Enhanced Behavioral Realism for Live Fire Targets

Rick Evertsz, Andrew Lucas, Cameron Smith, Matteo Pedrotti
AOS Group
580 Elizabeth Street, Melbourne, VIC 3000, Australia
{rick.evertsz, andrew.lucas, cameron.smith, matteo.pedrotti}@aosgrp.com

Frank E. Ritter
College of Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802
frank.ritter@psu.edu

Rob Baker
Defence Science and Technology Organisation
PO Box 1500, Edinburgh, SA 5111, Australia
robin.baker@defence.gov.au

Paul Burns
Australian Target Systems
161 Fallon Street, Albury, NSW 2640, Australia
pburns@atspl.com.au

Keywords:
Autonomy, BDI, Cognitive Architecture, Human Behavior Modeling, Live Fire Targets, Robots

ABSTRACT: *Live fire training is an essential component of infantry and Special Forces training. Recent developments in target technology have created a need for more sophisticated human behavior models that can drive the targets to behave in a realistic and challenging manner. Such behavior models enable the targets to exhibit convincing tactical behavior, as well as coordinating as a team to confront the trainee with a more formidable foe. This paper describes the current status of an ongoing project to augment disparate target types with sophisticated behavioral capabilities. The underlying CoJACK™ behavior engine enables the deployment of targets that behave realistically, react in a timely fashion, and exhibit sufficient variation to ensure that the trainees cannot predict how opponents will behave. The behavior models are developed using the VBS2 environment so that they can be validated in advance of target deployment on the targets. The technological approach is presented along with two illustrative scenarios. We conclude with a discussion of the lessons learnt and the way forward, including the development of autonomous target vehicles.*

1. Introduction

The firing of live ammunition is a core part of infantry training. Indeed, Special Forces engage in live fire training on a daily basis to ensure that they are always at the peak of their capability. Lately, live fire training technology has expanded from simple static, pop-up or rail-based pop-up targets to indoor, projection-based systems, and more recently, fast moving, open range, mobile robotic targets. Each type of target supports particular aspects of live fire training. For example, static pop-up and rail-based targets are good for basic skills training. However, they are less effective for advanced training because target location is too predictable.

With the advent of 3D, photorealistic virtual environments, projection-based systems have been developed that provide a more immersive experience in indoor, cinema ranges (Pair & Treskunov, 2006). Although providing significantly better immersion than pop-up targets, they are restricted to indoor environments, the image quality is limited (Darken & Jones, 2007), and 2D projections do not faithfully reproduce parallax effects; this means, for example, that a trainee cannot bring a concealed adversary into view by moving laterally.

Over the last 20 years, the Australian Army has developed wheeled robotic targets, and the latest generation is being deployed on their live fire ranges. On flat, smooth surfaces (e.g., concrete) the targets are fast moving and maneuverable, and can be challenging to hit. However, they have limited autonomous capabilities, such as path finding, and can only exhibit simple behaviors such as scattering when one of the robots is hit.

More advanced training scenarios require a much wider and more responsive behavioral repertoire than that provided by current robotic targets. Furthermore, greater variability in robot behavior is required to expose advanced warfighters to less predictable training scenarios.

There is also a need to integrate the various target types into a coordinated whole controlled by a single Range Management System. This approach ensures that the targets fulfill their respective roles in the training scenario while exhibiting a level of behavioral realism commensurate with their physical capabilities. Squad tactics can be implemented by coordinating the targets, for example, using a pop-up to draw fire just before a mobile robotic target moves to the next cover position. Broader behavior modeling also enables the simulation of different levels of tactical capability, for example those pertaining to militia, insurgents or well-trained infantry.

This paper reports on the current state of implementation of an ongoing project (the "AI Project") with the Australian Defence Force to augment live fire targets with sophisticated tactics that provide warfighters with practical experience countering different types of adversary that exhibit variability in tactics and response. The system also includes a VBS2-based simulation environment allowing scenarios and tactics to be tested before deploying the robot targets.

The target behaviors are implemented on the BDI (Beliefs, Desires, Intentions) cognitive architecture, CoJACK™ (Evertsz, Ritter, Busetta & Pedrotti, 2008; Ritter et al., 2012), which supports reactive and proactive decision-making in dynamic environments. CoJACK, is underpinned by a sub-symbolic layer that produces principled variation in reasoning. Variation is also produced by a moderator layer that simulates the effect of emotion and other non-rational factors on decision-making.

