Developing a Simulation for Border Safeguarding

Seanette van Rooyen
and
Louise Leenen

Abstract

Border safeguarding is the defence of territorial integrity and sovereignty, and this is a joint responsibility of the military and the police. Military doctrine for conventional warfare is not sufficient for Border Safeguarding operations due to the nature of the adversaries and their activities.

A perfect arena to play out Border safeguarding scenarios is a simulation environment. This paper discusses a project to develop an agent-based simulator supported by intelligent threat behaviour algorithms. Threats in the Border safeguarding context evade blue forces, hide out of line of sight, and travel at night and off main roads. The authors motivate the use of Agent Based Modelling and Constraint Programming techniques to support scenarios where new rules of engagement for Border safeguarding can be tested.

1. Introduction

The main objective of this study is to assist with homeland defence in joint Operations Other Than War (OOTW). Border safeguarding provides a platform to develop doctrine through simulation. Defence against aggression is replaced by arrests, and target damage assessment is replaced by the number of intercepts without kills. This limits the study to ground targets, often unarmed and with low hostility levels.

Border safeguarding is a Joint Command and Control (JC2) operation between the South African National Defence Force (SANDF) and South African Police Service (SAPS). It deals with situations such as rhino poaching, smuggling of narcotics into South Africa, and transporting stolen goods and vehicles out and across the border. In Section 2 we discuss the JC2 concept as well as the role of the Interoperability Development Environment (IDE) at the Council for Industrial and Scientific Research (CSIR) as described by Duvenhage [1].

In Section 3 we introduce our approach by motivating the use of simulation with an agent-based design and the use of constraint programming techniques. Current simulators do not provide agile and intelligent threat behaviour that realistically represents patterns of avoiding blue forces. Our approach addresses this shortcoming.

Section 4 briefly discusses preliminary findings without the use of intelligent threat behaviour. Section 5 contains our conclusions.

2. Joint Control and Command

Figure 1 shows how JC2 operate and communicate between the levels of war as defined in [2]. The levels signify Strategic goals, Operational objectives, and Tactical actions effected by Weapon Systems as shown in Vertical Interoperability rows.

Service divisions in the Horizontal Interoperability sphere operate and communicate between the silos of Joint Air, Land, Maritime and Information Warfare (IW) Capabilities as shown by ovals in the Tactical level. Shared processes in the Vertical Interoperability sphere are Situation Awareness (the result of In-Simulation training, planning, and rehearsal), Planning, Tasking and Control.

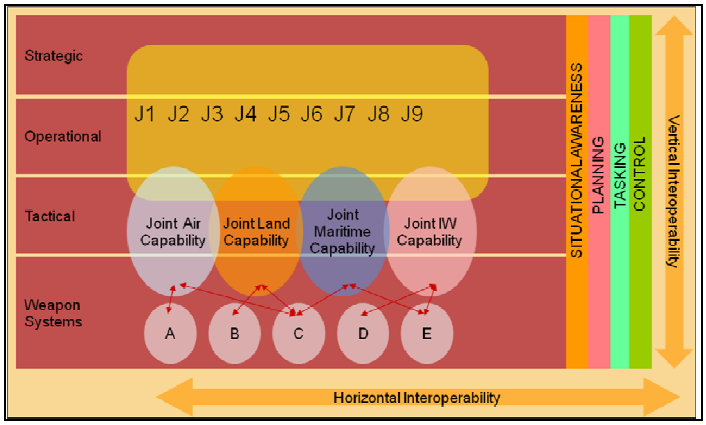

Figure 1: Figure 1 JC2 interoperability[1]

[1] **JOINT FUNCTIONS**
J1 Personnel
J2 Intelligence
J3 Current Operations (Including Joint Operations)
J4 Logistic Support
J5 Future Operations (Planning, including Joint Operations)
J6 Communications and Information Systems
J7 Training and Exercises
J8 Civil Military Cooperation (Interdepartmental Operations)
J9 Host Nation Support (Multinational Operations)

