DTA Technical Note 2009/7
NR 1525
ISSN 1175-6608

MANA-V (MAP AWARE NON-UNIFORM AUTOMATA – VECTOR) SUPPLEMENTARY MANUAL

G C McIntosh

September 2009

DTA Technical Note 2009/7
NR 1525
ISSN 1175-6608

MANA-V (MAP AWARE NON-UNIFORM AUTOMATA – VECTOR)
SUPPLEMENTARY MANUAL

G C McIntosh

September 2009

Abstract

MANA (Map Aware Non-Uniform Automata) is an agent-based distillation model developed by Defence Technology Agency (DTA) for use in military operations analysis studies. This document provides an overview of recent developments to the model, culminating in MANA version 5. In this version, the cell-based movement scheme of previous MANA versions has been replaced by a vector-based scheme. Advantages include: (i) larger battlefield regions can be defined which are able to be panned and zoomed, (ii) battlefield distances and agent speeds can be defined directly in terms of real world units (for example, metres, km/hr, nautical miles), (iii) sensor and weapons characteristics can also be specified directly using real world units, (iv) pre-defined libraries of sensors and weapons can be set up for importing into various scenarios without having to rescale their ranges to different battlefield sizes. Features new to MANA version 5 are described. Apart from changes to the movement algorithms, most features remain similar to MANA 4. The user is encouraged to consult the MANA 4 manual for a full description of all model features.

Defence Technology Agency
Auckland, New Zealand

EXECUTIVE SUMMARY

Background

MANA (Map Aware Non-Uniform Automata) is an agent-based model developed at Defence Technology Agency (DTA). It is essentially a combat model which has been used for a number military operations analysis (OA) studies. The agent-based modelling approach deliberately leaves out detailed physical attributes of the military entities being analyzed if these are not expected to have any bearing on the study at hand. This has the advantage that the model runs relatively fast on a PC or laptop. A large number of scenario excursions can be explored within a reasonable timeframe, so that unique tactics and outcomes can be discovered whereby a Blue force can achieve dominance over an enemy. Furthermore, unfavourable Blue force actions can be uncovered in order to provide a 'lessons learned' of what Blue force should not do in particular situations.

Development of the MANA model has been ongoing since its inception in 2000, with the model being continually upgraded when new features are required for OA studies. This report describes the most recent version of MANA, version 5. In previous versions, a grid-based scheme was used for moving agents on the battlefield. In that case, agents were confined to a maximum grid size of 1000 × 1000 squares. In version 5, the grid-based movement scheme has been dispensed with and a vector-based approach for agent movement implemented. This has a number of advantages:

1. Much larger battlefields can be set up which can be panned and zoomed.
2. All battlefield distances, sensor and weapons ranges and agent speeds can be defined in terms of SI units. This avoids having to relate grid squares to real-world distances (which was the case for previous MANA versions) and allows easier interpretation of scenarios.
3. The vector-based approach provides greater flexibility for developing new model features, for example, intelligent path finding algorithms.

This report explains the implementation of the vector-based movement scheme and describes features new to MANA version 5. MANA version 5 has been named MANA-V. The 'V' represents vector-based movement, and the Roman numeral 5.

Sponsor

DTA Project D0202.

Aim

To provide analysts with an understanding of MANA-V so that the model can be used effectively for military operations analysis studies. This document describes features which are new to MANA-V and were not in the previous version. It is recommended that users consult the MANA 4 manual in order to obtain a full description of all model features.

CONTENTS

1 INTRODUCTION

MANA (Map Aware Non-uniform Automata) is an agent-based model developed at Defence Technology Agency in New Zealand by the Operations Analysis group [1 – 3]. It has been used for a number of studies, including: the modelling of civil violence management [4], maritime surveillance [5], investigating modern warfare as a complex adaptive system [6], studies of network centric warfare (NCW) [7,8] and studying counter-IED measures [9]. It has been used by various defence science establishments amongst TTCP nations and has become one of the main modelling tools used at the bi-annual International Data Farming Workshops [10]. MANA has also been used for a number of student projects at the Naval Postgraduate School in Monterey (for example, see Refs. [11 – 14]). This report describes the most recent version of MANA, version 5.

Over the past decade, interest has grown in using agent-based models for combat modelling [15] and military operations analysis. This contrasts with more detailed combat simulations such as JANUS or CAEn which run in near real time. Agent-based models purposefully omit, or 'distill' out, detailed physical attributes of the military entities being modelled if these are not expected to have any bearing on a particular study [1]. This allows scenarios to be run relatively fast on a PC or laptop. Multiple scenario excursions can be explored within a reasonable time-frame in order to discover unique situations or tactics whereby a Blue force can achieve dominance over an enemy. Another feature of agent-based models is that, although the one-to-one interaction between pairs of agents in the model can be relatively simple, the combined effect of many agents interacting in a scenario can lead to complicated collective dynamics and emergent behaviour which was not intentionally input into the scenario [15,16]. In this regard, agent-based models have the potential to represent the more chaotic and non-linear aspects of military conflicts (see Chapter 1 of Ref. [2] for further discussion) when a more detailed combat model such as JANUS might be too prescriptive [17].

In terms of agent-based models, MANA has a number of advantages:

- It is user friendly and has an easily navigable user interface. Scenarios can be quickly edited 'on the fly' during development, allowing different ideas to be quickly explored. In this regard, MANA can be thought of as a 'sketch pad' for trying new military tactics.
- MANA is supplied as a pre-compiled executable, so runs relatively fast compared with other agent-based models built on interpreted languages such as Java.
- The model has a built-in data farming capability which allows rapid exploration of parameter spaces.
- MANA can model communications links between agents for information sharing [8]. Hence, aspects of network-centric warfare (NCW) can be studied.
- To model terrain features, MANA uses coloured bitmaps. This has the advantage that terrain features can be quickly altered using a simple image editor (for example, MS Paint), while a scenario is in development.

- MANA has a large number of trigger states which agents can go into upon various events occurring on the battlefield. This increases the richness of behaviour and insights that can be obtained from scenarios.

Development of MANA commenced in 2000 with inspiration coming from Ilachinski's early agent-based combat model, ISAAC (Irreducible Semi-Autonomous Adaptive Combat) [15]. It was decided that MANA should be specifically engineered with the requirements of NZDF in mind. MANA was intended as a flexible, low maintenance model. This relates well with NZDF being a small, adaptable force making the most of its resources.

Development of MANA has proceeded through several versions as follows.

MANA 2 (2002 – 2003): This represented the first fully usable version of MANA. It comprised the core of the present-day model with all essential movement weightings, sensor and weapons settings and terrain editing features in place. Sensor and weapons characteristics were represented using either a simple cookie cutter scheme or with range-dependent probability tables. The concept of a Situational Awareness (SA) map for squads was implemented allowing enemy contact information to be shared amongst agents in a squad. Rudimentary aspects of NCW could then be modelled; for example, a delay in passing information from an agent's sensors to its squad Situational Awareness map.

MANA 3 (2003 – 2004): This was a fully developed version of MANA which has been in widespread use. This version appeared quite similar to MANA 2, but included several refinements. As well as the squad Situational Awareness map, communications links between squads and information sharing were explicitly modelled so that aspects of NCW could be fully explored [8]. Accordingly, additional movement weightings were added so that agents could respond to information received remotely from other squads. A data farming capability was added to fully automate the exploration of a scenario's parameter space. Special movement algorithms for aircraft flight paths were also added in order to implement search patterns for scenarios involving maritime patrols[1].

MANA 4 (2005 – 2006) MANA 4 was released in early 2007. It is essentially MANA 3, but with several new features added on.

- A data streaming capability was added so that MANA could be used for human-in-the-loop experiments carried out by RAND Corporation.
- The main battlefield display has a zoom feature which allows finer control over agent placement and modifying terrain features in larger more intricate scenarios.
- A genetic algorithm was included to help automate scenario development, and to be used as a research tool [18].
- A data analysis tool was added to allow the graphing of casualty data and other combat statistics from multiple scenario runs.
- Finite sensor and weapons arcs were added in order to model agents with a limited field of view (FOV). Correspondingly, an angular dependence was

[1] These search algorithms make use of the Travelling Salesman algorithm.

added to agent behaviour whereby they can turn to face enemies they become aware of.

- New movement weightings were added which depend on the enemies' direction of facing. This allows agents to choose their direction of approach towards an enemy, and goes some way towards modelling aspects of precision manoeuvre.
- Squad formations were added with a choice of patterns such as line, column or wedge.

All previous versions of MANA used a grid-based scheme for agent movement on the battlefield. Overall, this scheme has worked well for the type of distilled scenario for which MANA was originally intended. However, there can be situations where the grid-based scheme imposes artefacts onto scenarios. For example, when deciding to move to a new grid square, agents have a choice of only 8 adjacent squares to move to (see Figure 1(a)). This can impose a directional bias onto agent movement along horizontal, vertical or diagonal directions, which may skew results in some scenarios. These movement algorithms also require a correction factor to allow for the fact that, under the grid-based system, agents would be travelling a factor of $\sqrt{2}$ faster in diagonal directions than horizontal or vertical directions.