This work represents a novel application of human behavior modeling to address the problems of flexible, autonomous control of multiple types of live fire target. The combination of sophisticated behavior models with live fire targets improves training effectiveness by facilitating the execution of advanced training scenarios while releasing the Range Manager to focus on the overall training goals rather than lower level target control, thereby reducing his workload.

2. Background

The most common approach to human behavior modeling in military applications is to use the scripting language provided by the synthetic environment, cf. JSAF (Ceranowicz & Torpey, 2005). This approach has proven inadequate for representing the variability inherent in real human behavior. Scripts tend to be inflexible and brittle, and trainees soon exploit their shortcomings, leading to diminished training value. Trainee exploitation of the inflexibility of scripted behavior also occurs with live fire training; for example, if the robots always scatter when one of their members is hit, the shooter can anticipate their reaction and be ready to pick them off. The shortcomings of scripted approaches to human behavior modeling have led to the exploration of more flexible representational paradigms, such as BDI, which is specifically designed for applications where the environment is constantly changing, requiring an adaptive mix of reactive and proactive behavior.

2.1 BDI Agents

The BDI paradigm (Wooldridge, 2000) stems from work on human practical reasoning by the philosopher Michael Bratman (Bratman, Israel & Pollack, 1988). In this view, rational agents have **beliefs** about the world, **desires** that they would like to achieve and **intentions** that they are committed to. This has proven to be an intuitive and powerful computational abstraction for describing, designing and implementing complex, intelligent systems.

A typical BDI agent executes a control loop in which it updates its beliefs to reflect the current state of the world, deliberates about what to achieve next, finds a plan for doing so, and executes a step of that plan. Each time around this cycle, it reconsiders its options to reflect any changes in the environment, and can change tack if a more pressing need arises.

This control structure allows BDI agents to deal with a rapidly changing environment without getting locked into a particular "train of thought" or sequence of actions. The first published application of the BDI paradigm to military simulation was in the air combat domain (Murray et al., 1995).

Although the BDI paradigm is a very effective computational abstraction for representing intentional agents, it is silent with regard to the inherent variability of human behavior. The prediction of biases, timing and errors in human performance is the province of cognitive architectures.

2.2 Cognitive Architectures

Cognitive architectures, such as ACT-R (Anderson, 2007), define the structural properties of the human cognitive system, that is, the information processing mechanisms that are invariant across tasks. These mechanisms predict the timing of, and errors in, human performance across a wide range of cognitive tasks.

Although most applications of cognitive architectures occur in laboratory settings, there are a number of studies in military domains, including air combat tactics using Soar (Tambe et al., 1995), submarine decision-making using ACT-R (Fleetwood, Santoro & Severinghaus, 2007), and suicide bomber behavior using CoJACK (Evertsz, Pedrotti, Busetta, Acar & Ritter, 2009).

2.3 Performance Moderators

Whereas a cognitive architecture such as ACT-R or CoJACK predicts how performance varies as a result of sub-symbolic, cognitive parameters such as memory retrieval threshold and latency, human decision-making is also influenced by affective factors such as emotion. Even well trained military personnel can behave irrationally under extreme stress and consequently good military strategy seeks to manipulate the emotions of the adversary. The modeling of affect in military simulation has been explored and applied most widely in the work of Silverman (2004), using a Performance Moderator Function server that cognitive architectures can interface to. An alternative approach, taken by CoJACK, is to tightly integrate moderators within the cognitive architecture itself so that they directly affect the architecture's cognitive parameters. The former approach offers greater interoperability, but the latter has the potential to be more computationally efficient which is important for real-time applications containing many cognitive entities.

2.4 CoJACK

CoJACK was selected as the project's behavior modeling platform because it provides a cognitive architecture, moderators and is highly optimized, providing the rapid response required to control multiple targets in real time. We briefly describe CoJACK below; further detail can be found in previous publications (Evertsz et al., 2008; Ritter et al., 2012).

CoJACK is the result of a 5-year, UK Ministry of Defence[1] effort to improve human behavior modeling and address perceived shortcomings in the usability of ACT-R and Soar (Newell, 1990). BDI languages represent behavioral constructs at a higher level of abstraction than production-rule based architectures like ACT-R and Soar. CoJACK extends the BDI language, JACK® (Winikoff, 2005), with the cognitive parameters of ACT-R and a moderator layer that supports the representation of emotional factors, such as fear, and physical factors such as fatigue.

