



way how behaviour varies as a result of physiological and affective factors (e.g. fear). CoJACK is unique in its synthesis of a BDI representation with a cognitive architecture. Whereas cognitive architectures (e.g. ACT-R, Anderson, 2007) typically represent procedural knowledge in terms of fine-grained steps, CoJACK offers a high-level plan representation underpinned by sub-symbolic computations that influence processing without obscuring that high-level viewpoint.

## 2.1 Decision Making

In all but the most trivial cases, an agent will have a number of alternative actions available to deal with the current situation. The mental process of selecting an action to perform is termed **decision making**. This process can be either rational or irrational, and when the pressure is on, can be less rational with less thought as to the overall consequences.

Many factors can contribute to a decision, including:

- Situation Awareness – the appreciation of the *relevant* aspects of the current situation.
- Predictive Capability – the ability of the agent to foresee the consequences of actions and the likely actions/reactions of other entities that are part of the scenario.
- Response Repertoire – the known action sequences for dealing with the current situation (skill set).
- Personal Preference – preferred methods of dealing with the current situation, often based on experience of previous successes and failures.
- Cognitive Effectiveness – the current state of the underlying cognitive architecture, affecting capabilities such as ability to recall facts, hold intermediate results in working memory, and stay focused on the problem.
- Affective State – the emotional factors that can influence a decision, for example, a high-level of fear can predispose a person to make an irrational decision.

These factors contribute to individual differences in decision making and should be reflected in virtual actors. Typically, virtual actors are implemented in relatively inflexible scripting languages. In contrast, CoJACK offers a high-level plan language that takes account of the six decision making factors listed above.

## 2.2 Knowledge Representation

CoJACK uses a graphical **plan** representation to encode the agent’s reasoning capability (procedural knowledge). A graphical plan (shown in Figure 1) defines the context in which it is applicable and the various steps that must be followed for it to deal with the situation.

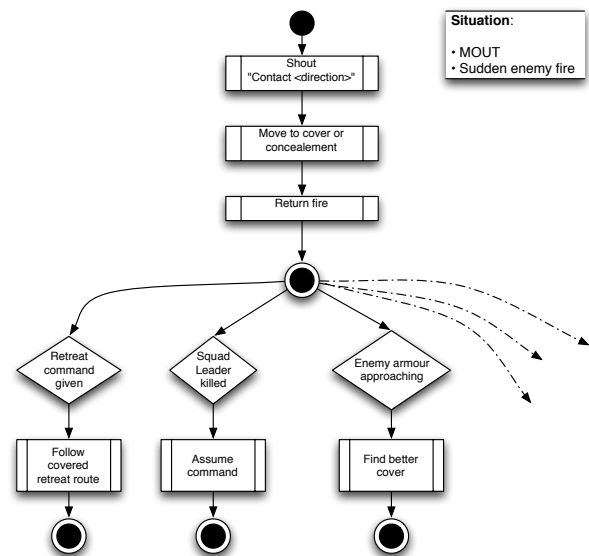


Figure 1. Graphical plan in CoJACK.

Plans are triggered in response to events the agent perceives in its environment, but also as a result of goals that the agents adopts. For example, if under heavy fire, an agent might adopt the goal to find cover. This goal would remain active until the agent succeeds or gives up. If the agent has a repertoire of plans that can be used to achieve this goal, it will try each in turn until it has found cover.

In addition to its procedural knowledge (“knowing how to do something”), CoJACK agents have declarative knowledge (“knowing about”). In Figure 1, knowing that the squad leader has been killed is an example of declarative knowledge. Declarative knowledge is stored in **beliefsets** within the agent.

## 2.3 Underlying Architecture

CoJACK simulates the *structural properties* of the human cognitive system, that is, the information processing mechanisms that are fixed across tasks (Newell, 1990). It is generally accepted that humans share a common cognitive architecture – CoJACK embodies such commonalities and, in part, expresses the individual variation as differences in the values of the agent’s internal (cognitive) parameters.