From a user point of view, the grid-based movement scheme can prove tedious if the user must convert between real-world coordinates and grid-based model distances when setting up a scenario. A reverse conversion must also be performed when extracting results from the model after scenarios have been run, and there is a trade-off between run-times and the fidelity of the model.

MANA version 5 has been developed to alleviate such issues and to allow greater flexibility in developing new features for the model. Essentially, the battlefield grid has been replaced by a vector-based movement scheme with a continuous coordinate system (see Figure 1(b)). This has a number of advantages:

- Since agents are no longer locked to a battlefield grid they have greater freedom of movement, and are able to travel outside the main battlefield perimeter. This allows extended battlefield maps to be defined which can be panned and zoomed. There is then the possibility of modelling larger scale conflicts with concurrent operations occurring in different zones of the battlefield.
- Battlefield distances and agent speeds can be specified directly in terms of real-world distances and times. This avoids cumbersome conversion factors between real-world coordinates and model grid distances which were required in previous MANA versions. The reverse conversion is also avoided when extracting measures of effectiveness from the model after scenarios have been run. Displaying scenarios in terms of real-world coordinates is beneficial for presenting the model to military personnel and decision makers who are not necessarily familiar with the detailed workings of the model.
- Sensor and weapons ranges can be specified directly in terms of real-world distances (metres). This allows weapons ranges to be entered into the model directly from known specifications (for example, from Jane's catalogues);

libraries of weapons, sensors and vehicle specifications can be maintained for use in various scenarios.

- From a software point of view, the vector-based approach allows the flexibility to develop new features. For example, Section 3.6 describes a new path finding scheme which allows agents to navigate the battlefield without becoming trapped behind obstacles.

By removing the battlefield grid, units of distance effectively become continuous. However, there remains a discrete aspect to the model in terms of defining the battlefield time step. If this is set too fine, scenarios will take an excessive amount of time to run. On the other hand, if the battlefield time step is defined too coarsely then key events occurring on the battlefield might be skipped or enemy detections missed, resulting in incorrect conclusions being drawn from a scenario.

Throughout the remainder of this report, MANA version 5 is referred to as MANA-V. The 'V' represents vector-based movement and the Roman numeral 5. In the following, the MANA-V movement scheme is described in more detail and a list of features new to MANA-V is provided. From a usability point of view, MANA-V is similar to MANA 4. For example, personality weightings, the sensor and weapons model, Situational Awareness maps and modelling of communication links remain mostly unchanged. Hence, an entirely new manual for MANA-V has not been written. We recommend consulting the MANA 4 user manual [3] to explain most features already present in the model, and using this manual as a supplement.

2 TECHNICAL ASPECTS OF MANA-V: VECTOR-BASED VERSUS CELL-BASED MOVEMENT

Previous versions of MANA used a grid-based movement scheme, as illustrated in Figure 1(a). Agents existed on a finite battlefield grid and moved from grid square to grid square at each time step depending on conditions on the battlefield. These conditions could involve the locations of enemies, other friendly units, or terrain features. The user ascribes personality weightings to agents to guide their general behaviour. For example, agents might be calibrated to prefer closer proximity to other agents of the same force in order to maintain numerical superiority over an enemy. Alternatively, agents could be defined as more aggressive and prefer to attack enemies. For each adjacent battlefield grid square available for an agent to move to at the next time step, a distance-dependent penalty function, P_i, is calculated for each non-zero personality weighting set by the user:

$$P_i = 1.0 + \frac{\sum_{m=1}^{M} (D_N(m) - D_O(m)) D_L(m)}{100 \sum_{l=1}^{M} D_L(l)} \qquad (1)$$

In Equation (1), M is the number of other active agents (both friendly and enemy) detected by the agent about to make a move[2], D_N and D_O are new and old distances

[2] These detections could be from an agent's own sensors or received remotely from other squads through communications links.

(a) **(b)**

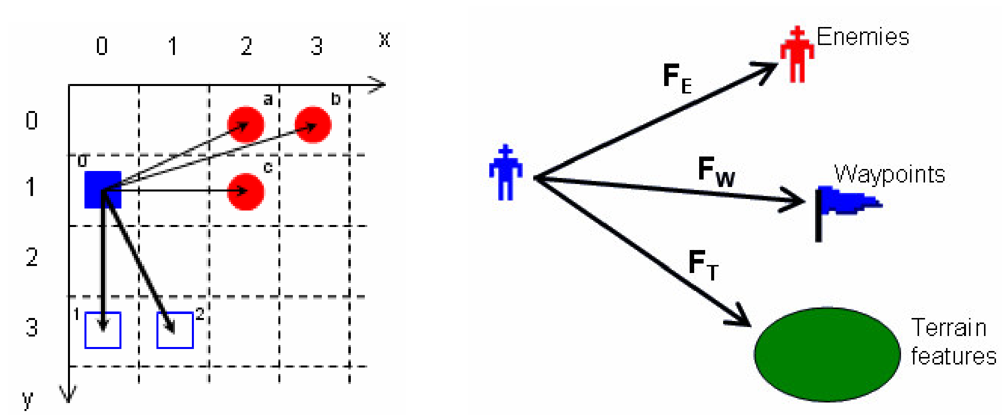


Figure 1: (a) The cell-based movement scheme used in previous MANA versions. The solid blue square represents an agent about to make a move. Hollow blue squares represent other friendly agents while red circles represent enemy agents. (b) The vector-based approach used in MANA-V. A resultant vector governing the agent's movement is calculated based on individual vectors to other agents and terrain features on the battlefield.

to all other agents respectively and D_L is a normalization factor to scale the calculation according to the size of the battlefield. The overall penalty value for each adjacent square is given by the sum over all P_i, multiplied by the associated personality weightings, W_i, defined by the user:

$$Penalty = \sum_{i=1}^{N} W_i P_i . \qquad (2)$$

Calculations using Equations (1) and (2) result in adjacent cells which are favourable for the next move receiving a penalty value < 1.0, while unfavourable cells receive a penalty value > 1.0. The cell receiving the lowest penalty value will be the preferred cell for the agent to move to at the next time step. Furthermore, a small random component is applied to the calculation so that the cell with lowest penalty value is most likely to be chosen but, on occasion, a nearby cell might be chosen instead. This helps to reduce the directional bias imposed by the grid-based scheme discussed above.

In MANA-V, the cell-based scheme has been replaced by a vector-based scheme, as illustrated in Figure 1(b). Each agent monitors all other agents and terrain features within sensor range and calculates a vector to each. Vectors can also be calculated towards distant agents appearing on the moving agent's Situational Awareness map which are not necessarily within the agent's own sensor range. This allows agents to respond to information received remotely through communications links. Vector addition is applied to all individually calculated vectors, weighted by associated personality weightings W_i defined by the user, to yield a resultant force vector, $\boldsymbol{F}$. For example, the resultant vector from Figure 1(b) is

$$\boldsymbol{F} = w_E \boldsymbol{F}_E + w_W \boldsymbol{F}_W + w_T \boldsymbol{F}_T, \qquad (3)$$

where $\boldsymbol{F}_E$, $\boldsymbol{F}_W$ and $\boldsymbol{F}_T$ are individual vectors calculated towards enemies, waypoints and terrain features respectively. Furthermore, w_i includes a distance-dependent factor so that agents which are further away have less influence on movement than closer agents.

Once a resultant vector $\boldsymbol{F}$ has been determined, agent movement is calculated from Newton's second law $\boldsymbol{F} = m\boldsymbol{a}$ and standard kinematic equations for constant acceleration:

$$S = S_0 + v\Delta t + a\Delta t^2/2 \qquad (4)$$

and

$$v = v_0 + a\Delta t, \qquad (5)$$

where $S - S_0$ is the distance travelled during time interval Δt and $v - v_0$ is the corresponding change in velocity.

A value of $\Delta t = 1$ second is used in MANA-V as the default battlefield time step, with the user free to change this value. There is a trade-off in setting Δt. If this time increment is set too fine then, although battlefield movement will be more precisely represented, scenarios may take too long to run. On the other hand, if the time increment is set too coarse then important detection or firing events might be missed, and agents are able to jump terrain features or 'walk through walls'.

Agent speeds can be set similarly to MANA 4. The default speed given to agents when opening a new scenario is 25 km/h, which is suitable for military vehicles proceeding cautiously through a complex battle-space. Changing this speed setting to 5 – 10 km/h would be suitable for infantry on foot. The user-defined speed specifies the maximum agent speed on open terrain. For slower going terrain (specified as Going < 1.0), the agent's achievable speed will be reduced in direct proportion.