CoJACK represents and executes tactics in a manner that Subject Matter Experts (SMEs) find easy to relate to, due to their representation at a higher level of abstraction. CoJACK's tactics representation includes a front-end that allows analysts to specify tactics graphically at a high-level without having to worry about low-level detail. The graphical representation is amenable to inspection by SMEs and thereby supports verification and validation of behavior.

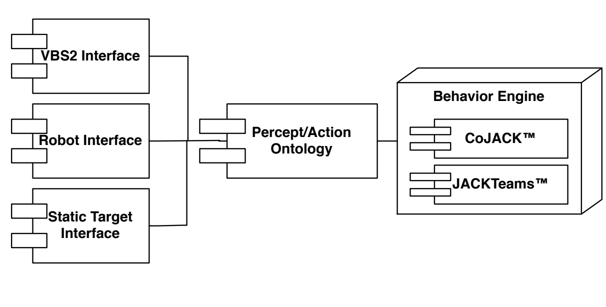
CoJACK has been used to model tank commander behavior, rules of engagement (Evertsz, Ritter, Russell & Shepherdson, 2007), terrorism (Evertsz et al., 2009) and counterterrorism scenarios (Evertsz, Pedrotti & Glover, 2010), as well as the domain described in this paper.

3. Technical Overview

The goal of the project is to augment multiple live fire target types with sophisticated behavioral capabilities that increase training effectiveness and reduce the Range Manager's workload. To this end, we developed an architecture in which CoJACK acts as a cognition/emotion server that receives perceptual input from sensors on the training range and sends behavioral action commands to the targets. The behavior models are developed and tested within a simulation environment (VBS2) before deployment on the training range. This section outlines the current status of the implementation.

3.1 High Level System Architecture

The overall system architecture is shown in Figure 1. The Behavior Engine comprises CoJACK and a JACKTeams™ extension supporting high-level specifications of coordinated target behavior. Percepts and actions are mediated through a domain-specific ontology that provides the Behavior Engine with an abstract representation of the available types of percept and action. The target-specific interfaces map between the percept/action ontology and the API (Application Programming Interface) of the target types. For example, the VBS2 interface currently uses ASI (Application Scripting Interface) to interact with VBS2. Each robotic target is represented as a BDI agent within the behavior engine.



[1] The Improved Human Behaviour Representation (IHBR) Project (Ministry of Defence Project: RT/COM/3/006).

Figure 1 – System Architecture

The ontology defines the target types and their salient properties. For example, the ontology will specify the maximum speed of a given type of robot target, and would also specify that a static target cannot move. This constrains the available behaviors for that target type in addition to the roles on a team that the target can fulfill. Although we are currently using an internally developed ontology language, we expect to move to a standards-based language, e.g., OWL[2] (Web Ontology Language) DL, to accommodate a larger ontology. OWL DL allows merging of independently developed ontologies and automatic inference of attributes and relationships.

The scenario definition file specifies the initial exercise setup, for example, the types of target, their location, team specifications and the initial goals of the target entities (Figure 2). This enables the Behavior Engine to select a plan of action that complies with the constraints of the given target type. Each of these scenario modifiers can be altered at runtime to vary the training in progress.

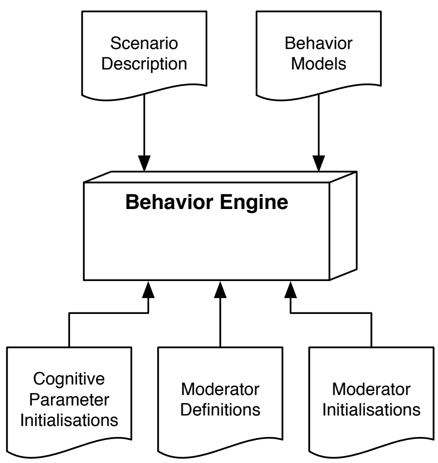


Figure 2 – Behavior Engine Initialization

New target types are under development, in particular an autonomous vehicle that may include a driver, commander and gunner working as a team. The modular architecture (Figure 1), and in particular the ontology, allows this to occur with minimal change to the rest of the system.