Memory is stored in declarative knowledge **chunks**, chunks have an activation level that can be propagated between associated chunks. CoJACK has parameters that can vary, leading to individual differences in behaviour (Evertsz et al., 2008). Parameters can delay the agent’s reasoning steps and accelerate/decelerate memory access. Parameters can also affect which memory elements (chunks) are retrieved/forgotten, and because chunks are used to represent the agent’s current activity, the agent can forget what it was doing.

## 2.4 Physiology and Affect

Physical and affective factors can have a significant impact on behaviour. For example, a drug as mundane as caffeine has been shown to reduce reaction time and increase the ability to focus, allowing performance on a vigilance task to remain virtually sustained at its original level, instead of decreasing over the span of an hour (Boff & Lincoln, 1988). Emotions, such as fear, affect decision making and can result in irrational choices. Though irrational, these behavioural tendencies have been honed over millions of years of evolution and are so deeply entrenched even extensive training can be ignored when the emotional drive is strong enough. Consequently, these factors need to be taken into account if we are to build realistic models of human behaviour.

A number of architectures have sought to address emotion and its interaction with cognition. PSI is a cognitive architecture designed to integrate cognitive processes, emotions, and motivation (Bach, in press; Bartl & Dörner, 1998). The architecture includes six drives (needs for energy, water, pain avoidance, affiliation, certainty, and competence). Cognition is modulated by these motive/emotional states and their processes.

PMFServ was conceived as a software product that would expose a large library of well established and data-grounded Performance Moderator Functions (PMFs) and Human Behavior Representations (HBRs) for use by cognitive architectures deployed in a variety of simulation environments (Silverman et al., 2000). It has now grown into a more complete architecture. PMFServ uses reservoirs of various resources to provide for different affective moderators and mental states. The principal feature of PMFServ is that it models human decision-making based on emotional subjective utility constrained by stress and physiology (Silverman, 2004).

Several other models of emotions and architectures that use emotions have been created (e.g. Gratch & Marsella, 2004; Hudlicka, 2002). Reviews of emotional models (Hudlicka & Fellous, 1996; Picard, 1997) typically present models and architectures that have not been compared and validated against human data. There appears to be one exception, an unpublished PhD thesis by Araujo at Sussex (cited in Picard, 1997). Several researchers are attempting to add several simple emotions and moderators to ACT-R (Ritter, Reifers, Schoelles, & Klein, 2007; Belavkin et al., 1999) and validate the model by comparing the revised model with an existing model and comparable data (Kase, Ritter, & Schoelles, 2008).

In CoJACK, emotional/physical factors are modelled using moderators: time-based functions applied to cognitive parameters. These can have internal reservoirs with decay functions that define how the reservoir level changes over time. Moderator inputs can comprise events from the simulation environment and events that are internal to the agent.

### 3. VBS2

VBS2 ([www.vbs2.com](http://www.vbs2.com)) is a state-of-the-art 3D battlefield simulation environment, with a broad suite of tools including After-Action Review (AAR) and real-time mission editing. The visual representation is exceptionally rich, and includes motion-capture animations and the modelling of ballistics, projectile impact (e.g. ricochet), detailed terrain surfaces, weather and more.

#### 3.1 Application Scripting Interface (ASI)

VBS2 has a scripting language that can be accessed via the Application Scripting Interface (ASI) to allow control by external AIs. ASI provides a wide range of scripting commands, many suitable for human behaviour modelling. The advantage of this approach is that it is readily available. The disadvantage is that the range of percepts and actions is limited. There is no way to access information or control an entity other than using the library of commands in ASI.

ASI supports the addition of external code as plug-ins to VBS2's engine. The plug-in gets registered to handle game events and drive entities in the game. VBS2 plug-ins are Microsoft Windows DLLs (Dynamic Link Libraries). They are launched automatically when VBS2 starts. Plug-ins can access the simulation engine through ASI, executing scripting commands and receiving the results.

VBS2 plug-ins are called synchronously by the simulation engine in two different ways:

- Regularly – on each rendered frame (every 15 to 30 ms), the VBS2 engine calls the `OnSimulationStep` method of the plug-in.
- Opportunistically – each time a trigger inside the game engine calls the `PluginFunction` script command. The `PluginFunction` returns a result to the game engine, allowing the external AI engine (e.g. CoJACK) to interact directly with the scenario events.