Equations (3) to (5) require defining a mass, m, for agents in the model. This is specified by the user in a relative sense. It is not possible to define an agent's mass absolutely since this would require detailed information about each agent's motive force (and would defeat the purpose of implementing an agent-based model). The default value provided, $m = 1.0$, gives finite inertia to agent movement and agent acceleration can be clearly noticed when they first commence movement. On the other hand, a value of $m = 0.1$ gives almost instantaneous acceleration, and reproduces behaviour similar to MANA 3 and MANA 4. A larger value for m (for example, $m = 10$) may be useful for some scenarios to represent the acceleration of large vehicles or seagoing vessels.

Applying a purely physics-based approach, as encapsulated in Eqs. (3) to (5), tends to produce unexpected behaviour. For example, agents will sometimes go into orbit around waypoints or ricochet across the battlefield. To prevent this type of anomalous behaviour, agents are given additional movement rules as follows:

1. If forces have ceased acting on an agent (for example, if all enemies have been killed), the agent will be brought to a halt.

2. Agents are disallowed from moving in a direction opposite to the net force acting on them. This rule prevents agents from going into orbit around each other or oscillating about waypoints.

These rules prevent anomalous agent behaviour from appearing, and often yield scenarios indistinguishable from MANA 3 and MANA 4. To a certain degree, forces corresponding to personality weightings in MANA-V can be considered as 'psychological forces', which encapsulate the agents' tendency for movement on the battlefield. This is in contrast to physical forces such as bumping into obstacles or terrain features.

Similar to MANA 4, agents in MANA-V include a direction of facing. MANA-V has been improved so that icons representing aircraft, vehicles or ships can change their displayed orientation depending on the agent's direction of movement. This is a cosmetic improvement which is useful for presenting model findings to audiences unfamiliar with detailed workings of the model. For example, scenarios involving aerial patrols will display aircraft icons facing correctly towards the direction they are flying, whereas previous MANA versions would have displayed aircraft with a fixed icon and appearing to fly backwards on occasions.

3 MANA-V FEATURES

Features new to MANA-V, which were not in MANA 4, are described in this section. We recommend consulting the MANA 4 user manual [3] for a full description of all model features. Previous MANA 3 and MANA 4 scenarios will not be fully compatible with MANA-V. Nonetheless, it is possible to load earlier version scenarios into MANA-V and still retain the overall intent of those scenarios. For example, all squad structures will remain intact and personality weightings should not require adjustment. Some broken features requiring repair include sensor and weapons parameters, battlefield dimensions and agent home box placement.[3]

Figure 2 illustrates the 'Edit Squad Properties' panel for MANA-V. As can be seen, the layout is fairly similar to MANA 4. Main differences are:

1. The 'Ranges' tab has been re-named 'Tangibles'. This includes settings which involve more physical attributes of the agents such as speed or armour thickness.
2. The 'Advanced' tab includes various personality weightings which were added to MANA 4 but were not in MANA 3; for example, specifying squad formations and defining movement direction depending on the enemy's direction of facing. This tab was previously named 'Personalities 2' and was included as a sub-panel of the 'Personalities' tab in MANA 4.
3. The 'Algorithms' tab has been removed since many settings under this tab involved defining agent movement under the grid-based scheme which are no longer applicable to MANA-V. However, the Travelling Salesman algorithm for target search has been retained and included under the 'Advanced' tab.

---

[3] This applies at the time of writing. Backward compatibility with earlier version MANA scenarios will be addressed in future updates of MANA-V.

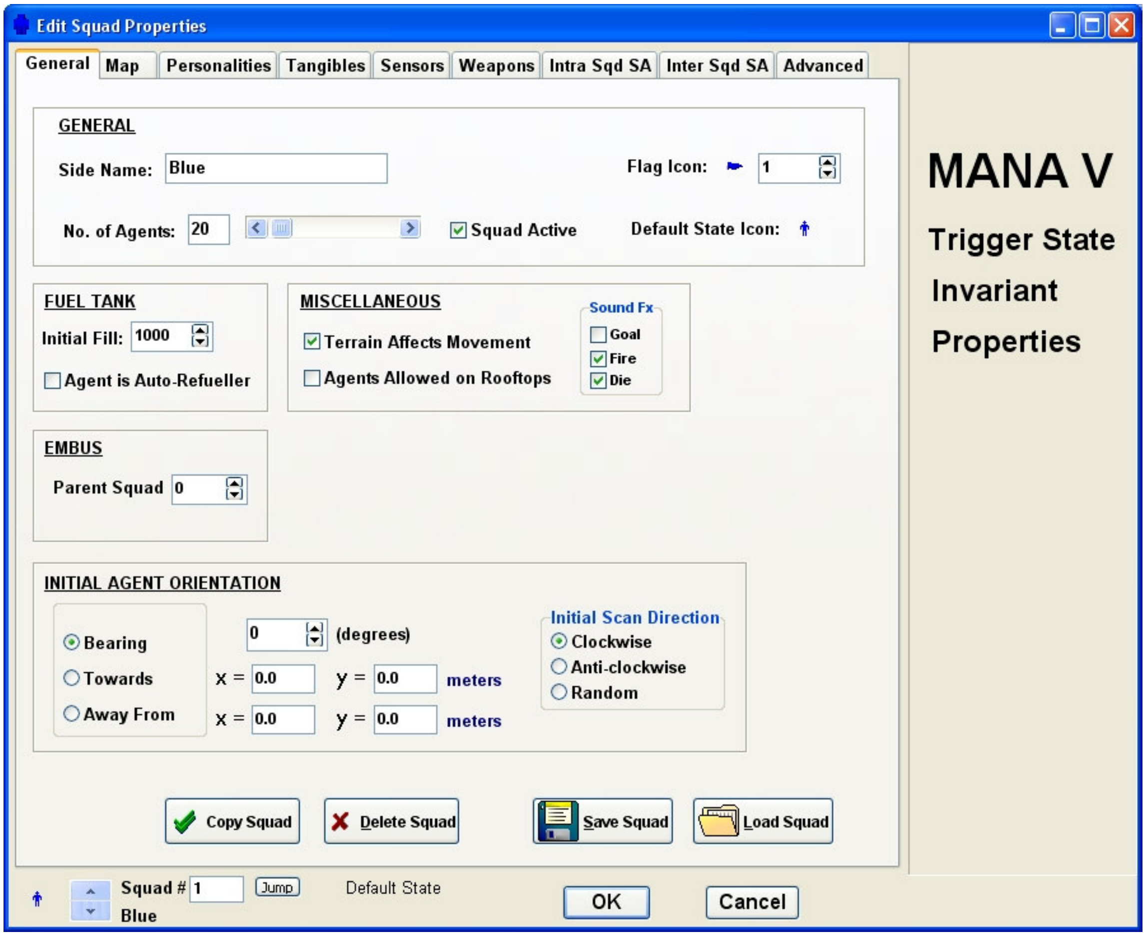


Figure 2: The 'Edit Squad Properties' panel in MANA-V. Shown are properties under the 'General' tab.

3.1 Defining the Battlefield

Similar to previous MANA versions, battlefield settings can be modified using the 'Edit Battlefield' panel, as illustrated in Figure 3. Many adjustable parameters on this panel are similar to MANA 4 settings. The main difference is that the battlefield size is now defined by two sets of parameters.

Global Map: This defines the overall size of the battlefield.

Local Map: This defines the size of the battlefield view seen on the main MANA panel, as illustrated in Figure 4. The local map must be defined within the global map.[4]

Battlefield Time Step: The battlefield time interval can also be set on the 'Edit Battlefield' panel. Retaining the default value of Δt = 1.0 seconds per battlefield time

[4] Note: if the overall battlefield size is reduced then care is required to ensure that all agents' home boxes and waypoints remain defined within the new global battlefield boundary. A warning of any breaches is provided upon closing the Edit Battlefield panel.

step is recommended unless particular scenarios require a different fidelity due to larger battlefield sizes or high speed agents representing aircraft or missiles.

Caution: In principle, many parameters such as agent speed, turning rate, weapons firing rates and sensor detection probabilities should rescale automatically with change in battlefield time step. However, exact rescaling is not guaranteed and it is recommended that scenarios be re-calibrated if the battlefield time step is changed. Also, time-dependent parameters associated with the squad Situational Awareness maps and communications links (for example, Latency and Contact Persistence) are specified directly in terms of model time steps in order to keep them well defined. Hence, these parameters will require adjustment if the battlefield time step is changed.

