November 2008

An embeddable testbed for insurgent and terrorist agent theories: InsurgiSim

Barry G. Silverman
University of Pennsylvania, basil@seas.upenn.edu

Aline Normoyle
University of Pennsylvania

Praveen Kannan
University of Pennsylvania

Richard Pater
University of Pennsylvania

Deepthi Chandrasekaran
University of Pennsylvania

See next page for additional authors

Postprint version. Published in *Intelligent Decision Technologies*, November 2008.
Publisher URL: http://iospress.metapress.com/content/120767/

An embeddable testbed for insurgent and terrorist agent theories: InsurgiSim

Abstract
Many simulators today contain traditional opponents and lack an asymmetric insurgent style adversary. InsurgiSim prototypes an embeddable testbed containing a threat network of agents that one can easily configure and deploy for training and analysis purposes. The insurgent network was constructed inside a socio-cognitive agent framework (FactionSim-PMFserv) that includes: (a) a synthesis of best-of-breed models of personality, culture, values, emotions, stress, social relations, mobilization, as well as (b) an IDE for authoring and managing reusable archetypes and their task-sets (Section 2). Agents and markups in this library are not scripted, and act to follow their values and fulfill their needs. So it's desirable to profile the agents (eg, faction leaders, cell logisticians, followers, bomb maker, financier, recruiter, etc.) as faithfully to the real world as possible. Doing this will improve the utility of InsurgiSim for studying what may be driving the insurgent agents in a given area of operation as Section 3 explains. InsurgiSim's bridge is an HLA federate and can be embedded to drive all or some of the insurgent agents in a 3rd party simulator. Three such examples are summarized in Section 4. The paper closes with next steps to improve InsurgiSim's capabilities and utility.

Comments
Postprint version. Published in *Intelligent Decision Technologies*, November 2008.
Publisher URL: http://iospress.metapress.com/content/120767/

Author(s)
Barry G. Silverman, Aline Normoyle, Praveen Kannan, Richard Pater, Deepthi Chandrasekaran, and Gnana K. Bharathy

Intelligent Decision Technologies 2 (2008) 1–11
IOS Press

An embeddable testbed for insurgent and terrorist agent theories: InsurgiSim

Barry G. Silverman*, Aline Normoyle, Praveen Kannan, Richard Pater, Deepthi Chandrasekaran and Gnana Bharathy
Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6315, USA

Abstract. Many simulators today contain traditional opponents and lack an asymmetric insurgent style adversary. InsurgiSim prototypes an embeddable testbed containing a threat network of agents that one can easily configure and deploy for training and analysis purposes. The insurgent network was constructed inside a socio-cognitive agent framework (FactionSim-PMFserv) that includes: (a) a synthesis of best-of-breed models of personality, culture, values, emotions, stress, social relations, mobilization, as well as (b) an IDE for authoring and managing reusable archetypes and their task-sets (Section 2). Agents and markups in this library are not scripted, and act to follow their values and fulfill their needs. So it's desirable to profile the agents (eg, faction leaders, cell logisticians, followers, bomb maker, financier, recruiter, etc.) as faithfully to the real world as possible. Doing this will improve the utility of InsurgiSim for studying what may be driving the insurgent agents in a given area of operation as Section 3 explains. InsurgiSim's bridge is an HLA federate and can be embedded to drive all or some of the insurgent agents in a 3rd party simulator. Three such examples are summarized in Section 4. The paper closes with next steps to improve InsurgiSim's capabilities and utility.

1. Introduction

As the nation faces 21st century adversaries and national security challenges, traditional military actions alone such as sorties flown, rounds fired, or tons of relief materials delivered, are proving less and less likely to guarantee the desired outcome unless complementary behavioral solutions are also considered. The alternative is to focus strategically on the desired outcome, study the adversary's likely behavior, and explore alternative ways to affect the desired result. This concept of better understanding and influencing potential adversaries (and friends) is central to the needs of diplomacy and national policy making. It is consistent also with the military's need to shift to a new paradigm where they consider all the "effects" of various actions as well as alternative pathways to "effect" the same outcome. This is coming to be labeled as effects based operations (EBO) or actions: e.g., see McCrabb and Caroli [9], Pendall [15]. When we take actions, make commitments or issue utterances, what are the behaviors of targeted and other leaders likely to be, especially those influenced and constrained by many sub-groups and special interests? As Smith [20] points out, EBO needs to focus on how we can "shape the adversary's thinking and behavior, rather than on simply defeating his forces." Unfortunately, as Caroli et al. [2] point out, most wargames include traditional threats and there are very few tools available to help understand how leaders and other adversaries behave, that capture what is in their hearts and minds, and that can be used to understand what might influence their behavior and effect the outcomes we desire. There is a need to convert wargames so they offer the proper training and/or analysis capabilities.

1.1. Theories of insurgency/terrorism

A number of theories currently exist about the roots of insurgent and terrorist movements, how they grow, and how recruitment might work (and be hindered) – e.g., see Borum [1]'s review of instinct, drive, social learning, and cognitive theories as well as Collier [5]'s

*Corresponding author. E-mail: basil@seas.upenn.edu.

ISSN 1872-4981/08/$17.00

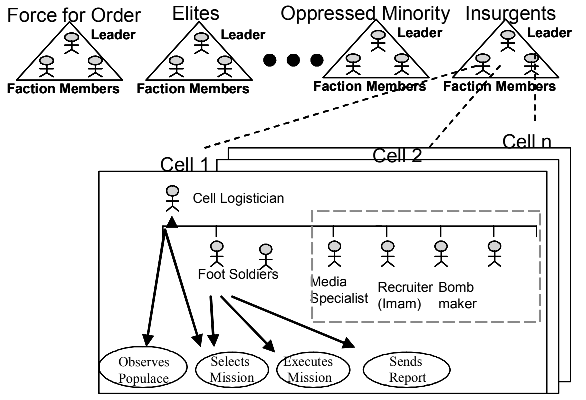

Fig. 1. Factions and agents in an insurgent area.

review of economic greed and poverty based theories. Many of these are ‘paper-based’ theories with survey and event data to support them. But these theories have not been implemented within a detailed socio-cognitive agent-based modeling and simulation framework such as we present here. Implementation is valuable since it serves to test the theories, expose their strengths and weaknesses, and uncover gaps in behavior that the theories do not explain. In general, science is often advanced by a combination of reductive analysis and synthetic usage. The goal of this effort is on the latter. Specifically, we have been constructing a testbed to research and possibly merge different terrorist and insurgency theories. We believe this testbed serves a second purpose of providing a toolset that can be embedded to alter and extend traditional wargames as Caroli et al. mention. We demonstrate both of these goals in this paper.