VBS2 offers a single function to execute scripting commands: `ExecuteCommand`. `ExecuteCommand` is not thread-safe and must be executed only when VBS2 is giving control to the plugin via `OnSimulationStep` or `PluginFunction`. Furthermore, when the simulator window is not in focus, the VBS2 engine suspends, the march of time is blocked, and no update is sent to plugins.

Before beginning the integration of CoJACK with VBS2, an analysis of the available percepts and actions was conducted. This can be viewed as the beginnings of an ontology for perception and action. For a given domain, an ontology is a knowledge representation that defines the types of entities, their key interrelationships and salient properties. For example, in the air combat

domain, an ontology might include types of **aircraft** (entities), the concept of a **target** (property of an entity), **adversary** (property of an entity), **2-ship** (interrelationship between 2 aircraft entities), **payload** (property of an aircraft entity), etc. One benefit of using an ontology is that it abstracts out the SE-specific detail, allowing the behavior models to work across a variety of SEs. The lack of a standard ontology for behaviour representation is a significant impediment to the rapid integration of AIs with SEs.

### 3.2 Perception

Unlike humans, synthetic entities are not usually provided with raw visual data, rather, they are given some representation of the products of human visual perception/cognition (e.g. object type, position, speed, orientation). However, the representation of these synthetic visual percepts is quite unlike that which the human cognitive system uses (Ritter, Baxter, Jones, & Young, 2000). For example, our visual system does not compute an [x, y] position for objects in the environment, nor does it compute the speed of an object in km/h. Yet this is what current SEs, including VBS2 provide. The richness of the visual data available to a synthetic entity is analogous to what a blind person would receive from a human guide: “There is a person 12m due west, he has a shouldered weapon aimed at a human who is situated 70m due north”. To acquire further information, the blind person must ask the guide specific questions such as: “Is the targeted human a civilian?” and “Is the human with the shouldered weapon friend or foe?”. Typically, the human guide will only answer questions, rather than being proactive.

Thus, it is important to bear in mind just how limited and plan-view oriented typical SE-provided perceptual information is. Limitations in perceptual information apply to all sensory modalities, not just vision.

VBS2 models terrain as a grid (cell-based terrain) upon which entities are placed. Each entity has a set of properties, including current position, speed and posture. These properties, and more, are the percepts that CoJACK has access to through ASI. They are too numerous to list here, but include entity allegiance, location, orientation, visibility, distance, speed, weapon direction, ammunition load and physical health. Environmental information includes collision detection facilities, terrain height, whether there is surface water on a given point, and the road segments that connect to a given road segment. VBS2 also exposes parameters that relate to self perception, including aerobic/anaerobic fatigue, endurance and morale.

### 3.3 Action

A major goal of the CoJACK/VBS2 integration is to facilitate the incorporation of psychologically valid variation into VBS2. The variation in a given CoJACK agent’s behaviour arises from its internal state and is

expressed by the agent’s VBS2 embodiment. VBS2 offers a wide action repertoire including, moving to a point, stopping, adopting a specific posture, aiming at a target, firing and even sitting down. It is important to note that the agent’s internal variation has no meaning if it cannot be expressed in VBS2. For example, if running away from an attack is not an available behaviour in VBS2, there is no point deciding to turn and flee.

Furthermore, VBS2 (and other SEs) can be a source of unanticipated behaviour variation (personal communication; Colin Sheppard). VBS2 has its own set of primitive behaviours and an architecture that governs how those primitive behaviours can be composed into sequences of actions. The interplay between CoJACK’s variation, that of VBS2 and mismatches between CoJACK-specified actions and how VBS2 actually expresses those actions, can be a further source of (unexpected) variability. This point applies equally well to perception; the CoJACK agent’s decision making can be skewed by unanticipated limitations in VBS2’s perceptual information.