3.2 Behavior Models

Individual target behavior is expressed in CoJACK. JACKTeams handles inter-target coordination. A given CoJACK behavior model comprises:

- A specification of situations it can respond to (external **events**, internal **goals**, overall world state).
- A description of the various ways of responding to a given situation (expressed as **plans**).
- A knowledge base of what the entity knows (represented as a **belief set**).
- Underlying cognitive parameters that affect how the model runs, leading to predictions of "thinking time" and errors that can occur. This is the fundamental way that cognitive variation occurs in CoJACK models.
- Moderators representing factors such as fear and morale, and a specification of how those moderators affect cognitive parameters and decision-making.

Procedural reasoning is represented in terms of **plans**. These can be graphical, as shown in the example in Figure 3, and this helps with validation by SMEs.

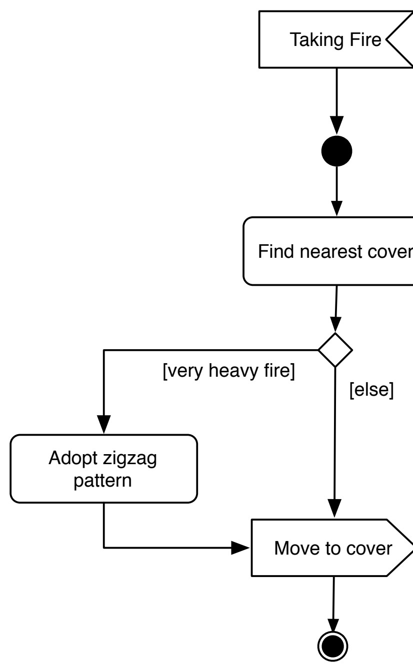


Figure 3 – Plan to Take Cover

The graphical plan, **Take Cover** (Figure 3), applies when a **Taking Fire** percept is received. It finds the nearest cover and, if taking very heavy fire, adopts a zigzag pattern before performing the action **Move to cover**.

3.3 Perception and Action

The various target types, including those simulated in VBS2, communicate with the system via relatively high-level percept and action descriptors. The Behavior Engine does not have access to raw visual data or fine level motor control of the targets. This is typical in

---

2 http://www.w3.org/TR/owl-features/

virtual environments, such as VBS2, but also applies to robots in general. In the case of robots, the robot is sent high-level commands, such as **Change Heading**, and it is left to the robot's control software to map this high level command to coordinated triggering of its low-level actuators.

3.4 VBS2 Trial Environment

A very important aspect of the system is the ability to develop and test behavior models in a simulation environment in advance of deployment on hardware targets. This not only helps with model validation, but also facilitates model development and debugging, and allows the utility of new target types to be evaluated in advance of their acquisition. It also enables trainees to refine their decision-making skills before facing the actual robots on the range.

The Percept/Action ontology insulates the behavior models from the majority of the specifics of VBS2. Having said that, VBS2's own AI can sometimes interfere with the commands sent by the Behavior Engine, even if the VBS2 AI has been disabled. For example, if a VBS2 actor is wounded and the Behavior Engine commands it to move, it will drop to the ground and start crawling. Nevertheless, in the vast majority of cases, the VBS2 actors can be controlled sufficiently well to evaluate the behavior models.

VBS2 provides a wide range of high-level percepts and actions, including location, orientation, speed and path finding. It also includes static pop-up target models.

3.5 Mobile Robot Interface

Two types of mobile target are currently being used for live fire training, 2-wheeled and 4-wheeled, with the latter suited to rougher terrain. Both are controlled via the same Robot Interface. The Robot Interface provides access to high-level robot actions such as **Move To Position** within some tolerance and at a particular speed, **Change Heading** within some tolerance, and **Stop**. The Robot Interface responds with whether the command was executed, and returns its current status, for example if the robot's body has been lowered to signify that it has been killed. Transient percepts are also provided, including position and velocity updates, and a notification if the robot's hit sensor detects that it has been shot.

3.6 Static Target Interface

The Static Target Interface not only controls the static pop-up targets, it also provides a target amalgamation function (this functionality is mirrored in the VBS2 Target Interface). Target amalgamation treats a group of targets as a single virtual entity. For example, when the Behavior Engine sends a command for a virtual entity to move forwards, this gets reified as a consecutive sequence of up/down state changes of the pop-up targets that lie between the start and end positions of the virtual entity. From the trainee's perspective, it looks as if a single entity pops up, then crouches down and moves (concealed) to the next position before popping up again, and so on, until it reaches its final position. Entity retreat is expressed by reversing the sequence.