To begin, one can readily envision an insurgency existing in a world where a number of factions or clans range across the spectrum from those desiring the rule- of-law to those interested in chaos and regime change for any of a variety of reasons (ethno-political grievance, greed, crime, etc.). This is depicted across the top of Fig. 1. Indeed, in the Maoist theory of armed struggle, the preparatory stage of an insurgency is characterized by actions that seek to affect separate factions of the population of the nations or regions they are trying to influence, causing different factions to iterate (dynamically) through several states ranging from animosity and paralyzing fear to sympathy and membership in the insurgent movement [8].

Ideally one would like to realistically simulate such behaviors for the purposes of being able to train against it and analyze what influences it in a given area of operation. To train/analyze how to coopt the agenda of an insurgency and mobilize the populace toward the rule of law one needs a set of simulated factions and insurgent agents readily adapted to any given region. Since members of a given populace will be at varying degrees of support for and participation with each side, this implies that the aim of counter insurgency is not solely to destroy groups at the enemy end of the spectrum, but also to progressively shift individuals and groups closer to the friendly end. In fact, since insurgent cells are often hidden amongst supporting members of the population (bottom of Fig. 1), a focus strictly on reaction to insurgent attacks can be counter-productive. It will leave the agenda in their hands, cause collateral damage to potentially woo-able factions, and make the force for order seem to have no successful agenda of its own. Instead one must encourage the force for order to use a ‘full spectrum’ of approaches to help diagnose the source of grievance, attempt to ameliorate the root causes, and build up whatever services and institutions that are lacking and potentially also causing discontent: eg., see USMC [**?**], Kilcullen [7], Patraeus [11], Nagl [12], Chiarelli [4].

If we were to create a cast of digital archetypes, say for Iraq as an example of a wide array of characters that might appear in many scenarios, we might be talking about archetypes of varying ages and genders covering moderate and extreme Sunnis, Shi’ites, and Kurds, as well as extremist Infiltrating Arabs, and of course unsuspecting civilian victims (contractors, NGO types, press corp, etc.). One could presumably use these to craft scenarios relevant to a range of locales, particularly if some of the archetypes also included various regional leader types, specific types of holy sites, hot houses, and other situational triggers. But how should we craft the archetypes of the various groups and leaders? What would make them legitimate? What would make them easily reconfigured for a new locale? How could an analyst or a scenario or training developer use them one day to recreate a Fallujah that probably can be won only at intolerable costs; the next day to construct a hunt for an Al Sadr type who is hiding in a religious site surrounded by his angry followers; a third day for chasing Taliban or Al Quaeda renegades through the foothills of Afghanistan populated by indifferent or hostile warlords and tribal members; or a fourth day to mock up the impact on the populace and on peacekeeping of the press release of news of Abu Grabin style prisoner mistreatments? The list goes on.

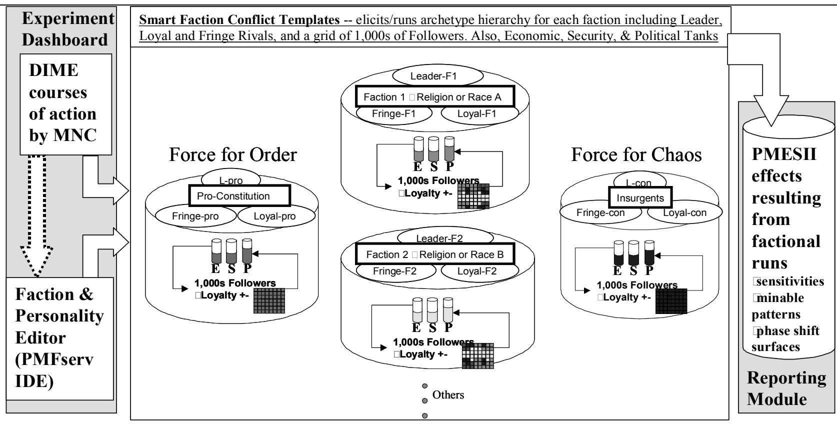

Fig. 2. Architecture for a highly usable FactionSim including DIME experiment dashboard, smart template of reusable archetypes, and PMESII reporting module.

2. Socio-cognitive agents

In this paper we pose the question of how well can the field of modeling and simulation (M&S) recreate insurgent operations such as just described? Specifically, we are particularly interested in the insurgency landscape as a complex social system and hence we want to explore the question of what can agent-based simulation offer? That is, if we use agents to help model the 'parts' and their micro-decision processes, can we observe macro-behaviors emerging that are useful for analysts and trainees to know about? Finally, if we want to model and simulate a social system from the bottom up, then it seems that we need to approach it with agent technology that covers both the social processes that influence people as well as cognitive processes that people use in reasoning and emoting over their fates. That is, we are curious about what can 'socio-cognitive' agents offer to the study of sub-state actors and/or stressed social systems?

Sun [21] provides a useful survey of the respective fields of social agents and cognitive agents and shows that there are very few environments that straddle both topics to provide socio-cognitive architectures. In this paper, we therefore illustrate one such architecture and provide some insights into how it works, what it is useful for, and whether its outputs provide any validity for training and analysis.