Significant work has been done in developing the Virtual Ground Based Air Defence System Demonstrator (VGD) in the Interoperability Development Environment (IDE) at the Council for Industrial and Scientific Research (CSIR) [1]. The IDE centre serves as a national test, training and doctrine development facility. It provides threats and blue force agents to the VGD simulation in order to create a realistic picture of the border safeguarding situation. The IDE attempts to integrate across the evolution of technology (i.e. radar, Air Picture systems, simulators, tactical radio communication, Blue Force Tracking, GPS and cell phones). Credible blue and red forces information and communication are received via the IDE. Valuable experience is gained and valid data is available from the IDE and the VGD which will be used in our simulation.

Figure 2 illustrates how the IDE Gateway connects to real-time live data sources such as Local Warning Radars, air traffic control centres, simulators and displays. The IDE Gateway has the capability to link to various protocols and systems including blue force tracking, tactical radios using LinkZA for live systems, operations planning software, simulators such as the VGD, flight simulators, a radar system, Threat Evaluation Weapons Assignment (TEWA) software, the Air Traffic Navigation System (ATNS) and the Air Picture Display System (APDS). [1]

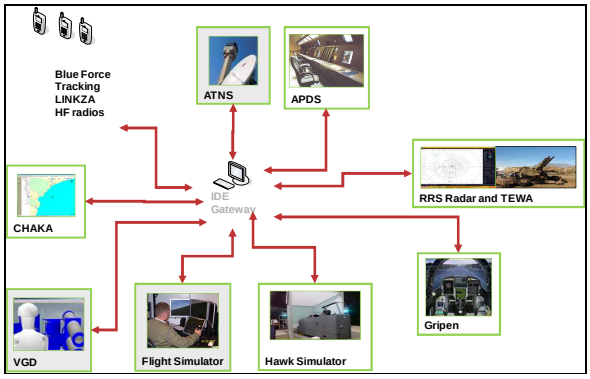

Figure 2: IDE Gateway[2]

3. Our Approach

One of the shortcomings of current systems is that Command and Control (C2) is defined for Ground Based Air Defence System (GBADS) and not for joint OOTW where new rules of engagement must be defined. Another gap is that current simulators do not provide agile and intelligent threat behaviour. Our objectives are to address this shortcoming, to explore economic, innovative solutions in times of dwindling funding by using simulation in the acquisition process, and to provide in-simulation training.

The JC2 sphere is vast, and communication up and down and between levels are required, especially for border safeguarding within the joint responsibility of the SANDF and the SAPS. The communication issue is not dealt with in this study. Perfect JC2 communication is assumed.

The benefits of simulation are well-known: it reduces costs in terms of training, doctrine development and testing of equipment. A virtual battlespace with sufficient realistic models having the look and feel of actual situations, the chaos of an actual battlespace, the precision of decisions required in actual operations, could be used to practice maneauvering, communication and fighting. [5]

The appeal of simulation [6] above other methods in border safeguarding is a result of:

- The lack of exact mathematical models for threat behaviour;
- The performance of system with complex threat behaviour can be tested, where current systems only provide pre-computed waypoints for threat movements;
- Different designs based on equipment and policies can be evaluated within one simulation;
- Experimental conditions can be changed programmatically which is not possible in a real world trial, i.e. daylight, time, and the number of objects inserted into simulation.

Some of the pitfalls of simulation are:

[2] **A subset of the IDE Gateway protocols and interfaces:**
ATNS South African Air Traffic Navigation System.
APDS Air Picture Display System developed by Saab Systems South Africa.
LinkZA The South African flavour of Link16, Radar communications protocol i.e. Local Warning Radar by Reutech Radar Systems (RRS).
TEWA Threat Evaluation Weapon Assignment by RRS [3] and CSIR.
ASTERIX All purpose STructured Eurocontrol suRveillance Information eXchange, defines the structure of the data to be exchanged over a communications link and is used for surveillance data sharing.
JC3IEDM Joint Command Control Consultation Information Exchange Data Model [4].
HLA RTI API High Level Architecture Run Time Interface Application Programming Interface [4].