The Static Target Interface also manages the speed of pop-up sequencing to match the movement rate specified by the Behavior Engine.

3.7 Teamed Behavior

Modeling team behavior is an essential requirement in military domains – coordination amongst unit members is key to success. The usual approach to team modeling is to script coordination at the level of individual team members so that the team behavior *emerges* from the interaction of the individual team members. This approach is unsatisfactory because individual and team tactics become intertwined, making them difficult to modify.

JACKTeams (Jarvis, Jarvis & Jain, 2007) is an extension of the BDI paradigm that separates team tactics from individual ones. A team is modeled as a separate reasoning entity with coordinated activity defined at the level of the team. This generic team-based capability provides a flexible basis upon which a wide variety of teaming algorithms can be implemented.

At runtime, the team is formed by assigning individual targets to relevant team **roles**. A team definition specifies which roles must be filled, i.e. what capabilities the entities must have to fill the role. Team formation is triggered automatically as part of the team instance construction. Role fillers (which can be either individuals or sub-teams) can be detached and attached at runtime, thereby supporting dynamic team formation and re-formation. This is important when a target is neutralized and its role needs to be filled by another entity, for example if the neutralized target was the team leader.

Team execution includes a propagation step that handles dissemination of information up and down the team hierarchy. The team has access to propagated information that is derived from the knowledge of its sub-teams. JACKTeams includes filters that determine if and when the propagation should occur, and which subset should be propagated to the encompassing team. Similarly, sub-teams can inherit a synthesized subset of the knowledge of the containing team.

3.8 Range Manager Interface

Increased autonomy is an essential prerequisite for reducing Range Manager workload during high tempo exercises because it requires a lot of attention to control the targets individually. Nevertheless, exercises do not always play out as expected – human participants will react differently from exercise to exercise, depending on experience and other factors. Consequently, it is vital that the Range Manager retains control of target behavior to take advantage of unexpected pedagogical opportunities or to prevent any unsafe incidents occurring.

To this end, a tablet-based GUI has been developed that enables the Range Manager to issue behavior overrides. These are specified at a high level, for example, “retreat” or “move to cover” rather than “move to point X,Y”. Behavior overrides include:

- assign a new goal to a target or target team (e.g., attack),
- command a target or target team to adopt a specified tactic (e.g., assault using bounding overwatch), and
- alter moderator levels for a target or group of targets (e.g., their morale level).

The Range Manager Interface forms one part of the overall Range Management System, currently under development. The first phase of this Range Management System has been tested extensively under live fire conditions, demonstrated to stakeholders and successfully passed acceptance testing.

4. Illustrative Scenarios

This section outlines a couple of example training scenarios using pop-up and mobile robot targets. Because the actual tactics and scenarios are classified, we have fabricated examples using information available in the public domain (Larsen, 2005). Although these fabricated examples cannot be used to infer the actual tactics and scenarios used in the project, they have similar properties and can be taken to be indicative of the types of tactics/scenarios implemented.

4.1 Frontal Assault – Militia

In a frontal assault, the targets (whether robots or sequenced pop-ups) move towards the location of the trainees, firing their weapons. Currently, robot weapon firing is implemented via the sound of machine gun fire emanating from speakers on the robots. The Range Manager makes a judgment call as to whether a trainee could have been hit by one of the robots. However, laser based “shoot back” systems and instrumented systems can be incorporated on the targets to provide a more objective means of assessment.

The speed of the attack and use of cover depends on the interaction between robot morale and fear levels. Loosely, very high morale masks fear and leads the robots to attack rapidly, making minimal use of cover. A state of high fear and low morale leads the robots to move more tentatively, making frequent use of cover and remaining concealed for longer periods of time. Fear rises and morale drops if robot team members are being “killed”, and the robots may decide to panic and retreat if their morale drops very low due to high losses. The Range Manager can manipulate the morale and fear levels through the GUI to increase or decrease the level of pressure on the trainees.

Militia models represent relatively untrained and inexperienced fighters. Inter-robot coordination is poor and the tactics are simplistic and relatively ineffective (as compared to trained insurgents and infantry). Rogue behaviors can be included, for example, having one robot stand its ground under heavy fire when one would expect it to retreat. This simulates the behavior of an over-zealous militiaman who is prepared to sacrifice himself to inflict losses on the enemy. Other variations in individual parameters lead to behaviors such as zigzagging during the attack, or accelerating and decelerating unpredictably.

