UrbanSim: A Game-based Simulation for Counterinsurgency and Stability-focused Operations

Ryan MCALINDEN[a], Andrew S. GORDON[a],
H. Chad LANE[a,1] and David PYNADATH[a]
[a] *Institute for Creative Technologies, University of Southern California, USA*

Abstract. The UrbanSim Learning Package is a simulation-based training application designed for the U.S. Army to develop commanders' skills for conducting counterinsurgency operations. UrbanSim incorporates multiple artificial intelligence (AI) technologies in order to provide an effective training experience, three of which are described in this paper. First, UrbanSim simulates the mental attitudes and actions of groups and individuals in an urban environment using the PsychSim reasoning engine. Second, UrbanSim interjects narrative elements into the training experience using a case-based story engine, driven by non-fiction stories told by experienced commanders. Third, UrbanSim provides intelligent tutoring using a simulation-based method for eliciting and evaluating learner decisions. UrbanSim represents a confluence of AI techniques that seek to bridge the gap between basic research and deployed AI systems.

Keywords. simulation-based training, serious games, social simulation, narrative-based learning environments, intelligent tutoring

Introduction

The last decade has seen enormous changes in the way that simulation technology is being used in military training applications. Fuelled by advances in the computer game industry, contemporary military training simulations are the product of equal parts pedagogical design and computer game technologies. The adoption of computer game technologies, in particular, has opened up new opportunities for the innovative application of artificial intelligence (AI) research, but often not in the most obvious ways. While there continues to be enormous interest and progress in the creation of autonomous virtual characters for virtual training environments [7], many of the recent success stories in the application of AI technologies for military training involve "under-the-hood" software components that, on their own, represent only a small portion of current AI research. Still, progress over the last decade has led us further than the simple path-planning AI of the early game-based military simulations that were deployed (e.g. [11]). For example, the recently deployed *Tactical Iraqi* training application [6] incorporates sophisticated speech recognition techniques to allow users to converse with virtual characters in a foreign environment, while the recently

[1] Corresponding Author: H. Chad Lane, USC Institute for Creative Technologies, 13274 Fiji Way, Marina del Rey, CA 90292 USA, email: lane@ict.usc.edu

deployed *BiLat* training application [9] integrates a generative character animation model to allow trainees to better learn about intercultural communication in the context of bilateral negotiations.

A common factor in each of these previous cases was that the AI technology was well suited to support the specific pedagogical goals of the training application, one of several practical considerations to be made when introducing technology into military training environments [4]. Although these considerations impose certain constraints on the appropriateness of certain AI technologies, the design space of simulation technologies that successfully incorporate AI remains large and largely unexplored. In order to make progress on the innovative application of AI for training purposes, novel combinations of learning objectives, simulation environments, and AI technologies should be explored and evaluated.

In this paper, we explore a research effort that combines learning and AI to support the U.S. Army in developing the skills of military commanders in the conduct of counterinsurgency operations. We describe the *UrbanSim Learning Package*, a practice environment for counterinsurgency operations based roughly on the design of commercial games that allow players to manage cities (e.g. Electronic Arts' *SimCity*). UrbanSim incorporates several "under-the-hood" AI technologies, three of which are described below after a brief overview of this training application.

1. UrbanSim Overview

Today's military leaders face extremely stressful and demanding situations that are, in many cases, not covered by standard tactics and doctrine. These operations, which combine both lethal and non-lethal aspects of warfare, have been referred to as "armed social work," in which military forces attempt to "redress basic social and political problems while being shot at" [8]. The overarching challenge is to develop leaders who possess adptive expertise and function effectively in complex environments. The goal is to prepare leaders for novel situations unlike any they may have experienced in the past [1,16]. To address this challenge, we developed an instructional software suite for military commanders and their staffs to practice directing and coordinating operations with a "stability-focused" component. The *UrbanSim Learning Package* (or *UrbanSim* for short) focuses predominantly, but not exclusively, on military operations in support of the local citizenry and government that take place after primary offensive and defensive efforts have concluded. UrbanSim has adopted a formal instructional design approach to guide the development of the underlying simulation-based experience. This design approach has proven successful in the development of similar pedagogical training aids, such as the BiLAT system [9], a precursor to UrbanSim that focuses on bilateral negotiations to improve cultural awareness. Building on the lessons learned from BiLAT, and applying the principles from Guided Experiential Learning (GEL) [2], UrbanSim was designed, developed, and deployed with a strong pedagogical focus. However, the resulting learning objectives called for a complex, dynamic, yet highly realistic simulated environment. This brought about the need to employ agent-based research technologies and transition them to software that would eventually be used in the classroom.