2.1. Social framework: FactionSim

FactionSim is a tool where you set up a conflict scenario in which the factional leader and follower agents all run autonomously. You are the sole human player interacting to try and use a set of DIME (Diplomatic, Informational, Military, and/or Economic) actions to influence outcomes and PMESII (Political, Military, Economic, Social, Informational, or Infrastructure) effects. Factions are modeled where each has a leader, two sub-faction leaders (loyal and fringe), a set of starting resources (Economy, E, Security, S, and Politics, P), and a representative set of N follower agents. A leader is assumed to manage his faction's E- and S- tanks so as to appeal to his followers and to each of the other tribes or factions he wants in his alliance. Each of the leaders of those factions, however, will similarly manage their own E and S assets in trying to keep their sub-factions and memberships happy. Followers determine the level of the P-tank by voting their membership level (a topic discussed in PMFserv). A high P-tank means that there are more members to recruit for security missions and/or to train and deploy in economic ventures. So leaders often find it difficult to move to alignments and positions that are very far from the motivations of their memberships.

FactionSim is well documented in the literature and that won't be repeated here: e.g., see Silverman et al. [18,19]. It also has attained a level of maturity. For example, it was used in 2006 to model 7 factions of Iraq with over two dozen named leaders and many archetypical followers (top layer of Fig. 1). That implementation was successfully tested and approved for realism by a group of SMEs that DARPA assembled for two weeks at one of the military COMs. As of this writing and for all of 2008, FactionSim is being used under DARPA sponsorship to assemble profiles and models of all the major factions of 15 countries around the Pacific Rim. Part of that effort is looking into automating the generation of factional models. As part of a separate AFOSR effort we are studying how various theories of insurgency may be implemented within FactionSim.

2.2. Agent cognition: PMFserv

Beginning in 1999, our lab has developed a human behavior model called Performance Moderator Function Server (PMFserv) that includes: (a) a synthesis of about 100 best-of-breed models of personality, culture, values, emotions, stress, social relations, group dynamics, as well as (b) an IDE for authoring and managing reusable archetypes and their task-sets.

The unifying architecture in Fig. 3 shows how different subsystems of PMFserv are connected. For each agent, PMFserv operates what is sometimes known as an observe, orient, decide, and act (OODA) loop [14]. PMFserv runs the agent's perception (observe) and then orients all the entire physiology and personality/value system to determine levels of fatigues and hunger, injuries and related stressors [6], grievances, tension buildup, impact of rumors and speech acts, emotions [13] and various mobilizations and social relationship changes since the last tick of the simulator clock. Once all these modules and their parameters are oriented to the current stimuli/inputs, the upper right module (decision-making/cognition) runs a best response algorithm to try to determine or decide what to do next. The algorithm it runs is determined by its stress and emotional levels. In optimal times, it is in vigilant mode and runs a subjective expected utility algorithm that reinvokes all the other modules to assess what impact each potential next step might have on its internal parameters. The agent calculates the subjective expected utility (SEU) it expects to derive from every action available to it, as constrained by perception and chooses the alternative that maximizes SEU. Thus

Best Response (SEU): (1)

$$\mathrm{Max}\{\Sigma \mathrm{U}(\mathrm{a}_k) * \mathrm{Pr}(\mathrm{a}_k) * \Phi(\mathrm{r}_j) + \psi\}$$

where utilities (U) for next actions, a_k, are the anticipated E|S|P (Economy, Security, Politics) tank gains or losses the actions afford combined with how those affect the agent's value system. Φ (r_j) is a function that captures the strength of positive and negative relationships one has with agent or object j that are effected or spared by a_k, and ψ handles merging and discounting (decay) prior action's affect on the agent's value system. When very bored, the agent tends to lose focus (perception degrades) and it runs a decision algorithm known as unconflicted adherence mode. When highly stressed, it will reach panic mode, its perception basically shuts down and it can only do one of two things: cower in place or drop everything and flee. In order to instantiate or parameterize these modules and models, PMFserv requires that the developer profile individuals in terms of each of the module's parameters (physiology, stress thresholds, value system, social relationships, etc.). Some of these features will be discussed in subsequent sections in explaining how the InsurgiSim agents reason about their missions.

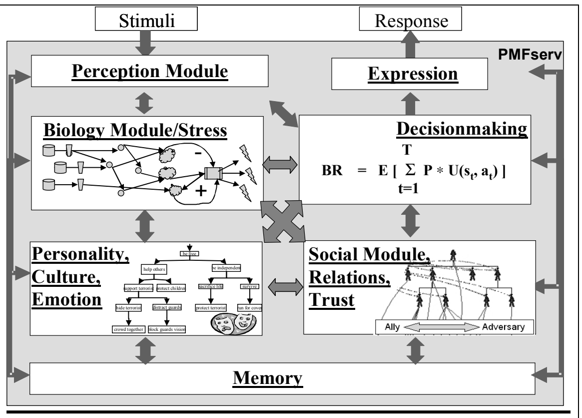

Fig. 3. PMFserv, an example cognitive architecture.

PMFserv is in use by an intelligence agency to model diplomatic decisions of world leaders for which it has passed statistical correspondence tests showing it is significantly in agreement with their decision-making: see Silverman et al. [18]. PMFserv has also reached the level where it can realistically simulate ethno-political conflicts among regional leaders and their followers vying over control of contested resources and assets (published validation studies for application in the Far East, MidEast Africa, and domestically are in Silverman et al. [16,17,19]. FactionSim makes use of PMFserv to run both leader and follower agents and to use their micro-decision making to study group dynamics and how macro-behaviors and new equilibria autonomously emerge. This is useful for studying the impact of actions and effects (DIME-PMESII) both from analytical as well as training and rehearsal perspectives.

3. InsurgiSim overview

InsurgiSIM is a human-trainer tunable, autonomous insurgency force (leader, followers, missions and daily life, etc.) that you can play against. It is intended for plugin to a multi-player game world, first person shooter, or in a wargame such as one of the US military's Semi-Automated Forces (SAF) environments to study tactics, observables, and effects-based operations and issues. Many of these worlds include adversaries but the AI is such that either they use a scripted finite state machine approach useful for force-on-force symmetrical encounters or else a human red team is expected to play out the asymmetric force's roles. This is time consuming and expensive. A prime objective of the InsurgiSim project is to provide an autonomous, culturally realistic, sentient insurgent force to drive the reasoning of the agent avatars in such a game world.

