] M. Alkhathami, L. Alazzawi, and A. Elkateeb, "Border Surveillance and Intrusion Detection Using Wireless Sensor Networks," International Jouranl of Advances in Engineering & Technology, 2015.

[3] H. Afzaal, and N.A. Zafar, "Formal Modeling and Algorithm of Subnet-based Backup Assigning in WSAN," 6th IEEE International Conference on Information and Communication Technologies (ICICT), 2015.

[4] H. Afzaal, M. Imran, and N. A. Zafar, "Implementing Partitioning Detection and Connectivity Restoration in WSAN using VDM-SL," 13th IEEE International Conference on Frontiers of Information Technology (FIT), pp. 71-76, 2015.

[5] H. Afzaal, and N.A. Zafar, "Algorithm and Formal Specification of Subnet-Based Communication in WSANs, " 1st International Conference on Dependable Embedded Wireless and Sensing networks, 2015.

[6] Y. Kim, "A study on operational procedures and deployment method of surveillance reconnaissance sensor network system," Agency for Defense Development ADDR-425-111039, 2011.

[7] M. Conrad, and D. Hötzer, "Selective integration of formal methods in the development of electronic control units," Proceedings of IEEE 2nd International Conference on Formal Engineering Methods, 1998.

[8] C. C. Haddal, and J. Gertler, "Homeland security: Unmanned aerial vehicles and border surveillance," library of congress washington dc congressional research service, 2010, July.

[9] R. Bellazreg, N. Boudriga, and S. An, "Border Surveillance using sensor based thick-lines," In Information Networking (ICOIN), 2013 International Conference on (pp. 221-226). IEEE.

[10] B. K. Maharrey, A. S. Lim, and S. Gao, "Interconnection between IP networks and wireless sensor networks," International Journal of Distributed Sensor Networks, 2012.

[11] M. Imran, M. Younis, Haider, N., and M. A. Alnuem, "Resource efficient connectivity restoration algorithm for mobile sensor/actor networks," EURASIP Journal on Wireless Communications and Networking, (1), pp. 1-16, 2012.

[12] M. Imran, N. Haider, and M. Alnuem, "Efficient movement control actor relocation for honing connected coverage in wireless sensor and actor networks," Local Computer Networks Workshops, pp. 710-17, 2012.

[13] N. A. Zafar, S. A. Khan, and K. Araki, "Towards the safety properties of moving block railway interlocking system," Int. J. Innovative Comput., Info & Control, 8(7), 5677-5690, 2012.

[14] M. Imran, N. A. Zafar, M. A. Alnuem, M. S. Aksoy, and A. V. Vasilakos, "Formal verification and validation of a movement control actor relocation algorithm for safety–critical applications," Wireless Networks, pp. 1-19, 2015.

[15] M. Imran, and N.A. Zafar, "Formal specification and validation of a hybrid connectivity restoration algorithm for wireless sensor and actor networks," Sensors, **12**(9): pp. 11754-81, 2012.

[14] S. Riaz, H. Afzaal, M. Imran, N. A. Zafar, and M. S. Aksoy, "Formalizing mobile ad hoc and sensor networks Using VDM-SL," Procedia Computer Science, 63: pp. 148-153, 2015.

[17] H. Afzaal, and N.A. Zafar, "Formal localized reactive subnet-based failure recovery model for sparsely connected wireless sensor and actor networks, 9th IEEE International Conference on Open Source Systems and Technologies, (ICOSST), pp.64-71, 2015.

[18] H. Afzaal, and N.A. Zafar, "Centralized confidentiality-based formal algorithm in WSANs," 1st National Conference on Trends and Innovations in Information Technology (TIIT), 2016.