Consisting of two separate but tightly interwoven applications, UrbanSim adheres to the GEL model by providing learners with a complete cognitive foundation required to conduct these types of complex, dynamic operations. The first application, the

UrbanSim Primer, provides the requisite conceptual and task knowledge required for the learner to lead a full-scale stability operation, from analyzing background information to coordinating the actions that are carried out in support of reaching a desired end state. Taking the form of an interactive tutorial, the UrbanSim Primer is broken into eight lessons, each of which contain a narrative, interview segments from former Commanders, and assorted practice exercises as a means of demonstrating specific tasks to the learner. Taking approximately one to two hours to complete, the self-paced Primer prepares the learner for the second application, the more complex UrbanSim Practice Environment.

The UrbanSim Practice Environment is a game-based social simulation that allows a learner to plan, prepare, execute, and assess a full stability operation. Similar to the mechanisms employed for a turn-based strategy game (such as *Civilization* or *Age of Empires*), the learner directs subordinate units to take action with and against agents in a virtual environment (shown in Figure 1), and attempts to successfully complete a mission using the products/strategies learned in the UrbanSim Primer. These actions are taken against key individuals, groups, and structures in an area of operation in the attempt to reach a desired end state, which is often associated with support of the local populace for the local government. Each turn-cycle in the game represents one day in simulation time, though actions can take multiple turns (i.e., days), and can be interrupted if conditions in the world do not allow the action to complete (e.g., money runs out to construct a school). Upon completion of a scenario, the learner is brought to a debrief phase where a summary of the mission is presented for the learner to evaluate their progress.

The UrbanSim Practice Environment is unique in that it incorporates several AI technologies in order to support the training objectives of the application. The AI technologies that were selected each address specific challenges in simulating the complexities of urban environments and in guiding users in learning the skills of counterinsurgency operations. First, we utilized the PsychSim social simulation tool to model the goals, interrelationships, and beliefs of the population of the urban

Figure 1. UrbanSim practice environment

environment, and to select the behaviors of these agents using a decision-theoretic framework. PsychSim was particularly appropriate for this application, as it simplified scenario authoring by automatically generating behaviors from a reusable library of entities that could be composed in different combinations for different scenarios. Second, we developed a case-based Story Engine to interject narrative elements into the simulation environment, where events are drawn from a case library of the real-world experiences of military commanders. The use of a data-driven Story Engine allows UrbanSim to generate events in the user experience that would be difficult to model in rule-based simulations, and provides a framework for quickly modifying UrbanSim content to reflect changes in the contemporary operating environment of the U.S. Army. Third, we developed a new intelligent tutoring system to deliver informative feedback on learner actions, facilitated by a look-ahead procedure provided by the PsychSim system. This technology was particularly appropriate for UrbanSim due to the nature of the skill set of counterinsurgency operations, where the lack of strict doctrine and procedural rules prohibit the development of an expert model of the sort typically employed in contemporary intelligent tutoring systems.

The integration of these AI technologies is facilitated by UrbanSim's system architecture, which follows a data-driven distribution model where these AI components to work together in a synchronous cycle. Each cycle begins when a learner specifies a set of actions to be executed by subordinate units for the given turn. These actions are then sent to the intelligent tutoring system for evaluation, which may initiate a question-answer tutoring dialogue with the learner. Once this dialogue is complete, the learner commits the actions and the simulation cycle is executed. The simulation cycle is comprised of two separate components that alter the current state of the world: the PsychSim social simulation tool and the Story Engine. Each component is run one step (a simulated day) to produce a set of effects based on the actions designated by the learner as well as those taken by the non-player characters in the environment. These effects are then aggregated and presented back to the learner in the form of various display mechanisms, such as spatial views and social network diagrams, and the cycle repeats.

Each of these three AI technologies are described further in the following sections.