4.2 Bounding Overwatch – Insurgents

This is a more sophisticated form of frontal assault and is employed by teams of fighters with more training. In this team tactic, one fire team provides fire support while the other (the “bounding element”) advances. The fire support team then advances once the other team has reached cover and has begun firing. The two teams are usually laterally separated relative to the enemy.

A number of variations are possible. If morale is very high, the bounding element may forego available concealment. Errors in plan execution, due to particular values of cognitive parameters, can cause the bounding element to stray into the line of fire of the fire support team.

5. Discussion and Conclusions

This paper introduces a method of integrating and augmenting live fire targets with sophisticated tactical behaviors. The resulting enhanced targets offer more realistic training while reducing the Range Manager’s workload and saving on range support manpower.

The evaluation of the current implementation is at an early stage; nevertheless some preliminary conclusions can be drawn:

- CoJACK-driven robot behavior provides a significant enhancement over the previously scripted approach.
- Variation in the values of CoJACK's cognitive parameters provides a richer training experience because the robots do not behave in exactly the same way every time.
- Providing the Range Manager with a straightforward means of overriding robot behavior is a key feature of the system. The default mode is for the robots to behave autonomously, but at times it is essential for the Range Manager to redirect their behavior during an exercise.

Although a wide range of adversarial behaviors can be implemented and used to control live fire targets, some are infeasible due to current limitations in robot capabilities. For example, the robots cannot detect near misses, nor do they know the exact location of the trainees, even if they are in view. These limitations are being addressed as part of Special Operations Command, a five-year plan for improved target capability and integrated range management.

Behavior expression can sometimes be compromised by the robots' inbuilt, low-level behaviors such as needing to circle when they become uncertain of their position. Similarly, VBS2 can sometimes execute actions independently of the commands sent from the Behavior Engine. This problem is not unique to VBS2 and needs to be addressed if behavior models are to be faithfully expressed by the platform models they are driving.

We are also investigating the application of Cognitive Tutor Authoring Tools to teach Point of Aim determination for hitting moving targets

5.1 Addition of Autonomous Vehicle Targets

As an extension to the current robots, which represent human actors, both adversaries and civilians, there is a requirement for autonomous vehicle targets that are suitable for heavy weapons training, sniper training, convoy ambush training and demonstrations. An autonomous vehicle target that is capable of withstanding heaving weapons fire will be a valuable addition to the already existing target systems and consequently will enhance the realism and complexity of training scenarios and demonstrations.

The design and integration of sensor systems, control systems, drive systems and vehicle platforms require careful consideration of the specific purpose of the target system and vulnerabilities of the components because the target systems are likely to sustain damage during employment. Many of the components are available off the shelf, as automated vehicles exist in various forms in various applications such as heavy haul trucks in mining operations.

Future development of the autonomous vehicle target offers the potential to include CoJACK-based actors in the vehicle. For example, an unarmored SUV 4x4 might include a driver, rear gunner and a commander, working as a team. This will present the trainees with the option to disable the vehicle or focus on the driver and/or the gunner, thus reinforcing the requirement for split-second decision-making.

5.3 Summary

This paper presents a novel application of human behavior modeling to provide flexible, autonomous control of multiple types of live fire target. The approach integrates disparate target types into coordinated teams that exhibit group tactics.

The use of the moderated cognitive architecture, CoJACK, provides real time performance and principled variation in target behavior, with JACKTeams providing an important group behavior capability. This approach supports the implementation of more advanced training scenarios.

The system reduces Range Manager workload and provides the basis for an overarching Range Management System that will control all the target types in an integrated manner and substantially reduce range support workload and costs.

6. Acknowledgements

The authors acknowledge the support of:

- **MAJ BRUCE HUGHES**: Project Director, Special Training Facilities, Defence Support Central and West.
- **MR PAUL NEVILLE**: Estate Development Officer (SO), Defence Support Central and West.

The cognitive tutor work is supported under N00014-10-C-0281 / N091-086/P10008.

7. References

Anderson, J. R. (2007). *How can the human mind exist in the physical universe?* New York, NY: OUP.