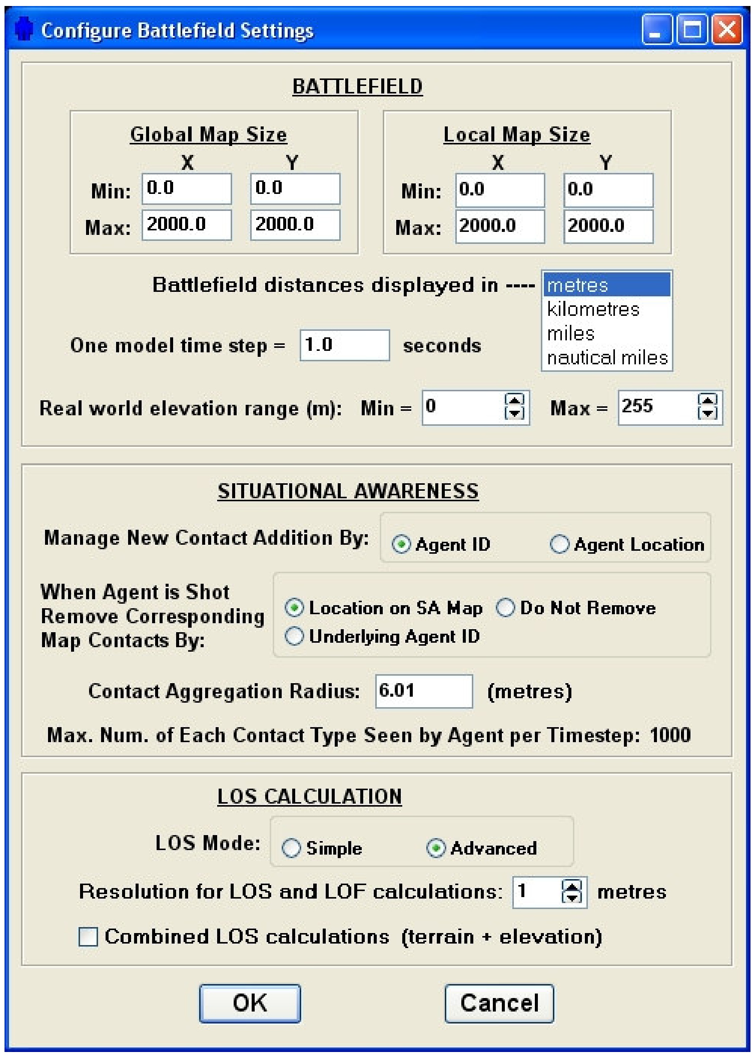


Figure 3: Defining battlefield dimensions using the 'Edit Battlefield' panel.

A global view of the battlefield can be accessed by clicking 'Mini Map' under the 'View' pull-down menu, as illustrated in Figure 5. The appearance of the Mini Map, showing a global view of the overall battlefield, is illustrated in Figure 6. Here, the white square denotes the battlefield view which will be seen on the main MANA panel, as illustrated in Figure 4. Clicking on the Mini Map allows the battlefield view to be changed. The battlefield view can also be zoomed using the slider bar beneath the information panel on the main MANA panel.

The default settings for the Local Map involve a square aspect ratio, as seen on the main MANA panel. If a non-square aspect ratio is defined then the main MANA panel should be re-sized accordingly so that the battlefield does not appear distorted. The global battlefield size can take on any aspect ratio in accordance with dimensions specified by the user. The Mini Map will then display the global battlefield with the correct aspect ratio. Similarly, the Situational Awareness map and Inorganic Situational Awareness map will display events for the entire battlefield using the same aspect ratio.

The MANA-V movement algorithms are calculated using SI units; distances are calculated in metres and time in seconds. There is an option for displaying battlefield coordinates in other units such as kilometres, miles or nautical miles, shown in Figure 3. However, this is a cosmetic feature which allows, for example, the displaying of maritime scenarios using nautical miles. Regardless of the displayed units, all movement calculations are carried out using metres as the unit of distance. Furthermore, all sensor and weapons ranges remain specified in metres.

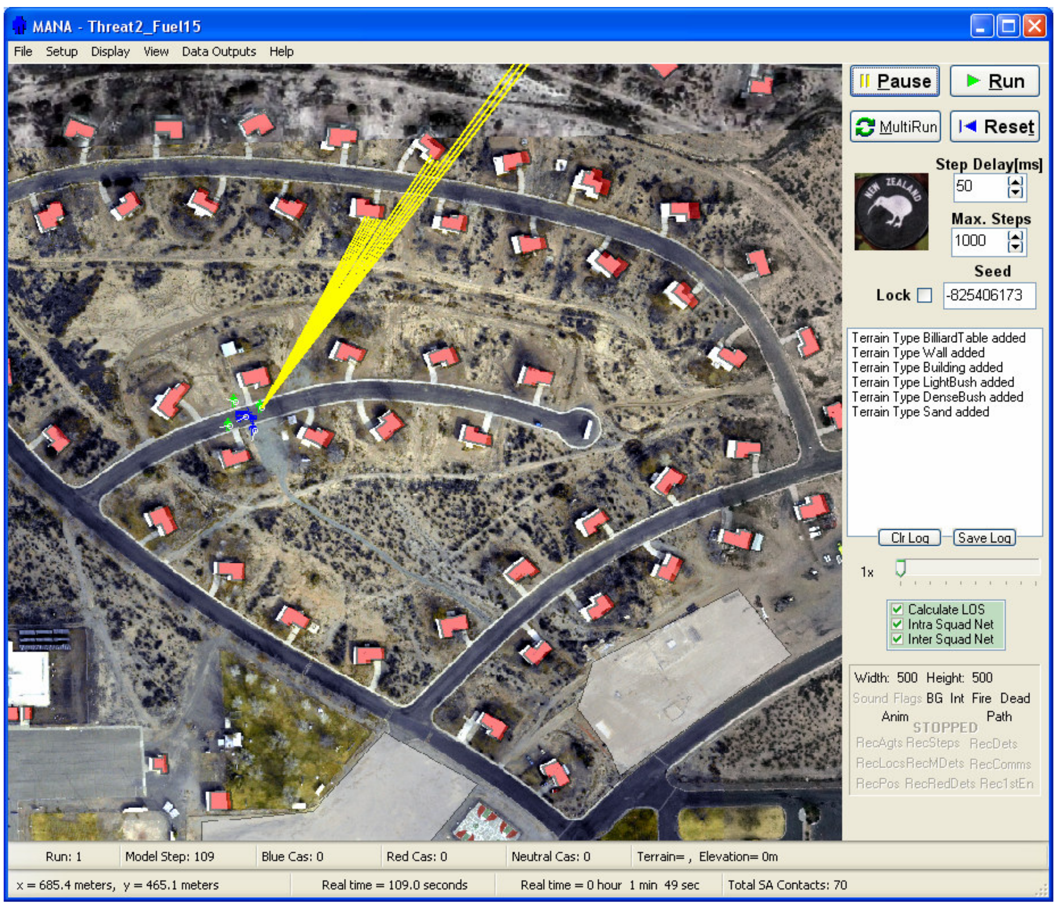


Figure 4: A typical MANA-V screen shot showing the main battlefield view. This example shows the Playas Training and Research Center (PTRC), New Mexico being modelled in MANA-V.

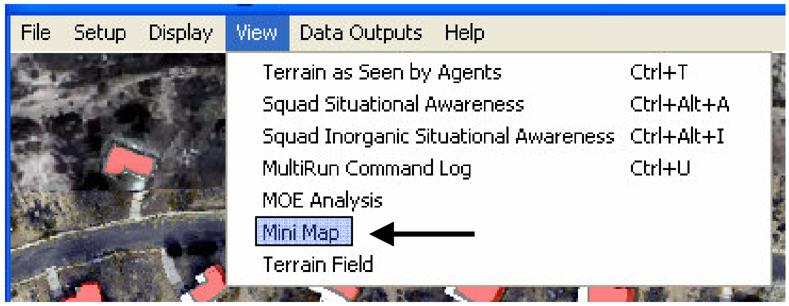


Figure 5: Accessing the mini-map which brings up a global view of the entire battlefield, as shown in Figure 6.

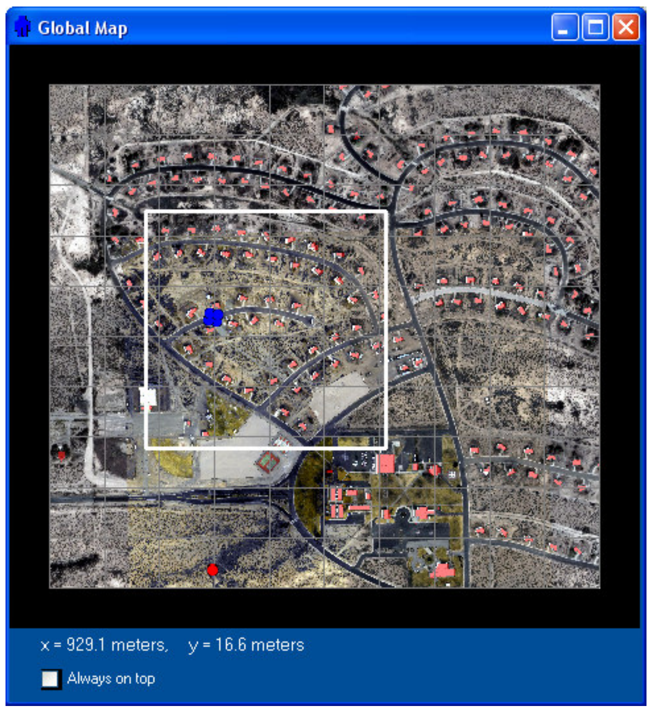


Figure 6: Illustrating the Mini Map which displays the entire battlefield. The white square defines the battlefield view which will be seen on the main MANA display, as shown in Figure 4.