Looking back at earlier Fig. 1 and as discussed in Section 2, FactionSim runs PMFserv agents as the insurgent leaders that populate the top layer of that diagram – the faction layer. In 2006, we added a second, more tactical layer – the cell layer. To do this we built several further archetypes of PMFserv agents that can be configured as an insurgent cell as shown at the base of Fig. 1. That is, we developed reusable archetypes of a threat network of agents (e.g., logistician, diverse followers, and soon- to-be-added bomb maker, financier, recruiter, etc.), world markups (e.g., safe houses, VBIEDs, mosques, FOBs, etc.), and mission scenarios in a graphical environment. These sentient agents have a daily routine (live, eat, pray, meet, etc.) and live amongst the rest of the populace. They have culturally appropriate values, emotions, stressors, and the like. We discuss them further in the next several subsections. First we examine the strategies of the faction leader who deploys them as a loose hierarchy or network of adversaries to the force-for-order. It should be noted that InsurgiSim's strategic and cell layers were each built separately as stand-alone elements in 2006 – the strategic layer for DARPA and the cell layer for embedding inside the JointSAF or JSAF environment to assist a red cell of the Urban Resolve effort. Since that time we have merged the two components into a unified architecture and in 2008 we are embedding it in a virtual village for the USMC. Rather than worry about historical details, in the remainder of this article we will discuss all applications as if they are the unified one.

3.1. Strategic layer

One selects the faction that is ideologically leading the cells and profiles and instantiates its leader. For example, a named Jihadist leader or imam, a charismatic guerilla, or a separatist. The faction leader carries out the overall campaign against the other factions, and in the present architecture is the primary source for recruiting new insurgents and other support to his cause. As described in Section 2.1, the faction leader uses its decision making to decide operations such as non-violent protests, psyops campaigns, economic ventures, defensive efforts, and so on. It does not closely control the cells, but it does signal when it thinks another faction should be attacked. If the cell carries out a successful attack, the factional leader also has the means to publicize this and generally will.

Central to a given leader's value system reasoning is its perception of who threatens it and/or whom it's vulnerable to. Likewise a given leader may be equally interested to estimate who can it influence to best increase its resource assets and thereby its power in the world. Obviously, individual leader value system weights will govern how aggressively a given leader pursues each of these vulnerability vs. power concerns, however, we assume that all leader agents need to be able to compute how vulnerable and/or powerful they are at each turn of a game. Since the game rules define precisely which resources can be used to take hostile actions against which other resources, one can derive a measure of a player's *vulnerability* directly from the state of the game world and the rule set. Intuitively, by factoring vulnerability into the world utility calculation, an agent can avoid world configurations in which another is poised to conduct a devastating attack. Adding border defenses, stocking up on supplies, and pulling money out of the economy can all be viewed as behaviors motivated primarily by vulnerability management.

The vulnerability formula (β) works by generating the percentage of a given player's tokens that can be expected to be lost to a given player in the coming round of attack actions (a_i). For each hostile action ($a_i \epsilon A$) that can be initiated by another player (g), the number of tokens available to attack and defend is tallied. From this the probability of victory is determined, and then multiplied by the percentage of tokens vulnerable to this attack versus the total number owned by the vulnerable player in each resource category. This is the expected percentage of tokens to be lost if this attack occurs in the next round. The maximum over all attacks, then, gives this player ℓ's vulnerability score β to player y.

$$\beta\text{xy} = \text{Max a} \in \text{A} \left\langle \Pr(\text{a}) * \frac{\sigma(\text{x,a})}{C(\text{x})} \right\rangle \quad (2)$$

Agents who purely manage vulnerability, while interesting in their behavior, are not entirely realistic. Human players tend to balance vulnerability against its inverse, *power*. Where vulnerability measures the expected number of tokens a player can lose to other players in the coming round, power measures the expected number of tokens a player can take from others. The calculation of the power heuristic is exactly the opposite as for vulnerability. Player A's vulnerability

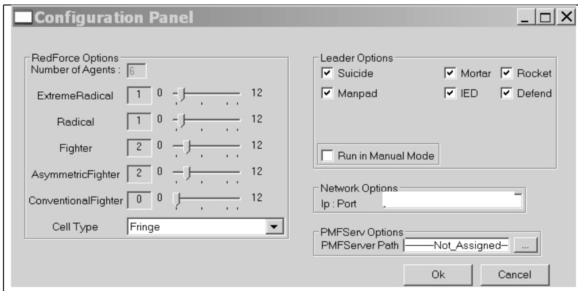

Fig. 4. Insurgent Cell Starting Configuration Panel.

to Player B is the same as Player B's power over Player A.

Taking the leader's perceived difference between power and vulnerability provides a surrogate for the leader's overall sense of utility of the current state of the world, G, when divorced from his value system and other factors:

$$\mathrm{U}l(\mathrm{Gx}) = \alpha\mathrm{x} - \beta\mathrm{x} \tag{3}$$

In standalone FactionSim games there is no spatial graphics or 3_D representation. The insurgent faction's security forces are simply poker chips that the leader decides to wager on a militaristic venture against another faction. In those games the battle outcomes are decided in a contest held by the game engine (using random coin flips for each poker chip played against another faction's defenses). By building the cell layer of Fig. 1, we are endowing a third party game world an ability to conduct and display the actual operations.

3.2. *Tactical decision layer*

To begin, a training developer decides how many cells to place under the factional leader and populate the game world with. For each cell, a GUI editor (Fig. 4) allows one to rapidly setup alternative configurations and types of insurgency forces and mission parameters, depending on what is desired for combat studies and training efforts. For a given cell, the upper right of Fig. 3 shows the types of missions one can currently allow the cell to undertake (chosen based on training needs). And one can check off whether PMFserv should play the cell logistician or if a human should play it instead.