- A large number of runs and experiments must be performed to make confident inferences from results (Monte Carlo simulation);
- The validity and credibility of simulation and models influence the results. Enterprise Architecture should be used to maintain traceability throughout the system life cycle;
- Stakeholder-buy-in is crucial in the system development life cycle.

Valid input data is pivotal in a simulation.

Agent-based Modelling (ABM) and simulation is applied to the border safeguarding problem using commercial software, the AnyLogic simulation package[3].

Figure 3 depicts the map of border with elements to be simulated. The environment has a Geographic Information System (GIS) map with contours, rivers and road as well as Line Of Sight (LOS) for the agents and static elements. Masts, arcs and areas are then placed on it before continually injecting threats and blue forces into the simulation. Weather and day light are frequently changed to simulate real operating conditions. Sensors are simulated to detect the threats under different conditions influencing the performance of detections ranging from blue (weak) detection to a red (very strong) detection. Different sensors (i.e. infrared cameras, radar and movement detectors) are plug-and-played into the simulation as needed, mounted on masts, aerostats or carried around.

Static elements shown in the figure include Provincial Head Quarters (HQ) (green square), Mission HQ (green circle with Patrol Vehicles), Company (Coy) HQ (green circles with Patrol Vehicles), Platoon (Plt) HQ (pink circles) deployed on the border, radars (white) on masts spaced 1.1 km apart across the border, an Aerostats with radar and or camera, Patrol Vehicles with radars and or cameras, and Dismounted Soldiers.

Threat Evaluation Resource Assignment in this simulation is based on closest distance between a threat and blue force. No weight is given to resource availability, communications, weapon load, or can-and-will-comply feedback. These criteria should be taken into account when pairing threats and blue forces. These pairings should be updated dynamically. If another blue force patrol vehicle closer to the border could more likely intercept the threat a switch over should be commended from the Sector or Company Control Centre by a commander looking at the Air Picture.

The equipment used should be expanded to include: Unmanned Aerial Vehicles (UAV), fixed wing aircraft and helicopter patrol, small vessels involved in illegal activities in the border area, and horses and motorbikes to pass through difficult terrain. Different sensors should be added to accommodate IW and Electronic Warfare (EW) information.

The manual placements of masts and sensors are inefficient and time consuming and should be automated. Sensor coverage and costs could be combined in finding an optimum placement of sensor in the border area of interest.

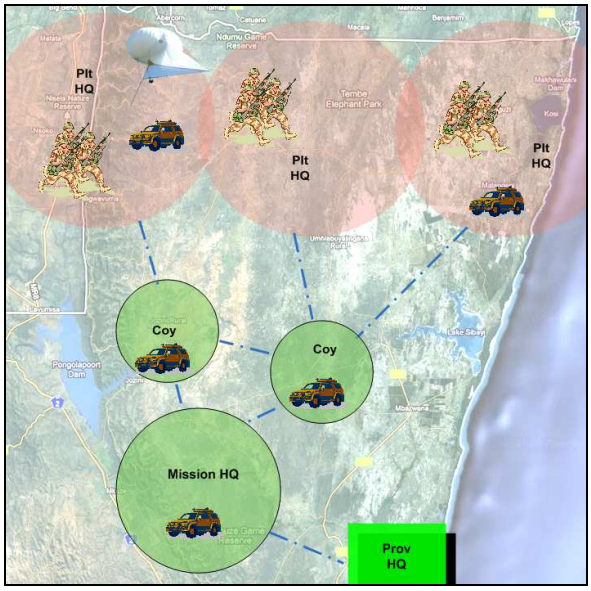

Figure 3: Border Safeguarding Deployment

In the following two subsections, we give a brief overview of agent-based modelling and a discussion of the benefits of employing constraint programming techniques to provide intelligent threat behaviour.