As in MANA 4, a default 2-man scenario is created by clicking ‘New’ on the ‘File’ pull-down menu. This scenario appears similar to previous MANA 2-man scenarios, with minor exceptions. The battlefield size will be set to 2 km × 2 km and weapons ranges will be set to 200 metres. This aligns with the philosophy of applying real-world coordinates in MANA-V and more realistic weapons ranges.

In previous MANA versions, problems can arise when combining elevation and terrain data for certain scenarios. For example, terrain which has been specified as 100% concealment should block all line-of-sight (LOS) between two agents at ground level. Similarly, a terrain feature with sufficient elevation should also block LOS at ground level. Problems can occur if an agent happens to be elevated above a particular terrain feature which the agent should be able to see over. Under previous

MANA versions, LOS would still be blocked if the elevated terrain feature is specified as 100% concealment. In MANA-V this issue has been addressed by providing a 'Combined LOS calculations' option, shown in the 'Edit Battlefield' panel in Figure 3. The default unchecked value will maintain the same behaviour as previous MANA versions. Setting this option to true will allow concealment terrain to only block LOS if it has sufficient elevation above the agents' sensor height. For example, this option is useful for scenarios involving urban environments with aerial platforms such as UAVs or agents placed on rooftops and buildings have specified heights.

For modelling urban environments, locating agents on rooftops may be required. However, MANA tends to treat Building/Wall terrain (defined as Going = 0, Concealment = 1, Cover = 1) as an obstacle, thus disallowing such agent placement. In MANA-V an option is provided under 'General' settings, called 'Agents Allowed on Rooftops' (see Figure 2). This inhibits the obstacle calculation for particular squads and allows agents to be placed on Building/Wall terrain. The agents' sensor height will need to be specified greater than the Building/Wall height in order for the agents to be able to 'see' out over their environment. The 'Combined LOS calculations' option (see Figure 3) will also need to be set true.

The 'Edit Battlefield' panel (see Figure 3) includes an option for specifying the resolution of LOS calculations.[5] As a rule of thumb, we suggest that scenarios with a battlefield size of less than ~2 km × 2 km should run sufficiently fast with LOS calculated to the maximum resolution of 1 metre. On the other hand, scenarios with sizes above ~10 km × 10 km may tend to slow down. This is particularly the case if LOS has to be calculated between distant agents on the battlefield. Lowering the resolution of the LOS calculation can improve the execution speed of the scenario. However, one should ensure that the resolution is not lowered below the size of significant terrain features on the battlefield. For scenarios without any terrain features, the LOS calculation can be turned off on the main MANA panel (see Figure 4).

3.2 Working with Terrain and Elevation Maps

Similar to MANA 3 and MANA 4, terrain features and elevation are defined in MANA-V using coloured bitmaps. Such a scheme allows terrain features to be easily changed 'on the fly' using a simple image editor such as MS Paint. Following previous MANA versions, different colours are used to specify particular terrain types, and a grey-scale bitmap is used to define terrain elevation.

The disadvantage of using bitmaps for terrain is that they can lead to large memory usage depending on the size of the battlefield. For example, we have succeeded in setting up a scenario with a terrain bitmap of 3280 × 5117 pixels on a Windows XP machine with 1 gigabyte of RAM. Given a scenario resolution of 1 pixel per meter (which we have found adequate for urban scenarios) this can be used to represent a battlefield region of ~3.2 × 5.1 km, which is sufficient to model interesting activity in an urban setting. The downside is that terrains of this size use ~500 megabytes of

---

[5] The LOS algorithm in MANA essentially involves stepping across the terrain until an obstacle is found.

system memory and the scenario can take several seconds to load. However, once loaded, the scenario runs with similar performance to smaller battlefields. With lower resolutions (less than 1 pixel/metre) larger battlefield regions can be represented. For open terrains, or maritime scenarios with no terrain features, there is no such limitation on the battlefield size.[6]

MANA-V also retains the Scenario Map Editor so that terrains can be edited from within a scenario. In this case, only the terrain visible in the main MANA battlefield view can be edited at a given time. If terrain features are being added to a blank battlefield, the user will be given an option for setting the size of the terrain bitmap.

If a scenario has been loaded into MANA-V which includes a terrain bitmap, changing the global map size via the Edit Battlefield panel will have an effect on the bitmap as follows. If the global map size is reduced, the terrain bitmap will be truncated at the perimeter of the new battlefield. Conversely, if the global map size is increased, the terrain bitmap will be appended with new blank regions. The newly allocated terrain will be specified as 'Billiard Table' (black), and the user then has the option to add terrain features to these regions as required; either using the Scenario Map Editor in MANA-V or using an image editing program such as MS Paint.

3.3 Sensor Behaviour

Sensor implementation in MANA-V remains very similar to MANA 4. The main difference is the Advanced Sensor mode where an option has been provided for choosing particular classes of agent which can be seen by a sensor (see Figure 7). For

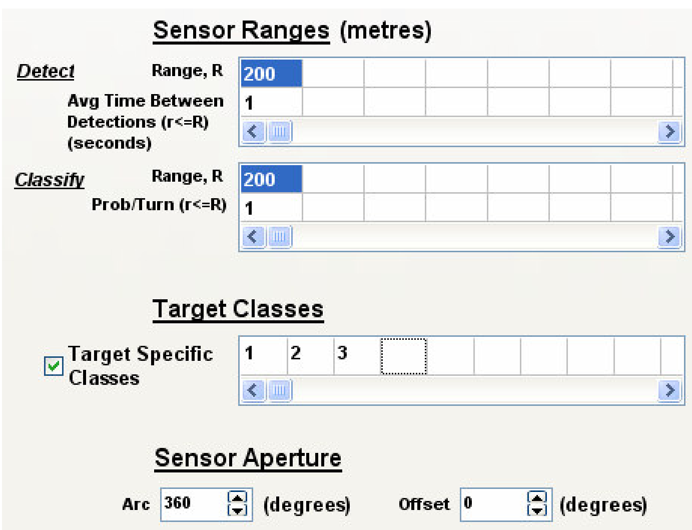


Figure 7: The Advanced Sensor mode in MANA-V. A table is provided to specify particular classes of agent able to be detected by a sensor. The default option, 'Target Specific Classes' is set to false and allows agents of all classes to be detected.

[6] Note that the earth's curvature could become a factor in larger battle spaces, and is not considered in the MANA model calculations.

example, this feature can be useful for modelling situations where sensors are able to detect vehicles out to greater distances than personnel. Previous studies indicate a clear delineation between the ability of sensors to detect personnel or vehicles in various environments and at different ranges [19]. One solution is to provide an agent with two sensors: one sensor for sensing vehicles and the other for sensing personnel. In this case, agents corresponding to vehicles or personnel would be given different class settings in the scenario.

3.4 Scan Modes for Agent Direction of Facing

Similar to MANA 4, agents in MANA-V can be defined with a direction of facing and sensors with limited field of view (FOV). The direction of facing can be displayed on the battlefield via the ‘Agent Orientation’ tick option under the ‘Display’ pull-down menu. Agents can turn to face enemies seen from their own sensors or via information obtained from their Situational Awareness maps. In the absence of enemy sightings, agents can be specified to scan the battlefield according to several options shown in Figure 8.

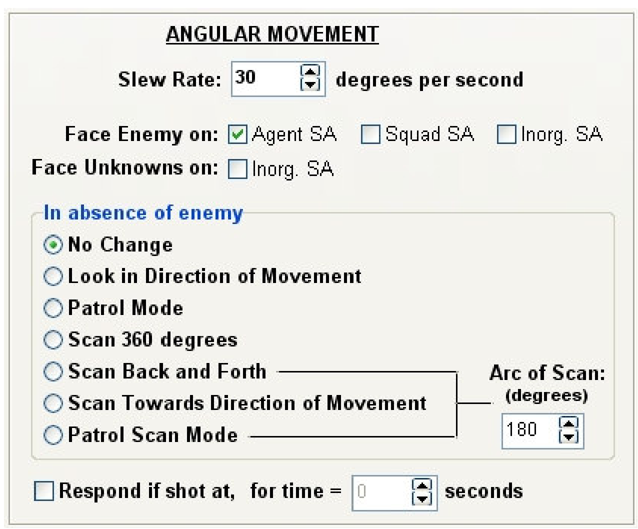


Figure 8: Various sensor scanning modes available in MANA-V.