### 3.4 Configuration

VBS2 supports the setting of some of an entity’s internal parameters, including physiological parameters such as endurance and affective parameters including morale. This provides an avenue for CoJACK to moderate an entity’s performance. It is also possible to enable/disable VBS2’s own AI functionality, thereby providing CoJACK with better control over entity action. For example, VBS2’s path planning function (PATHPLAN) can be disabled so that CoJACK has full control of navigation.

## 4. Integration Architecture

There are currently many different SEs available, each providing a unique collection of AI-interfacing capabilities. External control of AI units within SEs has been achieved in the past, with varying degrees of success, for example, (Jones et al., 1999) and Soar controlling entities in Unreal Tournament (Wray et al., 2002). Allowing external control of SE AI units has typically been added as an afterthought, and this can make external control of AIs problematic for behaviour modelling environments such as CoJACK.

CoJACK models can be very rich in terms of their modeling of cognition and affect, however, this is limited by the ability to perceive and act through the SE entities. For example, a model that increments its level of “fear” in response to an adversary’s threatening posture, will not do so if the SE cannot provide any information about that posture.

Unfortunately, as things stand, the range of scriptable perceptions/actions in current SEs, including VBS2, is quite limited. However, given the demands for more realistic AIs in SEs, this situation is likely to improve

in the future. Therefore it was critical not to tie CoJACK's decision making too closely to VBS2's perception/action repertoire. With this in mind, a layered architecture was adopted, one that largely insulates the CoJACK agents from the specifics of VBS2.

At its top level, the architecture is based on Boyd's OODA (Observe, Orient, Decide, Act) Loop (Coram, 2002). Incoming percepts are buffered (Observe), the situation is assessed (Orient), the agent decides what to do next (Decide), and then performs an action (Act).

#### 4.1 Observe

The SE Integration Layer maps incoming VBS2 messages to SE-independent Perception Events. SEs like VBS2 (on state of the art hardware) are able to render at a rate upwards of 70 frames per second, and can feed these updates at the same rate to CoJACK. This high-speed, *discrete* representation of environmental change is not how humans perceive an essentially *continuous* world. Therefore, CoJACK maps these to **aggregate events** that summarise a stream of percepts of a given type. This aggregation is a function of the type of data and can include an average, maximum or minimum value (or any user-defined aggregation function). These ephemeral aggregate events are then converted to memory elements and stored in Sensory Memory. Sensory Memory forms a snapshot of the current situation and is used to inform the next (Orient) phase of the process. Thus, perception is largely independent of the feed rate of the SE.

Incoming percepts are routed appropriately to the agent's moderators by low-level perceptual plans that perform pre-cognitive functions such as rapidly recognising a dangerous situation. For example, a **fear** moderator has been implemented for the Suicide Bomber Scenario described in the next section. This is fed by low-level perceptual plans that immediately identify threatening situations, that is, without more lengthy cognitive assessment. This function is analogous to one that was for many years thought to be served by the amygdala. However, more recent research suggests that the neuroanatomical situation is more complicated than it first appeared (Kalin et al., 2001).

#### 4.2 Orient

The Orient phase largely comprises Situation Assessment. Situation Awareness (SA) is defined by Endsley (1988) as:

*"... the perception of elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future."* (p. 97)

CoJACK includes an implemented interpretation of the Endsley model. Memory elements (termed **beliefs**) that derive directly from Sensory Memory are tagged as Level 1 SA (i.e. **perception**). Beliefs that are generated by cognitive procedures (termed **plans**) are labelled as Level 2 SA (i.e. **comprehension**). Plans that have been classified (by the modeller) as projecting into the future, will produce Level 3 SA (i.e. **projection**). These tags are used for tracing and monitoring and do not impact upon the agent's processing of percepts (Evertsz et al., 2008).

#### 4.3 Decide

In this phase, the agent decides what to do next. This could be to continue with what it is doing, or could involve dropping a goal to pursue a more critical one.

#### 4.4 Act

In this phase, the agent performs the next action specified in its chosen course of action.

### 5. Suicide Bomber Scenario

To provide a focus for the integration, a scenario was selected that would benefit from behavioural realism and variation. A Suicide Bomber (SB) scenario is one where the enemy could employ unusual and variable tactics, forcing the trainee to consider the adversary's behaviour and potential for variation.