The PMFserv cell logistician does not coordinate with the faction leader, but is a follower and will tend to respond to intermittent, high level guidance such as whether to attack another faction or not. When it responds to such signals, the PMFserv logistician is not scripted, but uses its perception and value system to decide everything such as what specific missions to undertake, when to do them, specific assets of the opponent to target, who to assign them to, and when to lay low. It constantly reassesses the expected utility of each course of action relative to its value system and by also making use of the same type of power and vulnerability computations as mentioned earlier (but for a narrower span of control). Because it uses its own individualized cognition, it is possible that a given cell logistician may be more or less aggressive than the factional leader had hoped. Its also possible for it to be stressed and to commit errors and react emotively.

3.3. *Mission layer*

The left hand side of Fig. 5 allows the training developer to alter the quantity and mix of types of foot-soldiers to include in the cell at the start of the simulation. These invoke different archetypes within PMFserv. At one extreme are those most loyal to the cause and willing to undertake any mission even a suicide bombing. At the other extreme are more conventional fighters, possibly even those who are coerced into missions. These latter tend to be new recruits and are the most likely to decide not to do a given mission assigned by the cell logistician. For example, on the right side of Fig. 5, the extreme Radical Agent can be seen to value the safehouse and emplaced IEDs more highly than the average Fighter Agent does (left side). Likewise the Fighter agent seems to see many daily life routines as having positive utility – going to place of worship, recreation, etc. These are utilities at the start of a random run.

Utility calculations come from an agent summing its positive and negative emotional arousals relative to how the world state is causing success and/or failure of the values it feels are important. Figure 6 illustrates this "cognitive appraisal" of the world with the right hand side depicting a portion of the Asymmetric Fighter agent's value system and its importance weights. The left side indicates an agent that is highly distressed, angered, and disliking of the world state. Figure 5b shows an agent right after conducting a mission (IED emplacement and detonation), and the positive emotional construals that precipitates.

This discussion points out that the foot soldiers of the cell live at the safehouse which happens to be at the upper right of Fig. 6b. They are the squares in green. Yellow circles with stars inside are potential targets that the logistician might communicate with a mission

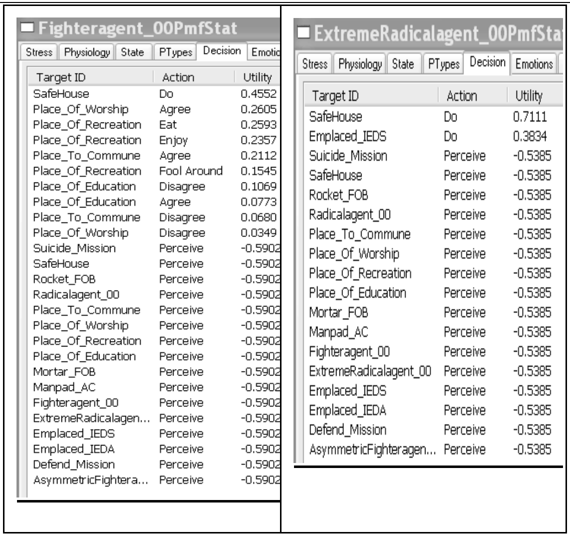

Fig. 5. How two different foot soldiers differ in their assessment of the utility of various missions.

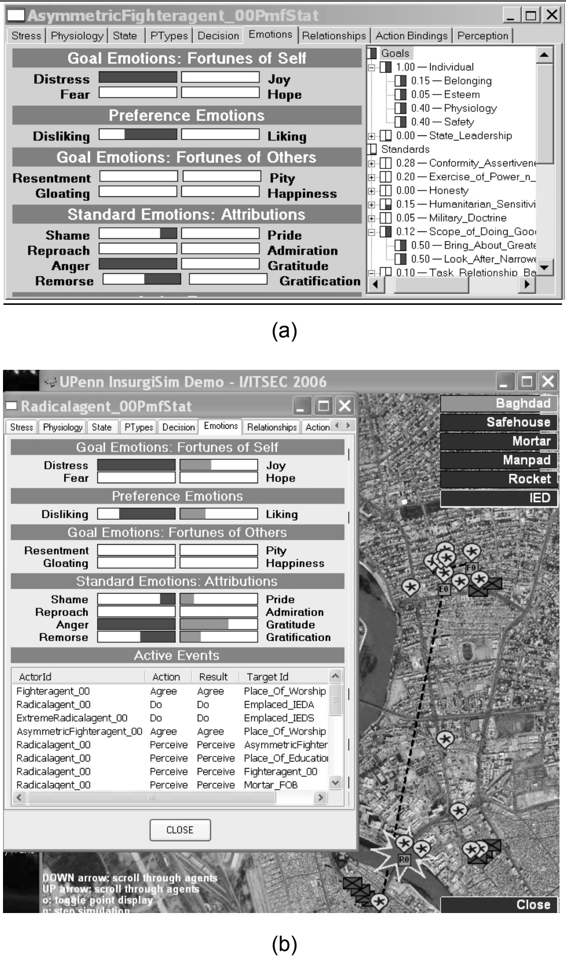

Fig. 6. (a) Shred of a bayesian-weighted value hierarchy and the emotions it precipitates. (b) How conducting a mission alters an agent's emotional state.

assignment, while rectangles (blue) are the forces for order in this environment. The explosion symbol at the base of Fig. 6b designates a successful IED mission that destroyed a bridge, some adversaries, and some civilians. The popup window shows the PMFserv view into the head of the agent that carried out the IED mission

The bottom of the Fig. 6b popup also lists the missions currently being carried out by all the agents of the cell. One can see that many of the cell members are in fact doing daily life activities. Based on their utility calculations of the world state, agents may opt to do missions from the cell leader or not. Further, even if an agent accepts a mission, these other missions (daily life routines) are still options that it may return to. When carrying out a mission, an agent may pause it to do something else, resume later, or abort altogether. Thus it might pause when carrying an IED from the safe-house to a car if the blue forces are noticed outside, and then resume after they move on. The decision to start a mission, pause, resume, or abort is up to the individual agent and is based on its re-assessment of its strength of group affinity (membership level), congruence with the value system and action choices of the group leader, and concern for its own safety and other values. As membership and congruence grow or fall, the agent will alter its loyalty and willingness to undertake assigned missions. Or it may abort a given mission altogether due to lack of commitment to the group and concern for its own safety, and instead choose to go eat some food, meet a friend, etc.