No Change: The agent will maintain a constant direction of facing. This will be the initial direction specified by the user or the direction the agent was facing when it last sighted an enemy.

Look in Direction of Movement: As the name suggests, agents will look towards their direction of movement.

Patrol Mode: This applies for a squad of agents moving together as a group. Agents will look outwards away from the centre of the group. Thus, each agent in the squad will have a designated zone of surveillance for the squad.

Scan 360 Degrees: As the name suggests, agents will rotate their direction of facing through 360° in accordance with their slew rate.

Scan Back and Forth: Instead of scanning around 360°, agents will scan back and forth. The speed of the scan will be determined by their slew rate. There is an additional option, 'Arc of Scan' (see Figure 8), to define the angular range through which agents scan back and forth. The centre of this range is defined by the agent's direction of facing. This feature is useful for modelling agents at observation posts.

Scan Towards Direction of Movement: This combines 'Look in Direction of Movement' with 'Scan Back and Forth'. Agents will scan back and forth while looking towards the direction they are moving. This feature is useful for modelling sensors on moving vehicles. The 'Arc of Scan' setting also applies to this option.

Patrol Scan Mode: This combines 'Patrol Mode' with 'Scan Back and Forth'. Agents grouped together in a squad will scan back and forth while looking out towards their own zone of surveillance. The 'Arc of Scan' setting defines the angular extent of the scan.

For agents scanning 360° or scanning back and forth, there is an option to set the direction in which they commence scanning under the 'General' settings panel, as shown in Figure 2. The initial scan direction can be specified as clockwise, anti-clockwise or random.

3.5 Embussing and Debussing

MANA 4 has an embussing feature to allow agents to be attached to a parent agent (for example, representing infantry carried onboard vehicles) and to debus these agents upon particular events occurring on the battlefield. MANA-V extends this feature to allow agents to re-embus after they have debussed. Various options for controlling embussing and debussing behaviour in MANA-V are shown in Figure 9.

Figure 9(a) shows options available for triggering agents to embus or debus during a scenario run. These settings are accessed under the 'Edit Squad Properties/Tangibles' panel. Similar to MANA 4, one defines a squad of agents to be embussed to a parent agent at the start of a run. The parent squad is defined under the 'General' settings (see Figure 2).

If 'Release from Parent Squad' and 'Release Child Squads' are both left unchecked, the child squad will remain embussed to the parent for the duration of the scenario. On the other hand, if any of these options are ticked for a particular trigger state, the child agents will debus from their parent and take on independent behaviour. In particular:

Release from Parent Squad: If this is checked, the child squad will leave its parent squad.

Release Child Squads: If this is checked, the parent squad will release all child squads that it has embussed.

Activating debussing in this way requires using different trigger states to fully implement. One would typically specify agents to debus upon occurrence of particular

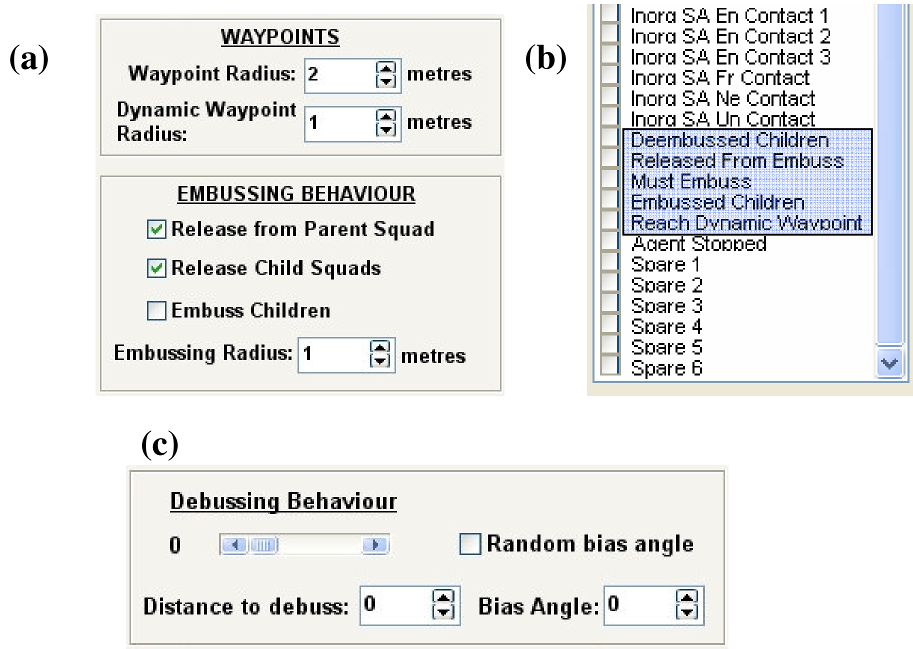


Figure 9: Options for setting embussing and debussing behaviour. (a) Tick boxes to define whether agents will embus or debus in particular trigger states. (b) Various trigger states to facilitate embussing and debussing activity. (c) Personality weightings to control debussing behaviour.

events on the battlefield. Once agents are debussed the 'Release from Parent Squad' and 'Release Child Squads' check boxes become ineffective.

Embus Children: MANA-V includes the option for parent squads to re-embus their child agents after they have debussed. If the 'Embus Children' check box is ticked in a particular trigger state, all child agents associated with that parent will be commanded to move towards the parent agent and re-embus. The 'Embussing Radius' option (see Figure 9(a)) defines the tolerance to which child agents are considered to have embussed after they reach the parent squad. Once child agents have re-embussed, the 'Release from Parent Squad' and 'Release Child Squads' check boxes become newly active, allowing child agents to debus in a different trigger state at a later stage.

Care is required when activating the 'Embus Children' feature. For example, if child agents happen to be too far from the parent when commanded to embus, they may become trapped behind obstacles along the way and be incapable of embussing. Furthermore, child agents should not be triggered to both embus and debus during the same time step.

There are several trigger states available to facilitate the implementation of debussing and embussing behaviour, as illustrated in Figure 9(b).

De-embussed Children: This trigger state applies to the parent squad. It allows different parent agent behaviour to be defined once all child agents have been released.

Released from Embus: This trigger state applies to child agents once they have been released from their parent. It allows unique agent behaviour to be defined after debussing; for example, taking up defensive positions.

Must Embus: This trigger state applies to all child squads once they have been commanded to embus. It allows the user to define behaviour to facilitate child agents moving towards the parent squad. For example, one would give the child agents a non-zero speed in this trigger state.

Embussed Children: This trigger state applies to the parent squad. It allows unique behaviour to be defined for the parent once it has embussed all its child agents; for example, drive on if the parent agent represents a vehicle.

Reach Dynamic Waypoint: This trigger state applies to child agents. It allows agent behaviour to be defined once they have fully debussed.

MANA-V includes settings to assist in defining the behaviour of child agents while they are debussing, illustrated in Figure 9(c). These settings are accessed from the 'Edit Squad Properties/Advanced' panel. If this personality weighting is non-zero, agents in a squad will debus towards equally distributed directions around the parent agent. The user can define how far the child agents will debus to and the general movement direction. This feature makes use of dynamic waypoints to influence agent movement. Hence, the 'Reach Dynamic Waypoint' trigger state can be utilized once child agents are fully debussed to implement a new set of behaviours. The 'Dynamic Waypoint Radius' option (see Figure 9(a)) defines the tolerance to which this trigger state is implemented. As an example, these features are useful for modelling infantry dismounting from vehicles during an urban patrol.

3.6 Path Finding

MANA-V includes a path finding algorithm to help agents navigate through a complex battle-space and avoid becoming trapped behind obstacles or terrain features. This feature is accessed from the 'Setup' pull-down menu, illustrated in Figure 10. If this option is selected, path finding calculations will be carried out whenever the scenario is loaded or after the terrain is modified (for example, using the 'Scenario Map Editor'). Depending on the size of the terrain, the path finding calculations may take several seconds to perform. However, once carried out, the path finding information is available to all agents in the scenario.

Path finding calculations are based on specific terrain types in a scenario. Paths are calculated towards best going, best concealment or best cover terrain features, as defined by the user. Agents will then be able to find their way towards these terrain types, taking the shortest route regardless of obstacles or terrain features in the way. Movement towards or away from specified terrain types are then set via the personality weightings 'Easy Going', 'Cover' and 'Concealment' under the 'Edit Squad Properties/Personalities' panel.