Bratman, M. E., Israel, D., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, *4*(4), 349-355.

Ceranowicz, A. & Torpey, M. (2005). Adapting to urban warfare. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 2(1), 3-15.

Darken, C. J. (2007). Computer graphics-based models of target detection: Algorithms, comparison to human performance, and failure modes. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology,* 4(3), 262-276.

Evertsz, R., Pedrotti, M., & Glover, W. (2010). Realistic virtual actors for training in counterterrorism. In *Proceedings of SimTecT 2010.*

Evertsz, R., Pedrotti, M., Busetta, P., Acar, H., & Ritter, F. E. (2009). Populating VBS2 with realistic virtual actors. In *Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation.* 09-BRIMS-04.

Evertsz, R., Ritter, F. E., Busetta, P., & Pedrotti, M. (2008). Realistic behaviour variation in a bdi-based cognitive architecture. In *Proceedings of SimTecT 2008.*

Evertsz, R., Ritter, F. E., Russell, S., & Shepherdson, D. (2007). Modeling rules of engagement in computer generated forces. In *Proceedings of the 16th Conference on Behavior Representation in Modeling and Simulation.* 123-134, Orlando, FL: U. of Central Florida.

Fleetwood, M. D., Santoro, T. P., & Severinghaus, R. J. (2007). An ACT-R model of an approach officer in a modified periscope depth task. In *Proceedings of the Behavior Representation in Modeling and Simulation (BRIMS) Conference.*

Jarvis, B., Jarvis, D., & Jain, L. (2007). Teams in multi-agent systems. In Z. Shi, K. Shimohara, & D. Feng (Eds.), *Intelligent Information Processing III* (Z. Shi, K. Shimohara & D. Feng, Eds.). Springer US.

Larsen, C. (2005). *Light Infantry Tactics: For Small Teams.* Bloomington, IN: AuthorHouse.

Murray, G., Steuart, D., Appla, D., McIlroy, D., Heinze, C., Cross, M., et al. (1995). The challenge of whole air mission modelling. In *Proceedings of the Australian Joint Conference on Artificial Intelligence.* Melbourne, Australia.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Pair, J. & Treskunov, A. (2006). Projector-Camera Systems for Immersive Training. University Of Southern California Marina Del Rey, CA, Inst For Creative Technologies.

Ritter, F. E., Bittner, J. L., Kase, S. E., Evertsz, R., Pedrotti, M., & Busetta, P. (2012). CoJACK: A high-level cognitive architecture with demonstrations of moderators, variability, and implications for situation awareness. *Biologically Inspired Cognitive Architectures*. 1(1). 2-13.

Ritter, F. E., Yeh, K.-C., Cohen, M. A., Weyhrauch, P., & Kim, J. W. (in press). Declarative to Procedural tutors: A family of cognitive architecture-based tutors (Special Issue on best of BRIMS 2013). *Computational and Mathematical and Organizational Theory*

Silverman, B. G. (2004). Human performance simulation. In J. W. Ness, D. R. Ritzer, & V. Tepe (Eds.), *The science and simulation of human performance.* (pp. 469-98). Amsterdam: Elsevier.

Tambe, M., Johnson, W. L., Jones, R. M., Koss, F., Laird, J. E., Rosenbloom, P. S., et al. (1995). Intelligent agents for interactive simulation environments. *AI Magazine*, 16(1), 15-40.

Winikoff, M. (2005). JACK™ intelligent agents: An industrial strength platform. In *Multi-Agent Programming.* Springer.

Wooldridge, M. (2000). *Reasoning about rational agents.* Cambridge, MA: MIT press.

Author Biographies

RICK EVERTSZ is a cognitive scientist at AOS Group, interested in the incorporation of cognitive and affective constraints into BDI agents.

ANDREW LUCAS is the Managing Director of AOS Group, Melbourne, Australia.

CAMERON SMITH is responsible for the development and deployment of the advanced behaviors on the mobile robots.

MATTEO PEDROTTI is working on the integration of CoJACK with virtual environments.

FRANK RITTER is on the faculty of the College of IST, an interdisciplinary academic unit at Penn State to study how people process information using technology.

ROB BAKER is Senior Special Operations Science and Technology Adviser – West, DSTO.

PAUL BURNS is an ex-commander of the Australian Special Air Service Regiment and a Company Director of Australian Target Systems, Albury, Australia.