#### 5.1 Vignette

In the SB scenario, the SB walks into a crowded market area; a BLUFOR patrol is observing the scene. The SB's goal is to destroy the BLUFOR unit by detonating the bomb he is carrying. In a variant of this scenario, the SB's goal is to maximise the civilian casualties. A diagram of the layout is shown in Figure 2.



Figure 2. Layout of the example's market area.

## 5.2 Entities

There are two BLUFOR entities (US Marines) in a HMMV (High Mobility Military Vehicle), these are:

- a Gunnery Sergeant, and
- a Lance Corporal (driver).

In addition, two BLUFOR entities are dismounted and standing in front of the vehicle:

- a PFC (Private First Class), and
- a Private.

There are 10 civilians milling about in the market area.

There is one SB. No mentor or observer is present in this scenario. Generally, the mentor would be there to remotely detonate the device if the SB has second thoughts. The observer's role would be to record the outcome (for publicity).

## 5.3 General Agent Design

Currently, all of the civilians are controlled by CoJACK, as is the SB. The BLUFOR entities are currently controlled by human players, but the next phase of the project will allow their control by CoJACK, and will include the ROE implementation described by Evertsz et al. (2007). This will allow the investigation of the interplay between ROE and SB tactics.

The agents share the high-level OODA loop design described earlier. When deciding what to do next, if the agent is already executing the chosen task, it continues with what it is doing. In CoJACK, the currently executing task receives a boost in activation level when it is reselected and so is gradually strengthened. When there is a choice between alternative plans to achieve the same goal, CoJACK uses **plan utility** to select the one that is most likely to succeed. The utility of a plan is increased every time it is successful, resulting in a type of reinforcement learning.

Higher levels of fear reduce plan utility and this has interesting effects on behaviour. The observed effect is analogous to a loss of confidence in the effectiveness of a given course of action. The degree of influence on plan utility is configurable for different plans. By annotating plans in this way, those that are perceived to be "risky" will be less likely to be selected when fear is higher. Although it is difficult to validate the weightings used in this approach, the resulting behaviour is intuitive and convincing, and has training value.

## 5.4 Fear Model

A currently used and appropriate way to model fear is to represent it as a reservoir. When fear is triggered in the behaviour model, the amount of fear is used to

initialize the level of the fear reservoir (instantaneously). The level of fear in the reservoir then decays over time with a half-life that is provided as an input to the moderator. This approach appears to be consistent with data on the effect of suppressive fire. This provides a robust implementation, and one that can be expanded and modified as additional data and needs arise. Variability between agents can be provided by varying the uptake and decay of fear constants as well as the base level. Variability with time is provided by the decay of fear. This approach is useful as a first step, as we want the models to be more complex but we also need them to not be too complex and difficult to understand, modify, and debug.

## 5.5 Morale Model

Like fear, the morale moderator is implemented using a reservoir that decays over time. In the SB scenario, only the SB is initialised with a level of morale, and this does not get incremented during the scenario (there are no events that increase the SB's morale).

High morale favours plans that confront the threat.

## 5.6 Fear/Morale Interaction

Fear and morale have opposing/complementary effects on plan selection. Plans that confront a threat have higher utility when morale is high, but lower utility when fear is high.

## 5.7 Civilian Agent Design

The Civilian agents have a number of perception plans, including ones that recognise a SB when it is nearby. Civilian actions include plans for roaming around the market, and fleeing and shouting when frightened.

Low-level perception plans increase the level of fear if the SB is detected. This happens with a variable probability (default is 0.5) when the SB is within 1.5 metres. Furthermore, if the other civilians start to scatter, the civilian's fear level will increase.

## 5.8 Suicide Bomber Agent Design

In addition to its perception plans, the SB agent has plans to move around the market, zero in on a target, flee or detonate its IED (Improvised Explosive Device).

The SB's level of fear increases with the initial detection of the BLUFOR entity, when a BLUFOR entity initially looks at him, and when a BLUFOR entity targets him.