An example is given in Figure 11. Figure 11(a) illustrates a scenario where the yellow coloured road has been specified as the best going terrain in the scenario. A path

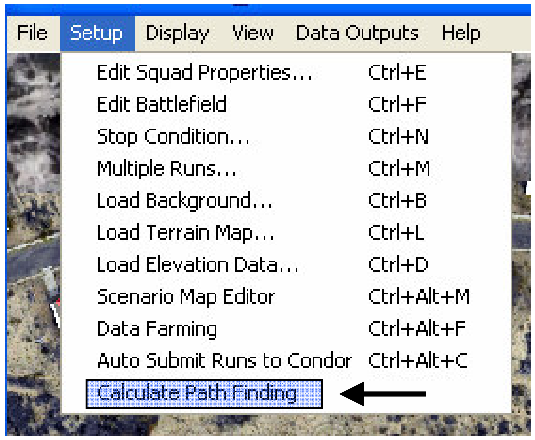


Figure 10: Calling up intelligent path finding calculations in MANA-V.

finding field has then been calculated, as illustrated in Figure 11(b). Here, the colour scale progresses from 'hotter' to 'colder' colours as agents approach the destination terrain. Effectively, the terrain field defines a gradient which agents follow down in order to reach their destination. The user can inspect the terrain field calculated for a particular scenario via the 'Terrain Field' option under the 'View' pull-down menu, as shown in Figure 11(b).

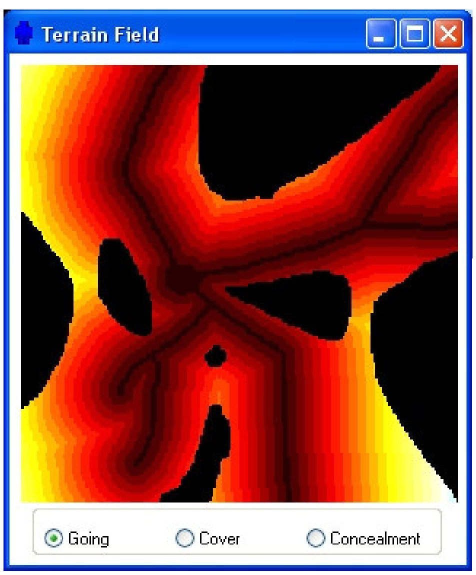
Huygen's wavelet principle has been utilized to calculate the terrain field. The destination terrain, specified by the user, acts as a source of outgoing waves. These waves propagate away from the destination terrain as illustrated in Figure 11(c), where each new wave-front is generated from the old wave-front via a series of equally sized wavelets. This method has the advantage that the wave-fronts naturally wrap around obstacles, and is analogous to the 'Flood Filling' method used for path finding in mazes involving a discrete, grid-based scheme.

This path finding method will not be suitable for more dynamic situations such as tracking enemies on the move. However, we have found it useful in a number of scenarios. For example, agents can be defined to:

1. Seize certain locations.
2. Take cover indoors.
3. Avoid certain areas.
4. Move to the nearest highway for improved mobility.
5. Move in and attack enemies travelling along a main road.
6. Agents acting as naval vessels can navigate along jagged coastlines without becoming trapped by peninsulas.

(a)

(b)



(c)

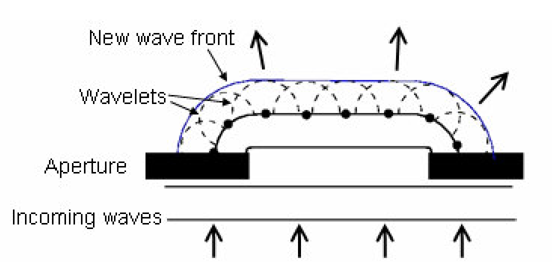


Figure 11: (a) An example scenario to illustrate path finding in MANA-V. Yellow roads have been defined as the best going terrain in this scenario. (b) Terrain field calculated for agents to move towards the road. Agents move down the gradient from brighter to duller colours, with movement strength specified by the 'Easy Going' personality weighting. (c) Illustrating the method for calculating terrain fields using Huygen's Principle.

3.7 Stop Conditions

The number of stop conditions available in MANA-V has been increased, as illustrated in Figure 12. Previous MANA versions only included stop conditions for either Blue or Red force reaching their final waypoint, or when particular numbers of casualties had been sustained. In MANA-V these stop conditions have been extended to include neutral agents. One can also define stop conditions for a particular squad.

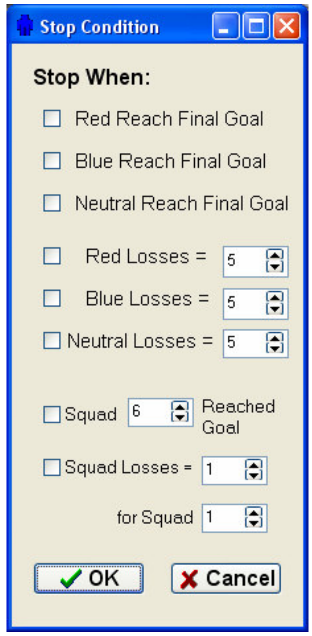


Figure 12: Various stop conditions available in MANA-V.

3.8 Collateral Damage

A collateral damage feature has been added to the MANA-V weapons model for direct fire weapons, as illustrated in Figure 13. This feature is useful for modelling civilian casualties in an urban environment. If the 'Allow Collateral Damage' check box is activated then any agent caught in the cross-fire between two other agents (see Figure 13(b)) will be shot. If the intervening agent is shot, the enemy agent originally intended to be shot will remain unharmed.

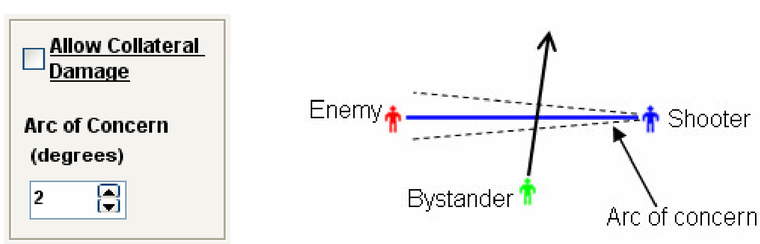


Figure 13: The collateral damage feature for direct fire weapons. Agents can be shot if caught in the cross-fire between two other agents.

The 'Arc of Concern' defines the tolerance to which the intervening agent will be shot. The calculation is normalized such that the overall probability of being shot within the Arc of Concern remains constant. Hence, a wide Arc of Concern would correspond to an inaccurate weapon, such that an agent would have less chance of being shot during a particular time step, but a reasonable chance of being shot overall while traversing the Arc of Concern.

3.9 Random Patrols

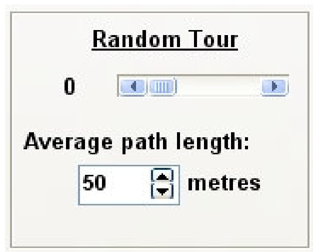


Figure 14: Setting random patrol routes for agents.

The random patrol feature added to MANA-V allows agents to travel random routes, and is useful for maritime scenarios where ships can be given random search patterns. It is implemented using dynamic waypoints. Once an agent reaches one waypoint, a new waypoint is randomly assigned. New waypoints are chosen such that the distance from the old waypoint follows an exponential distribution, $P(d) = \lambda\exp(-\lambda d)$. The user defines an average distance for each leg of the journey by setting the characteristic path length (λ in the exponential distribution). This setting is accessed under 'Edit Squad Properties/Advanced', as illustrated in Figure 14. The tolerance to which agents are defined as having reached a dynamic waypoint is specified by the 'Dynamic Waypoint Radius' setting in Figure 9(a).

3.10 Data Outputs

Data outputs remain mostly unchanged in MANA-V. Main exceptions include the 'Record Positions' data output file. Previous MANA versions write out each agent's ID, status (alive, injured or dead) and location at each time step. In MANA-V the agents' trigger states are also written out for every time step. This can allow extracting additional measures of effectiveness from this raw data, and pinpointing precise time-frames over which events occur on the battlefield.

The 'Record Casualty Location Data' output file has also been extended. As well as information regarding casualty locations, this output file now includes information on the shooter for each casualty: the shooter ID, location from which the shot was fired, the shooter squad ID and the particular weapon used.

3.11 Home Boxes

Home box and waypoint settings remain virtually the same as previous MANA versions. The main difference is that home box and waypoint locations can now take

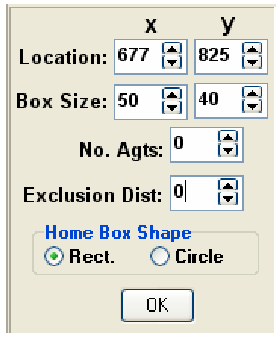


Figure 15: Agent home box settings.

on negative coordinates in accordance with the expanded nature of the battlefield in MANA-V. When changing settings under the ‘Edit Squad Properties/Map’ panel, the area that can be edited will match the view on the main MANA panel. If a different area of the battlefield is required for home box placement, this will need to be specified by clicking the appropriate location on the Mini Map and then re-opening the ‘Edit Squad Properties/Map’ panel. Alternatively, the home box and waypoint coordinates can be modified directly within the scenario’s XML file.