2. Population modeling with PsychSim

We simulate the population in UrbanSim (including relevant people, groups, and environment) using the PsychSim social simulation tool [13]. PsychSim can model an entire social scenario, where a diverse set of entities, either groups or individuals, interact and communicate among themselves. Each entity has its own goals, relationships with other entities (e.g., friendship, hostility, authority), private beliefs, and mental models about other entities. PsychSim generates the behavior for these entities and provides explanations of the result in terms of each entity’s goals and beliefs.

2.1. Authoring a Population Model

An author creates the population model for a given scenario by instantiating the desired agents from a library of generic PsychSim models. These generic models (e.g., “Mayor”, “Neighborhood”, “Insurgents”, “Utility”) specify the default state feature values (e.g., a “Mayor” has “political power”), actions (e.g., “Insurgents” may “detonate an IED”), goals (e.g., a “Neighborhood” wants to maximize its “physical security”), action dynamics (e.g., “detonate an IED” decreases the “physical security” in the target “Neighborhood”), and relationships (e.g., a “Mayor” may be “politically aligned” with “Insurgents”). Having selected the instances out of these generic models, the author then specializes them, giving them appropriate names and overriding the default values as necessary (e.g., one neighborhood may have a higher “physical security” value than default, or the specific Mayor may *not* be aligned with any of the insurgent groups in the given scenario). This instantiation process generates a set of operational PsychSim agents corresponding to the relevant entities in the scenario.

2.2. Behavior Generation

During each execution cycle, these agents generate their behavior in turn through a bounded look-ahead procedure that seeks to maximize expected reward by simulating the behavior of the other agents and the dynamics of the world in response to their selected actions. Each agent computes a quantitative value of each possible action by using the authored dynamics to project the effects of each action, weighing the effects against its goals. Thus, the agent is seeking to maximize the expected reward of its behavior (as in a Markov Decision Problem). However, PsychSim's agents exhibit only bounded rationality, constrained by the finite horizon of their look-ahead procedure.

2.3. Behavior Explanation

We also exploit PsychSim's quantitative agent models to provide explanations of the agents' behavior that serve as input to UrbanSim's intelligent tutoring. The agent's behavior generation process constitutes a piecewise linear function that we can invert for a fast sensitivity analysis [13]. We can thus examine each agent's decision and immediately compute the conditions under which it would have chosen a different, preferable action (e.g., the insurgents would not have detonated the IED if the neighborhood had been more secure). PsychSim can also examine the dynamics of the player's actions to identify those moves that may bring about the desired state change (e.g., the neighborhood would have been more secure if there had been a patrol present). The complete set of explanations and suggestions provides a raw input to UrbanSim's Intelligent Tutoring System.

3. Story-driven simulation

There is a rich tradition of pedagogical storytelling in military organizations, where "war stories" are swapped in both formal and casual settings to illustrate points and debate tactics and strategy. This tradition naturally transfers to computer-based learning environments for military trainees. For example, the *Air Campaign Planning Advisor* [5] guided trainees through the process of military air campaign planning through the presentation of approximately one thousand video clips from 12 experts, who told stories of their experiences in the Gulf War, Bosnia, Somalia, and Haiti. More recently, there has been interest in story-based learning environments [3], where the real-world stories of experts are used to design analogous fictional situations that relay the points of these stories. Often realized using computer game technologies, story-based learning environments move the content of expert stories closer to the experience of the trainees, albeit mediated through the designers who craft the fictional situations.

For UrbanSim, we sought to take this characteristic of story-based learning environments one step further in the direction of story-*driven* learning environments, where the events in the fictional training simulation are more directly generated from a corpus of real-world stories. For this purpose, we developed a story-driven simulation engine that works in parallel with the PsychSim social simulation tool described in the previous section, and created a new story repository based on the real-world experiences of commanders of counterinsurgency operations during the Second Gulf War.

3.1. Story Collection

In the fall of 2007, we conducted five hours of story-collection interviews with five U.S. Army Lieutenant Colonels, each having just completed service as a battalion commander during Operation Iraqi Freedom. We collected sixty-four stories related to counterinsurgency operations using story-collection techniques that had been successful in the development of previous story-based learning environments [3].