If the goal is to maximise civilian casualties, the plan utility of the **Detonate** plan goes up in proportion to the number of civilians in close proximity. This is achieved using the same method by which fear dampens plan utility, i.e. by having a moderator,

**effectiveness**, that increases in value when there are many civilians nearby. If there are not enough civilians within range, plan utility never rises enough to trigger the plan to detonate.

## 5.9 Scenario Runs

At this stage, our analysis of the scenario runs is qualitative in nature. Given the same set of plans (procedural knowledge) and beliefs (declarative knowledge), the SB behaviour varies significantly, solely as a result of variation in its cognitive parameters that are moderated by its changing level of fear and morale. Similarly, the civilian behaviour is driven by the level of fear that results from the events played out during the scenario (e.g. proximity of SB, or the observation of other civilians fleeing).

The resulting behaviour is quite plausible. Observers spontaneously attribute emotions to the entities. When the SB ignores the BLUFOR threat, this is seen as evidence that he is zealous and consequently fearless. These plausible behaviours are generated without any hard-coded scripting of the entities.

## 6. Discussion

Although our approach is similar to PMFServ with regard to how moderators translate to behaviour, our approach tightly integrates the moderators with a cognitive architecture that is easy to program at a high level.

While this project is ongoing, there are several lessons that we can report.

1. Accessing basic percepts from VBS2 can be cumbersome (e.g. asking the question “How many soldiers are targetting me?”). This can result in long and complicated scripting messages that end up hindering the progress of the VBS2 engine. This places a limit on the number of percepts that can be handled. Getting SEs to efficiently support emotional cognition remains an area for further work.
2. The basic VBS2 AI used to drive the platforms is tightly integrated with the engine and it can be difficult to prevent entities from reacting spontaneously and independently of CoJACK. Using the **disableAI** call is problematic because this disables other services, including perception. Our current workaround is to keep the basic VBS2 AI enabled and to intervene by sending the entity a command to halt.
3. The path finding algorithm is well suited to handling large numbers of units in open space, but not so good for driving a single unit in cluttered environments. Path finding provides a place for intelligence to play out, a source of variability and an area for affect to influence.
4. When sending scripting messages to VBS2, there is a significant delay in the reaction of the unit

(of the order of a second). For instance, if we want a unit to start running as soon as a weapon is waved at it, we experience some unwanted delay in the time it takes to start running away. We suspect that this delay is due to VBS2 internal AI path finding.

5. Using the **doMove** command is the most flexible way to control entity movement (rather than using VBS2 waypoints). However, this requires that the agent be given a set of destination positions at simulation startup. This means that one cannot use the Real Time Editor (RTE) to modify already loaded positions.
6. Ultimately, an API would be a better way to interact with VBS2, but this is not currently available.

These reservations aside, progress thus far is very promising. The realism of the behaviour of the VBS2 entities is significantly enhanced by controlling them with CoJACK through ASI, notwithstanding the fact that an API needs to be developed at some point in the future.

CoJACK's high-level graphical plan language facilitates the process of writing behaviours and is an effective medium for communicating those behaviours to SMEs (Subject Matter Experts) for feedback. We see this as an important requirement in achieving more widespread adoption of tools such as CoJACK in the video games, training, and operations research.

## 7. Acknowledgements

This work was supported by the UK MOD's Analysis, Experimentation and Simulation corporate research programme (Project No: RT/COM/3/006). We thank Colin Sheppard for useful discussions about CoJACK. We would also like to thank Peter Morrison and Mark Dzulko of Bohemia Interactive for their support throughout this project.

## 8. References

- Anderson, J. R. (2007). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press.
- Bach, J. (in press). *Principles of synthetic intelligence: Building blocks for an architecture of motivated cognition.* New York, NY: Oxford University Press.
- Bartl, C., & Dörner, D. (1998). PSI: A theory of the integration of cognition, emotion and motivation, in *Proceedings of the 2nd European Conference on Cognitive Modelling*, 66-73.
- Boff, K. R., & Lincoln, J. E. (1988). *Engineering data compendium*. Wright-Patterson Air Force Base, OH: Harry G. Armstrong Aerospace Medical Research Laboratory.