4. Inter-operation with external simulators

InsurgiSim is intended to drive the behavior of agents in 3rd party simulators. PMFserv has previously been embedded behind a number of applications, game-worlds, and simulators – e.g., Unreal, Sony OpenSteer, Big World, a diplomacy game engine (Athena's Prism), OneSAF, and JSAF. Such implementations generally work with a Bridge or Gateway program that links PMFserv (or InsurgiSim) and the 3rd party simulator. Here we illustrate the latter of these interoperation efforts by explaining the PMFserv side, the bridge, and finally the JSAF side.

4.1. PMFserv sim side services layer

PMFServ has been designed to be easily leveraged by external simulators to provide human PMFs. Because the domain and resolution of simulations using PMFserv services can vary greatly PMFserv was designed to be domain agnostic. PMFserv uses a simple interchange mechanism to allow its models to interact and be driven by external simulations. This interchange mechanism is simple enough that it can be developed on a per simulation basis.

The reader may recall that each PMFserv simulation contains agents and a series of objects that the agents can perceive and make decisions about. This collection of objects represents the "World" that the agents inhabit. The objects contain rules that drive how an agent perceives them in terms of his value system. PMFserv's integrated development environment (IDE) provides the tools needed for composing these objects. A running version of PMFserv remains until an external simulation makes a change to a PMFserv object. Once an object has had its state changed PMFServer reacts to the change by updating all of the agents who can observe the change. This results in an Agent's physiology, perceptions, stress, emotions and decisions being updated. These new updated agents are immediately available to affect external agents as PMFServ's "Performance Moderators".

The state information that is shared between PMFserv and an external simulation is determined during the knowledge engineering phase of development. For example if you are modeling a car in PMFserv you might include information about its current speed and direction. You might event want to include something as esoteric as what the car's bumper sticker says. This depends on what is determined to be relevant to the perceptual types and affordances that capture the behaviors and culture that pertains to the simulation's domain. Once you have modeled an object in PMFServ, an external simulation can access this object at runtime. This means a racing simulation could share state information between one of its 'cars' and the one modeled in PMFServ.

4.2. *Interchange layer and bridge services*

External simulators and InsurgiSim will generally be asynchronous and operate on differing time-scales. A key service of the bridge is to facilitate this asynchrony so that each side is only notified when changes occur that are significant to that side (publish and subscribe pattern). Via a simple exchange of property and value information InsurgiSim models can thus provide PMFs to a wide range of simulation architectures without major changes to its code base. Another service of the bridge layer is to support translation lookup tables. These are authored at setup time. They contain translations between InsurgiSim's parameters (e.g., name, value, units, aggregation, etc.) and those of the third party simulator.

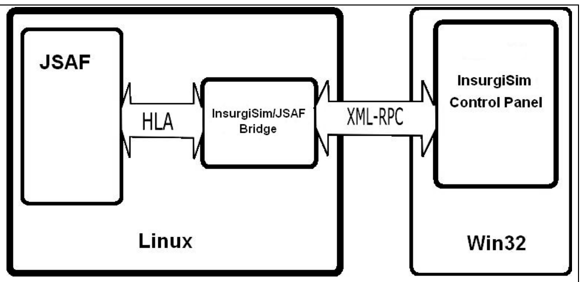

Fig. 7. The Bridge approach to embed InsurgiSim.

To support such services, we have defined an interface layer between InsurgiSim and any 3rd party simulators, using a global blackboard and a simple synchronization loop. Each agent has methods which both check desirable InsurgiSim parameters and execute motions and graphics on the selected platform and application. Thus we need to define the parameter sets and the protocols to set, query, and update state of mutual interest to both InsurgiSim and the human display models. It needs to be bi-directional since low-level activities (navigation, perceptions, threats, injuries, etc.) can clearly affect PMFs.

The Bridge communicates with JSAF using HLA and with InsurgiSim using XML-RPC (see Fig. 6). InsurgiSim exposes the properties of its objects via an API. This API allows a simulation to both set state and retrieve state from any PMFserv agent or object. Because of the loop's simplicity, InsurgiSim can be used with a multitude of disparate simulators. Using this bridge approach, PMFserv has been synchronized with external simulations via Microsoft's COM (Unreal Tournament), the HLA/RTI protocol, and TCP/IP (LeaderSim project). Another application of PMFserv similar to InsurgiSim is called CrowdSim and it has used this same bridge architecture to interface PMFserv agents to drive behaviors of crowd members in Python's C-API (opensteer crowd model), DARPA's Real World, JSAF, and OneSAF. The interchange choice is highly dependent on the external simulation and its own internal simulation loop. The nature/number of state information exchanges that are necessary to support the accompanying PMFServ Model and the native environment of the external simulation also play a large roll in the development of the a synchronization loop.

A generic synchronization loop is:

1. External simulation calls InsurgiSim and informs it of a state change. Alternatively InsurgiSim could poll an external Simulation for pertinent state information.
2. InsurgiSim processes this new information and updates its agents.
3. The external simulation requests any PMF Moderator variables that it uses.
4. The external simulation uses the new information to moderate itself.
5. New external simulation state forces the loop to repeat.

This bridge is portable and can be placed to reduce latency. As Fig. 7 shows, InsurgiSim currently runs on any MS Windows platform. It is written in a portable Python, but certain of its security admin functions are currently tied to MS operating systems. JSAF, in turn, is optimized to run under Linux. Since JSAF updates its state in real-time and InsurgiSim is needed less frequently (but runs faster than real time when invoked), the messaging flow was optimized by placing the bridge on the Linux platform with JSAF.