As a representative example, one commander told us the story of a suicide-bombing attack resulting in numerous casualties, carried out by a woman and child at a location where the U.S. Army was recruiting police officers. In response, the commander launched a campaign to sway public opinion against the insurgents. He solicited the families of children that were killed in similar attacks for cute photographs of these children, then printed and distributed pamphlets with these pictures around the city with the explanation that these were the children that were being killed by insurgent bombers. The campaign effectively turned public opinion against the insurgents, and fewer suicide bombings subsequently occurred.

3.2. The UrbanSim Story Engine

To use these stories to generate events in UrbanSim, these stories were encoded as sequences of events, each with preconditions and effects, using an XML formalism in the language of the UrbanSim world state model. For example, the story above was represented as a sequence of two events (the insurgent bombing and the pamphlet campaign). Event preconditions were encoded in two forms. First, *world state preconditions* specify characteristics of the fictional simulation environment that would have to be true in order for a similar event to occur. For example, suicide bombing attacks can occur when the military power of the insurgency exceeds a threshold. Second, *user action preconditions* specify actions that the trainee must have previously taken for a similar event to occur. For example, a bombing during the recruiting of police officers can only occur if the learner has directed a company of soldiers to recruit in a city neighborhood. These two preconditions are represented as follows:

```
<WORLD_STATE_PRECONDITION CLASS="Insurgents"
  ATTRIBUTE="Military Power"
  OPERATOR="GREATERTHAN" VALUE="0.1" />
<USER_ACTION_PRECONDITION SUBJECT="Company"
  VERB1="Recruit Police"
  OBJECT="Neighborhood" />
```

During runtime, these preconditions are matched against the current world model, where high-level class variables in the preconditions (e.g. *Neighborhood*) are bound to simulation instances (e.g. the *Northwest Sector*) using a class hierarchy. When all preconditions of a story event unify with the simulation state, the story event executes. Information about the story event is presented to learners as pre-authored situation reports (text), and the underlying simulation state is modified to reflect the effects of the event. For example, the first event of the story described above causes reductions in the physical security of the targeted neighborhood, the support for the U.S. forces, and the military power of the Iraqi police force. These effects are encoded as follows:

```
<RELATIVE_EFFECT SUBJECT="Neighborhood"
  FEATURE="Physical Security"
  VALUE="-0.2" />
<RELATIVE_EFFECT SUBJECT="Neighborhood"
  FEATURE="Coalition Support"
  VALUE="-0.2" />
<RELATIVE_EFFECT SUBJECT="Police"
  FEATURE="Military Power"
  VALUE="-0.1" />
```

The stories of counterinsurgency operations that we collected and encoded into UrbanSim provide an initial proof-of-concept of our story-driven simulation approach. However, we believe that the success of this approach will depend most heavily on the size and relevance of the story corpus to particular training objectives. Toward this end, our current research focus is to develop authoring tools that enable military training developers, instructors, and learners to contribute their own stories to UrbanSim without the assistance of our research group.

4. Intelligent tutoring in UrbanSim

The UrbanSim Practice Environment presents learners with a huge problem space with many solution paths of varying degrees of quality. This complexity is a strength of the application in that it attempts to provide a realistic practice environment through modeling of human behavior. However, it also presents challenges in terms of learning. Specifically, large open learning environments that rely on discovery learning can be problematic for novices [10]. The need for guidance is a reoccurring and established principle of instructional design [12] and is delivered in two key forms: through the Urban Primer (discussed earlier) and by an intelligent tutoring system (ITS) that provides feedback in the UrbanSim Practice Environment.

Expert human tutors and the best ITSs deliver *formative* feedback – that is, “information communicated to the learner that is intended to modify his or her thinking or behavior to improve learning” [15]. Explicit feedback can be used for a variety of reasons, such as to verify the correctness of an action, explain correct answers, remediate misconceptions, reveal goal structure, and more. Feedback can be delivered immediately after an action, or after some delay. The best choices for feedback content and timing depend on many things, including task domain, nature of the skill being learned, the aptitude of the learner, whether the learner has a performance orientation, and more [15]. The UrbanSim ITS can support both immediate and delayed feedback.