3.1 Agent-based Modelling

Agent-based Modelling is a modelling technique where entities are modelled as autonomous agents that are independent, intelligent, dynamic and self-directed. This means that they can sense information and then act on their own. In our

[3] http://www.xjtek.com/

simulation, we create models of independent blue and red force agents, and their actions and interactions with each other and with the environment. Different scenarios with the agents in the environment are played out on the simulator and analysed as a whole.

3.2 Constraint Programming

Our aim is to simulate threats that move intelligently: they want to move to safety (over a border) as fast as possible whilst avoiding blue forces. In previous work [7,8] we have found a constraint programming approach to solving similar path finding problem to be very efficient. A constraint-based approach allows for flexibility in the modelling process by modelling safety requirements (the avoidance of blue forces in this case) as constraints that have to be satisfied while optimising path costs (in this case the difficulty of traversing the path will be minimised). One of the major benefits is that additional requirements can easily be incorporated by adding new constraints or modifying existing constraints.

Another aspect that we address in our threat behaviour algorithm is dynamic behaviour. Although there are many approaches to solve path finding problems, there are few approaches that address problems where information is partial or uncertain. Our approach is to employ a dynamic path finding algorithm formulated as a constraint satisfaction algorithm. More information on constraint programming can be found in [9].

We thus have constraints to model the requirements for threats to maintain a safe distance from blue forces and to avoid main roads and obstacles, and that they favour hiding places out of line of sight of sensors (radar and cameras) and blue force patrol vehicles and dismounted soldiers. The objective function is to minimize the time to reach the destination, the other side of the border, where a get-away vehicle may be waiting whilst satisfying all the constraints.

The disadvantage of the constraint programming D* algorithm is the amount of resources (time and memory) used for computing large maps. This could be alleviated by setting the resolution of the maps to bigger units and using a better processor.

4. Preliminary Experiments

A balance is struck between using simulation software and developing own code. The existing models and simulation engine saves development cost by not re-designing the wheel. Financial overheads of simulation libraries or software make programming a more economic option. Open source libraries could be used but are often not supported well, bugs are not fixed and library incompatibility occurs. The threat behaviour algorithm is being developed in Java within the AnyLogic simulation.

We have designed and executed a number of experiments in the existing simulation without the planned intelligent threat behaviour algorithms. Sensors can be switched on or off (radars, infrared camera, electronic support sensors, and eye night vision goggles). Sensors can be mounted to masts, aerostat, and platoon vehicles or carried around by dismounted soldiers. Environmental conditions vary according to distribution curves to simulate weather and lighting patterns according to the time of day. The number of threats, stolen vehicles, cars attempting to cross the border illegally, dismounted soldiers and platoon vehicles injected into the simulation can be set in each experiment.

The experiment in Figure 4 shows *Car1* and *Car3* injected north of the border, heading to the destination R22 across the border. If they reach their destination across the border without being intercepted, they are home free and are injected into the simulation again. The *StolenVehicle* is injected at R22 and heads north across the border. The sensors on the platoon cars *PL1AS2* and *PL3CS2* indicate a black detection, as they get a better detection from the sensor in the vehicle the arc will change colour from blue, green, yellow orange to red. Red indicates a hot detection with identifying quality. These detections are sent to the Air Picture where the commander assigns a blue force to a threat based on closest distance.

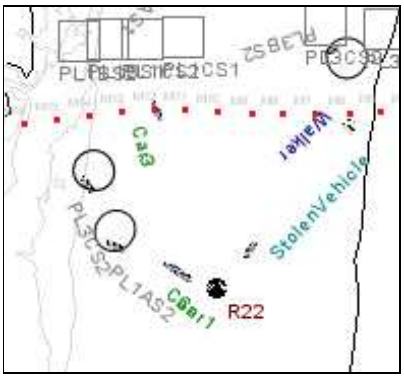

Figure 4: Experimental Scenario

From these experiments with trivial threat movement along waypoints, unrealistically high numbers of interceptions of threats were made. With intelligent threats that take evasive actions, C2 mission success testing will be improved.