4.3. JSAF layer services

To coordinate JSAF with InsurgiSim (and its two software servers for this project RedLarry and Leader), the Bridge needs to synchronize JSAF entities with InsurgiSim agents, forward user configuration (safe houses, ICs, targets, etc.) from JSAF to InsurgiSim, and finally translate each agent's high-level decisions into concrete JSAF entity tasks. Therefore, to communicate with JSAF, the Bridge is also an HLA federate. As Fig. 8 reveals, it subscribes to InsurgiSim control interactions and publishes JSAF task interactions. Specifically, when a human operator inputs information into the PMFServ Orders menu (it is called PMFserv in the JSAF menu structure), JSAF sends a corresponding interaction to the Bridge. Through these interactions, users can constrain and alter missions along with their parameters (e.g. objects, vehicles, or weapons) and as earlier Fig. 1 portrays. The Bridge also detects and forwards dynamic environment information that may affect an InsurgiSim agent's decision (such as the number of enemy or blue units in the immediate vicinity of a JSAF agent/entity).

As the Leader and Red Larry (which hosts the foot soldiers) agents make decisions, the Bridge controls their corresponding JSAF entities via task interactions, such as "move to building X" and "mount vehicle Y". That is, our Leader Server runs FactionSim and the Logistician Agent who picks out the missions for the Red Larry foot soldiers. They in turn decide how to carry out these missions, the phases or task steps needed, when daily life routines are warranted, when to abort, and so on. They also monitor how well their corresponding JSAF entities are doing in executing these tasks. Finally, when an InsurgiSim agent has made a choice, the Bridge translates it to JSAF's virtual world. Therefore, to synchronize the decisions from InsurgiSim with the tasks in JSAF, the Bridge must continuously monitor the tasking status of both InsurgiSim agents and JSAF entities. Users can learn what orders InsurgiSim has issued by looking at the messages passed between the Leader, the Red Larry agents, and the Bridge.

5. Conclusions and next steps

This article has described a reusable insurgent force simulator that allows one to test different theories of insurgency and plug them into a 3rd party simulator to drive the insurgent avatars and to play against them for training or analysis purposes. The theories contained in the default version of InsurgiSim straddle individual psycho-physiological ones (stress, emotion, sacred values, etc.) as well as social ones (belonging, mobilization and grievance, group membership, motivational congruence). One can edit these starting theories with the internal editors of FactionSim and PMFserv. As such, InsurgiSim serves as a successful proof-of-existence test for socio-cognitive agent architecture.

The first version of InsurgiSim's cell layer was completed and fielded at the end of 2006 as part of the Urban Resolve testbed: see Ceranowicz and Torpey [3]. Its role there was to support the red team so they did not have to operate every foot soldier and decide every mission. The first version of InsurgiSim's strategic layer was also fielded in 2006 as part of an initiative to assemble a country model of Iraq. Other country models are now under construction.

In addition, we have plugged InsurgiSim into two other applications. The screen shots of Bagdad shown in this article are not JSAF, but are those of a test harness we built in MS Windows. The only entities in this world are the InsurgiSim agents, and they are embodied only within a 2-D or plan level viewer. This does allow us to view their decision making and movement around the terrain.

All layers of InsurgiSim are being unified and put to use as part of a virtual village we are currently assem-

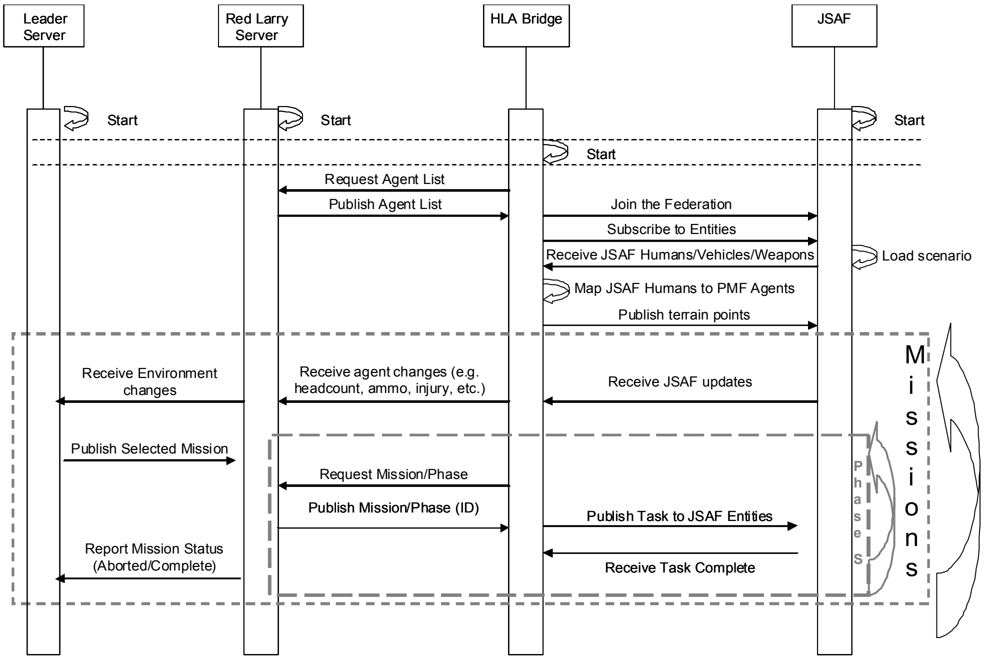

Fig. 8. Message flows for the JSAF federation.

bling for the US Marine Corp. Called NonKin Village, this is a gameworld that brings life to agents of all factions of FactionSim in sort of a SimCity style of play although it supports street level interaction and dialog with agents to learn their issues, needs, grievances, and alignments and to try to assist them in countering the agenda of the InsurgiSim faction. If you mis-manage the situation, various factions and members might be drawn to the insurgent faction's side.

5.1. Next steps

To support InsurgiSim, we successfully implemented first versions of several features that are now open for further research. In fact, there are a host of further research directions that a project like InsurgiSim opens up. We mention but a few of these here.