One of the goals of UrbanSim is to teach about the broader and unintended effects of actions taken in stability-focused operations (specifically, directing action to anticipate 2nd and 3rd order effects). Understanding the role of non-player characters (NPC) in the PsychSim models is part of this. Each NPC agent acts to achieve its goals and makes decisions based on the state of the world. Although the learner cannot directly order non-U.S. Army NPCs to take (or not take) certain actions, she or he can certainly affect the world state. A key goal for the ITS is to help the learner understand this idea, and to take actions that 1) limit the ability of NPCs to take harmful actions, and 2) enable NPCs to take helpful actions. In other words, *the learner should be thinking about how their actions influence the actions of others* – the ITS frames its feedback in this light and attempts to reveal the reasoning behind NPC's actions: why they made the decisions they made, what consequences (seen or unseen) were most relevant, and under what circumstances different decisions would have been made.

To support learning of unintended consequences, we have implemented an *anticipate-wait-relate* tutoring strategy. That is, after the learner has proposed an action and the ITS has decided to apply the strategy, three steps are taken. (1) Elicit the anticipated effects of that action. That is, ask the learner to assess that choice by indicating, via drop down menus, how she or he expects that action to affect the world state. (2) After this input, allow the game to proceed for some number of turns (which is only one at the time of this writing, but longer delays are possible). (3) Finally, the ITS presents the learner with the *actual* results for comparison.

The system is able to provide feedback before or after the initial action proposal, or later (step 3), along with the comparison between anticipated and actual outcomes. Our focus thus far has been on this delayed form of feedback. Application of this strategy requires answers to at least two questions. First, what is used to trigger the strategy? In other words, when should the learner be prompted to anticipate the effects of an action? Second, how should the ITS support reflection on the results of the comparison? Of course, the learner could easily be asked to anticipate outcomes to every action, but this would quickly become a distraction. Also, learning could potentially occur by simply allowing the learner to inspect the predicted versus actual outcomes and learn from them. This also is unappealing, especially given the rich PsychSim models that drive NPC behavior. In fact, our approach leverages these models and the reasoning capabilities of PsychSim.

Our initial approach to answering the question of when to ask the learner to anticipate is to use PsychSim's look-ahead functionality. If progress toward the desired end-state of the world is about to decrease from a learner action, the ITS will ask the learner to anticipate the action's effect on features that are used to assess this progress. We currently focus on potentially damaging actions the learner can take. Regarding feedback, we have implemented an approach based on *causal chains* of the reasoning behind the NPCs. These causal chains reveal the state changes that occur based on learner actions, allowing the learner to see the connections between their actions, the world state, and the ensuing NPC actions. Figure 2 shows two examples of how causal chains show the effect of actions on the world state and their subsequent impact on lines of effort. Additionally, the ITS also queries PsychSim to reveal what conditions would have led to different NPC actions. For example, an NPC's ability to have taken one very bad action may have been impossible had the learner taken a different action at that turn. The ITS's aim here is to support the learner's reflection in imagining what other actions could have produced such world states.

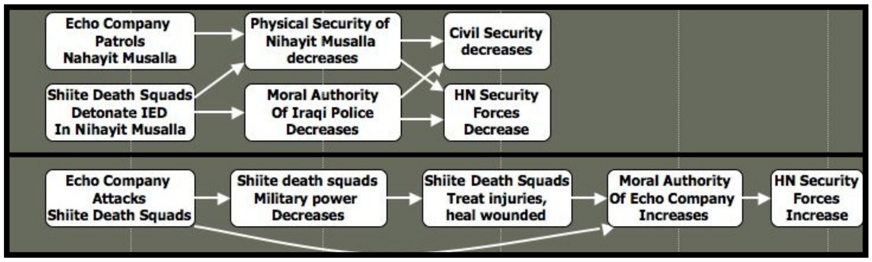

Figure 2. Causal chains in UrbanSim. Actions (on the left) induce changes to the world state, which subsequently lead to changes in *lines of effort* (e.g., Civil Security).

5. Conclusion

Contemporary simulation-based training systems, which utilize advances in computer game technology to address specific training needs, offer many new opportunities for the innovative application of AI technologies. However, the design space of possible combinations of learning objectives and technologies is large and largely unexplored. In this paper, we have presented our implementation of a design solution for a particularly novel combination of learning objectives and technologies. The skill set related to the conduct of military counterinsurgency operations was well suited to the design of turn-based strategy games and city management simulations built by the entertainment industry. However, to support the acquisition of skills in this environment, several "under-the-hood" AI technologies were required. To effectively model the behavior and complex interrelationships among stakeholders in urban environments, we used the PsychSim social simulation tool. To interject narrative elements that reflect the real-world experiences of commanders in counterinsurgency operations, we developed a story-driven simulation engine. To deliver informative feedback to learners about the counterinsurgency skills that they are practicing, we developed a new intelligent tutoring system that uses an anticipate-wait-relate tutoring strategy enabled by simulation-based look-ahead.