Another new home box feature in MANA-V is that agent home regions have the option of being defined as circular (see Figure 15). For circular home shapes, the ‘Box Size’ *x* and *y* settings still define the home region size. However, the larger of the two settings will set the outer radius of the circular home region while the smaller of the two settings will define an inner radius. This yields a doughnut shaped home region. For example, one could define a perimeter shaped home region by setting the *x* and *y* parameters almost equal. Such a home shape would be useful for setting enemies to appear at random locations surrounding a protected base. Similar to rectangular home box shapes, the mouse can be used to sweep out home regions on the edit map using circular shapes.

4 SUMMARY

This document has introduced a new version of MANA, called MANA-V, and given a description of new features added since MANA 4. Essentially, MANA-V dispenses with the grid-based movement scheme used in previous versions and uses a vector-based scheme for agent movement. This provides a number of advantages. Since agents are no longer restricted to a finite grid, they have greater freedom of movement and larger battlefield regions are able to be defined which can be panned and zoomed. For example, larger scenarios can be set up with concurrent operations occurring on different regions of the battlefield and which can be separately panned to and monitored.

The vector-based scheme allows battlefield distances, weapon ranges and agent speeds to be specified directly in terms of real-world measures such as metres, nautical miles or km/hr. Hence, parameters can be fed directly into the model without requiring conversion factors that were needed under the grid-based scheme. For example, weapons ranges and firing rates may be entered directly into the model from Jane’s weapons catalogues. From a military practitioner’s point of view, the model will appear less abstracted and allow easier visualization of a scenario's intent. Furthermore, scenarios can be set up with images from GIS data or Google Earth images, corresponding to real locations where operations might be carried out.

The vector-based scheme allows greater flexibility in terms of developing new model features. For example, we have been able to develop an effective path finding scheme which allows agents to travel to distant locations on the battlefield without becoming trapped behind obstacles. Other new features in MANA-V include: the ability to embus as well as debus agents, a number of new scan modes available for narrow FOV sensors and the ability to set up random patrols.

MANA-V also includes the genetic algorithm which was developed for MANA 4 [18]. The Data Analysis Tool and the human-in-the-loop feature have not been retained in MANA-V, since these features were developed for specific studies carried out at the time MANA 4 was being developed. Finally, we stress that MANA 4 remains a useful model if one does not require scenarios with expanded battlefields or real-world locations.

ACKNOWLEDGEMENTS

David Galligan and Michael Lauren are thanked for helpful discussions and suggestions during the development of MANA-V. The following people are also thanked for providing beta testing and feedback on the usability of MANA-V: Mark Anderson and Narelle Silwood of DTA and Marc West of DSTO.

REFERENCES

1. M. K. Lauren and R. T. Stephen, 'Map-aware Non-uniform Automata (MANA) a New Zealand Approach to Scenario Modelling', Journal of Battlefield Technology, Vol. 5, No. 1, 2002.
2. D. P. Galligan, M. A. Anderson and M. K. Lauren, 'MANA (Map Aware Non-Uniform Automata) Version 3 Users Manual', DTA Technical Note 2004/6, April 2005.
3. G. C. McIntosh, D. P. Galligan, M. A. Anderson and M. K. Lauren, 'MANA (Map Aware Non-uniform Automata) Version 4 User Manual', DTA Technical Note 2007/3, May 2007.
4. S. Y. Yiu, A. W. Gill and P. Shi, 'Using Agent Based Distillations to Model Civil Violence Management', Journal of Battlefield Technology, Vol. 6, No. 1, 2003.
5. D. P. Galligan, 'Operational Aspects of Imaging Radar Systems in Maritime Reconnaissance Aircraft', Journal of Battlefield Technology, Vol. 6, No. 3, 2003.
6. M. K. Lauren, 'Fractal Methods Applied to Describe Cellular Automaton Combat Models', Fractals, Vol. 9, No. 2, p177, 2001.
7. James Moffat, 'Complexity Theory and Network Centric Warfare', DoD Command and Control Research Program publications, 2003.
8. D. P. Galligan, 'Modelling Shared Situational Awareness Using the MANA Model', Journal of Battlefield Technology, Vol. 7, No. 3, 2004.
9. P. Dobias and S. Bassindale, 'Modeling Convoy Counter IED Tactics and Procedures in MANA: Can it be done?', Defence R&D Canada, Centre for Operational Research and Analysis, DRDC CORA TM 2007-48, November 2007.
10. See http://harvest.nps.edu for information regarding the International Data Farming Workshops. These workshops were previously run under the guise of Project Albert, http://www.projectalbert.org.
11. B. P. Abbott, 'Littoral Combat Ship (LCS) Mission Packages: Determining the Best Mix', Naval Postgraduate School, Monterey, 2008.
12. V. Lalis, 'Exploring Naval Tactics with UAVs in an Island Complex using Agent-Based Simulation', Naval Postgraduate School, Monterey, 2007.
13. V. Pfeiffer, 'Communication Aspects in Urban Terrain', Naval Postgraduate School, Monterey, 2006.
14. S. T. Terence Ho, 'Investigating Ground Swarm Robotics using Agent Based Simulation', Naval Postgraduate School, Monterey, 2006.
15. A. Ilachinski, 'Artificial War: Multiagent-Based Simulation of Combat', World Scientific Publishing, 2004.
16. M. K. Lauren, 'Fractals and Combat Modelling: Using MANA to Explore the Role of Entropy in Complexity Science', Fractals, Vol. 10, No. 4, 2002.

17. M. A. Anderson, M. K. Lauren, D. P. Galligan, 'Exploring the 2002 Seminar Games JANUS Wargaming Results with MANA', DTA Report 196, 2003.

18. G. C. McIntosh and M. K. Lauren, 'Genetic Algorithms Applied to Course-of-Action Development Using the MANA Agent-Based Model', Journal of Battlefield Technology, Vol. 9, No. 3, 2006.

19. M. A. Anderson and P. S. Mistry, 'Calculating Sensor Probabilities of Discrimination', DTA Report 203, 2004.

DOCUMENT CONTROL SHEET

1. ORIGINATING ACTIVITY Defence Technology Agency Auckland, New Zealand	2. RELEASE AUTHORISED BY
3. REPORT NUMBER DTA Technical Note 2009/7	4. CONTROL NUMBER NR 1525
5. DATE September 2009	6. NUMBER OF COPIES 10
7. SECURITY CLASSIFICATION UNCLASSIFIED	8. RELEASE LIMITATIONS
9. TITLE: MANA-V (MAP AWARE NON-UNIFORM AUTOMATA – VECTOR) SUPPLEMENTARY MANUAL	
10. AUTHOR(S) G C McIntosh	11. AUTOMATIC DOWNGRADING
12. COSATI CODES 12,15	
13. KEYWORDS EJC THESAURUS TERMS: Operations Research Simulation Modelling	NON-THESAURUS TERMS: Operations Analysis, MANA, ISAAC, Combat Model, EINSTein, Agent-based models, Cellular automata, Complex adaptive system.

14. ABSTRACT

MANA (Map Aware Non-Uniform Automata) is an agent-based distillation model developed by Defence Technology Agency (DTA) for use in military operations analysis studies. This document provides an overview of recent developments to the model, culminating in MANA version 5. In this version, the cell-based movement scheme of previous MANA versions has been replaced by a vector-based scheme. Advantages include: (i) larger battlefield regions can be defined which are able to be panned and zoomed, (ii) battlefield distances and agent speeds can be defined directly in terms of real world units (for example, metres, km/hr, nautical miles), (iii) sensor and weapons characteristics can also be specified directly using real world units, (iv) pre-defined libraries of sensors and weapons can be set up for importing into various scenarios without having to rescale their ranges to different battlefield sizes. Features new to MANA version 5 are described. Apart from changes to the movement algorithms, most features remain similar to MANA 4. The user is encouraged to consult the MANA 4 manual for a full description of all model features.

INITIAL DISTRIBUTION

	No. of Copies
NEW ZEALAND	
Director, DTA	2
CAD Branch, Trentham Army Camp	1
Director, NZ Army Simulation Centre, Linton Military Camp	1
Spares	1
AUSTRALIA	
DSTO Research Library, Edinburgh	1
CANADA	
Director Operational Research (Joint and Land), National Defence Headquarters, 101 Colonel By Drive, Ottawa, Ontario, Canada K1A OK2	1
DRDKIM, Ottawa	1
UNITED KINGDOM	
DSTL Knowledge Services, Glasgow	1
USA	
SEED Center for Data Farming, Naval Postgraduate School, Monterey	1