From the results it is inferred that the validity of the system experiments rely on the quality of the models and agent behaviour. Near perfect detections by radar, communication and decisions will not assist in doctrine development for real situations, communication failures and evasive threats.

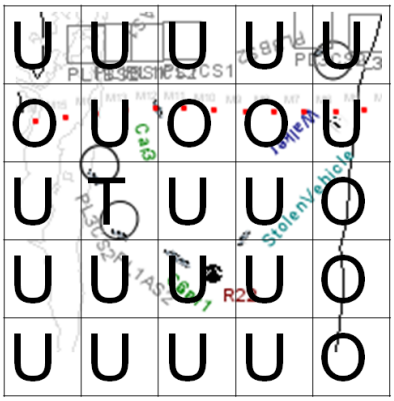

Figure 5: Modelling of Path Constraints

We are developing our constraint programming algorithm to guide the movement of threats. The area of interest is regarded as a grid by superimposing a grid such as depicted in Figure 5 on Figure 4. A grid is laid over the original map to divide the map into (x,y) coordinates cells. Each cell has a label, Unoccupied (U), Occupied (O) or Threat (T), and a cost value. The cost is the difficulty of traversal of the cell. When a blue force occupies a cell, the node is updated with 'T' indicating that the threat would keep a safe distance from that node. The Occupied label represents the fact that there is an obstacle and that cell cannot be traversed. The cost value is computed as a function of line of sight, within the main road combined with difficulty of traversal in area etc. The best path is then computed from the constraints. Different cost values can be assigned to vehicles and people travelling on foot. Constraints are added to avoid cells that are occupied and that are marked as threats. Unoccupied nodes can be selected for inclusion in threat traversal paths.

5. Conclusion

Border safeguarding presupposes new rules of engagement; threats are not hostile, are unarmed, and are civilians. They are pursued to be intercepted, not killed, and then handed over to the SAPS to be placed under arrest. The blue forces have new roles, and commanders now get instructions from another hierarchy.

The purpose of the study is to use constraint programming to insert intelligent threat behaviour in a border safeguarding agent -based simulation. Our simulation models autonomous agents in a simulated environment. The environment and agents have dynamic properties and behaviour, and interact with each other and the virtual world.

This simulation could be applied in other military spheres such as air and maritime defence, Information Warfare, and Electronic Warfare. These concepts could also be applied for perimeter control in civilian situations.

The results show that dynamic threat behaviour is vital for a border safeguarding simulation and doctrine development. More work needs to be done on threat resource pairing, blue force pursuits, C2, Tracking Air Picture, and dynamic sensor placement.

References

[1] Duvenhage, A., 2008. *The Evolution of a C2 Protocol Gateway*. EURO SIW.

[2] J OPS, 2009., *Joint Warfare Publication JWP 105 Part 7 Joint Air Defence Operations.*

[3] Roux, J.N., 2010. *Design of a Threat Evaluation Subsystem in a Ground-Based Air Defence Environment (PhD).* University of Stellenbosch.

[4] Uys, D.C. & Le Roux, W.H., 2009. *Investigating Interoperability between JC3IEDM and HLA.* EURO SIW, pp.1-16.

[5] Lt. Col. Neyland, D.L., 1997. *Virtual Combat: A Guide to Distributed Interactive Simulation*. Stackpole Books.

[6] Averill M. Law, S., 2007. *Simulation Modeling & Analysis*. New York: Mc Graw Hil.

[7] Leenen, L., Terlunen, A. & Le Roux, W.H., *2011. A Constraint programming Solution for the Military Path Finding Problem*, to appear in *Mobile*

Intelligent Autonomous Systems: Recent Trends, CRC Press.

[8] Leenen, L., Vorster, J. & Le Roux, W.H., 2010. *A Constraint-based Solver for the Military Unit Path Finding Problem*. SpringSim, 2010. Orlando, USA.

[9] Rossi, F., Van Beek, P. & Walsh, T., (editors) 2006. *Handbook of Constraint programming*. First edn. Elsevier.