We believe that this design solution generalizes well to a wide range of other learning environments. The skill set related to effective counterinsurgency operations is analogous to a variety of other skills that involve managing and intervening in complex social environments, e.g. reducing gang violence in an urban environment through law enforcement. In addition to turn-based strategy games, the AI technologies that we employed in UrbanSim could be adapted for use in real-time strategy games and god-games. The exploration and evaluation of systems in this broader design space is an important direction for future work.

Acknowledgements

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions

expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

[1] Army Training and Leader Development Panel Washington DC, (2003) *The Army Training and Leader Development Panel Officer Study Report to the Army*. http://handle.dtic.mil/100.2/ ADA415810

[2] Clark, R. (2004) *Design Document For A Guided Experiential Learning Course*. Final report on contract DAAD 19-99-D-0046-0004.

[3] Gordon, A. (2009) Story-Based Learning Environments. In Nicholson, D., Schmorrow, D., & Cohn, J. (eds.) *The PSI Handbook of Virtual Environments for Training and Education, Vol 2*. Westport, CT: Praeger Security International.

[4] Hart, J., Wansbury, T., & Pike, W. (2009) So you want to us a game: Practical considerations in implementing a game based trainer. In J. Cohn, D. Nicholson, and D. Schmorrow (eds.) *The PSI Handbook of Virtual Environments for Training and Education, Vol. 3*. Westport, CT: Praeger Security International.

[5] Johnson, C., Birnbaum, L., Bareiss, R., and Hinrichs, T. (2000) War Stories: Harnessing Organizational Memories to Support Task Performance. *Intelligence 11*(1):16-31.

[6] Johnson, W. L. (2007) Serious use of a serious game for lan*guage learning. Proceedings of the International Conference on Artificial in Intelligence*. IOS Press, Amsterdam, 2007.

[7] Kenny, P., Hartholt, A., Gratch, J., Swartout, W., Traum, D., Marsella, S., & Piepol, D. (2007) Building Interactive Virtual Humans for Training Environment. *Proceedings of 2007 Interservice/Industry Training, Simulation & Education Conference, Orlando, FL, November 2007*.

[8] Kilcullen, D. (2006) Twenty-Eight Articles: Fundamentals of Company-level Counterinsurgency, *Military Review*, May-June 2006, 103-108.

[9] J. Kim, R.W. Hill, P. Durlach, H.C. Lane, E. Forbell, M. Core, S. Marsella, D. Pynadeth, & J. Hart, BiLAT: A game-based environment for practicing negotiation in a cultural context. *International Journal of Artificial Intelligence in Education* (in-press).

[10] Kirschner, P., Sweller, J., & Clark, R. (2006) Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75-86.

[11] Korris, J. (2004) Full Spectrum Warrior: How the Institute for Creative Technologies Built a Cognitive Training Tool. Proc*eedings of the 24th Army Science Conference, Orlando, Fl, December 2004*.

[12] Merrill, M. (2002) Pebble-in-the-pond model for instructional development. *Performance Measurement 41,* 39-44.

[13] Pynadath, D. & Marsella, S. (2004) Fitting and Compilation of Multiagent Models through Piecewise Linear Functions. *Proceedings of the 2004 International Conference on Autonomous Agents and MultiAgent Systems*, 1197-1204.

[14] Pynadath, D. & Marsella, S. (2005) PsychSim: Modeling Theory of Mind with Decision-Theoretic Agents. *Proceedings of the 2005 International Joint Conference on Artificial Intelligence*, pp. 1181-1186.

[15] Shute, V. (2008) Focus on formative feedback. *Review of Educational Research, 78*(1), 153-189.

[16] Wong, L. (2004) *Developing Adaptive Leaders: The Crucible Experience of Operation Iraqi Freedom*. Carlisle, PA: Strategic Studies Institute of the U.S. Army War College.