- Bratman, M. E. (1987). *Intention, plans, and practical reasoning*. Cambridge, MA (USA): Harvard University Press.
- Coram, R. (2002). *Boyd: The fighter pilot who changed the art of war*. New York, NY: Back Bay Books/Little, Brown and Company.
- Endsley, M. R. (1988). Design and evaluation for situational awareness enhancement, in *Proceedings of the Human Factors Society 32nd Annual Meeting*, 97-101.
- Evertsz, R., Ritter, F. E., Busetta, P., & Pedrotti, M. (2008). Realistic Behaviour Variation in a BDI-based Cognitive Architecture, in *Proceedings of SimTecT 2008*. SIAA Ltd.: Melbourne, Australia.
- Evertsz, R., Ritter, F.E., Busetta, P., Pedrotti, M., & Bittner, J.L. (2008). CoJACK – Achieving principled behaviour variation in a moderated cognitive architecture, in *Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation*. 80-89. 08-BRIMS-025. U. of Central Florida: Orlando, FL.
- 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*, 123-34. 07-BRIMS-021. U. of Central Florida: Norfolk, VA.
- Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4), 189-208.
- Georgeff, M. P., & Ingrand, F. F. (1989). Monitoring and control of spacecraft systems using procedural reasoning, in *Proceedings of the Space Operations Automation and Robotics Workshop*.
- Gratch, J., & Marsella, S. (2004). A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, 5(4), 269-306.
- Hudlicka, E., & Fellous, J.-M. (1996). Review of computational models of emotion (Technical Report No. 9612): Psychometrix. Arlington, MA.
- Hudlicka, E., & Pfautz, J. (2002). Architecture and representation requirements for modeling effects of behavior moderators. In *Proceedings of the Eleventh Conference on Computer Generated Forces and Behavioral Representation*, 9-20. 02-CGF-085. Division of Continuing Education, University of Central Florida: Orlando, FL.
- Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine*, 20(1), 27-41.
- Kalin, N. H., Shelton, S. E., Davidson, R. J., & Kelley, A. E. (2001). The primate amygdala mediates acute fear but not the behavioral and physiological components of anxious temperament. *The Journal of Neuroscience*, 21(6), 2067-74.
- Kase, S. E., Ritter, F. E., & Schoelles, M. (2008). From modeler-free individual data fitting to 3-D parametric prediction landscapes: A research expedition. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 1398-1403. Cognitive Science Society: Austin, TX.
- Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- Picard, R. W. (1997). *Affective computing*. Cambridge, MA: MIT Press.
- Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction*, 7(2), 141-173.
- Ritter, F. E., Reifers, A. L., Schoelles, M. J., & Klein, L. C. (2007). Lessons from defining theories of stress for architectures. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 254-262). New York, NY: Oxford University Press.
- 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. 469-498). Amsterdam: Elsevier.
- Silverman, B. G., Johns, M., Shin, H., & Weaver, R. (2000). Performance moderator functions for human behavior modeling in military simulations, *Prepared for Human Behavior Program, Defense Modeling and Simulation Office*.
- Wray, R. E., Laird, J. E., Nuxoll, A., & Jones, R. M. (2002). Intelligent Opponents for Virtual Reality Trainers, in *Proceedings of the 2002 Interservice/Industry Training, Education and Simulation Conference*.

## Author Biographies

**RICK EVERTSZ** is Research Director at AOS Group. He has a background in cognitive science and is currently working on the incorporation of cognitive and affective constraints into BDI agents.

**MATTEO PEDROTTI** is a software architect at AOS Group, working on behaviour representation, SE integration, GUI design, and tracing and monitoring tools for agent architectures.

**PAOLO BUSETTA** is a senior software architect at AOS Group. He has worked on numerous agent-based systems for research as well as military and commercial applications.

**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. He edits the *Oxford Series on Cognitive Models and Architectures* and is an editorial board member of *Human Factors*, *AISBQ*, and the *Journal of Educational Psychology*.