(1) *Terrain reasoning* – PMFServ and InsurgiSim currently reason about the terrain from a mission-level perspective (e.g., consider relevant locations, select targets, and think up what supplies to bring where). This nicely complements JSAF entities' ability to navigate to points in the environment, move around, avoid collisions, aim and fire weapons, etc. FactionSim leaders, on the other hand reason strategically about each faction's relative strength, assets, and point of vulnerability. In the present JSAF version, strategic reasoning is wasted since the terrain data is devoid of up-to-date information on which factions control what assets and what condition they are in. The same applies to the populace identities and factional memberships, and to the economic status of each faction. Projects like the Army's Human Terrain program hold the potential to alter how country databases are marked up with things like property ownership, socio-economic status and value of various assets, and factional leanings of the populace. We think FactionSim offers guidance on what those markups should provide and offers a rich base for exploring how agent based reasoning about terrain would use it.

(2) *Campaign reasoning* – the InsurgiSim agents at all levels (strategic, tactical, and mission) do not come up with new plans. Instead, they are able to use emotions and value systems to reason about and select between alternative plans, missions, and life routines and for which group to do them. Also, PMFserv has built in functions for agents to form a model of other (MOO) agents. This is based on a technique of mirroring one's value system through an alignment matrix to determine the other's values. Mirroring is a human bias and often leads to projecting inappropri-

ate assumptions about the motivations of others. Hence it is of interest to see where this type of MOO and other more rational MOO approaches lead the agents. These and other forms of campaign reasoning are worth exploring further.

(3) *Theory of insurgency/terrorism* – As stated at the outset, the default version of InsurgiSim comes with built in theories of insurgency and terrorism based on social-psychological models of sacred values and ethno-political factional grievance as well as more materialistic greed concerns such as control of resources. A big part of the reason for constructing InsurgiSim the way we did atop a socio-cognitive agent framework is so that one has a theory testbed with which to conduct studies of how people's behavior shifts as different theories are attempted and as varying policies for mitigation are pursued. One alternative theory we have begun investigating is the dual sector model of a developing economy. Many other such theories merit study as well.

(4) *Cultural bridge* – At present the bridge provides translation services limited by the few character animations that are often found inside of the 3-D gameworlds. PMFserv pumps out parameters about emotive, physiologic, and motivational state that few 3-D viewers in the military domain are able to animate. To make matters worse, the ideal translation table should suggest changes in gestural and communicative acts that are relevant for the culture being gamed. That is, if PMFserv outputs a warm greeting request, this translation could invoke the proper gestures and terms to utter for the relevant culture. This is feasible, but it needs someone to research and assemble all the parts.

Acknowledgement

The research on the InsurgiSim cell layer was supported by JFcom (J9) via the UO FACT program. The FactionSim related work was supported by AFOSR and DARPA. The unified version is being supported by AFOSR, USMC, and ONR. None of these agencies are responsible for the views expressed here.

References

[1] R. Borum, *Psychology of Terrorism*, Tampa: Univ S Florida Press, 2004.
[2] J.A. Caroli et al., Tools for Effects Based Course of Action Development and Assessment, *Proc CCRT Symposium*, San Diego: DODCCRP.Org, 2004.
[3] A. Ceranowicz and M. Torpey, Adapting to Urban Warfare, *Proc IITSEC Conf*, 2004.
[4] P.W. Chiarelli and P.R. Michaelis, *The Requirement for Full-Spectrum Operations*, Military Review, July-August, 2005.
[5] P. Collier, *The Bottom Billion: Why the Poorest Countries Are Failing and What Can Be Done About It*, Oxford: Oxford U Press, 2007.
[6] I.L. Janis and L. Mann, *Decision Making: A Psychological Analysis of Conflict, Choice, and Commitment* New York: The Free Press, 1977.
[7] D. Kilcullen, *Twenty-Eight Articles: Fundamentals of Company-Level Counterinsurgency*, ISPERE – Joint Information Operations Center, 2004.
[8] S.B. Griffith, (Translation), *Mao Tse-Tung On Guerrilla Warfare*, New York: Praeger, 1961.
[9] M.J. McCrabb and J.A. Caroli, Behavioral Modeling and Wargaming for Effects-Based Operations, *Proc Military Operations Research Society Annual Meeting*, Washington DC: MORS, 2002.
[10] J. Nagl, *Learning to Eat Soup with a Knife*, Chicago: U. Chicago Press, 2002.
[11] D.H. Patraeus, *Observations from Soldiering in Iraq*, Military Review, January-February, 2006.
[12] J. Nagl, H. Patraeus and J. Amos, *U.S. Army/Marine Corps Counterinsurgency Field Manual*, 2006, Avail. Amazon.com.
[13] A. Ortony, G.L. Clore and A. Collins, *The Cognitive Structure of Emotions*, Cambridge: Cambridge University Press, 1988.
[14] F. Osinga, *Science, Strategy and War: The Strategic Theory of John Boyd*, UK: Routledge, 2006.
[15] D.W. Pendall, Effects Based Operation and the Exercise of National Security, *Mil Review* (Jan 2004), 20–31.
[16] B.G. Silverman, M. Johns, J. Cornwell and K. O'Brien, Human Behavior Models for Agents in Simulators and Games: Part I – Enabling Science with PMFserv, *PRESENCE* **15**(2) (April 2006), 139–162.
[17] B.G. Silverman, G. Bharathy and K. O'Brien, Human Behavior Models for Agents in Simulators and Games: Part II – Gamebot Engineering with PMFserv, *Presence* **15**(2) (April 2006), 163–185.
[18] B.G. Silverman, G.K. Bharathy, B. Nye, Eidelson, Modeling Factions for Effects Based Operations: Part I – Leader and Follower Behaviors, *J. Computational & Mathematical Organization Theory* (Dec 2007).
[19] B.G. Silverman, G.K. Bharathy, B. Nye, T. Smith, Modeling Factions for Effects Based Operations: Part II – Behavioral Game Theory, *Journal Computational & Mathematical Organization Theory* (2007), (Accepted for Publication).
[20] E.A. Smith, *EBO: Applying Network Centric Warfare in Peace, Crisis, and War*, Washington DC: CCRP, 2003.
[21] R. Sun, *Cognition and Multi-Agent Action*, Cambridge: Cambridge Univ Press, 2004.