# Asymmetric Wars, Computational Intelligence and Infrastructure Security

V Rao Vemuri

University of California, Davis
rvemuri@ucdavis.edu

**Abstract.**

Ironically, advancements in information technology are impacting the international security agenda. With the advent of the Internet, globalization, increasing privatization of state functions, and the openness ethic that is pervading societies, a new type of threat, dubbed "asymmetric threat" appears to be emerging. In this new threat environment the world governments are faced with a number of low-intensity conflicts characterized by less discriminate attacks on civilian populations, infrastructures and the like. This paper considers the issue of using smart software agents in the development and deployment of gaming simulations, particularly with reference to games used in asymmetric warfare simulations.

## I. What is Asymmetric War?

Asymmetric warfare is not new. The civil disobedience movement led by Gandhi against a colonial power is a benign example from yesteryears. Terrorism is a more realistic example of contemporary relevance. Asymmetric warfare is a conflict between two unequal parties where one side's comparative advantages are pitted against its enemies' relative weaknesses. Typically, in an asymmetrical context one party is a technologically sophisticated government, a multinational corporation, or an international organization and the other party is some sort of an extremist group. Typically the extremist group (or a coalition of groups) uses weapons and tactics in ways that are unplanned or unexpected. Asymmetric attacks generally exploit vulnerabilities. Their battle spaces are cities and towns. Their targets could be critical national infrastructures. Their targets are both physical and psychological. The psychological impact resulting from an attack that brings down the cell phone network or the e-commerce of any technologically advanced nation would be grave indeed. Although the affluent countries seem to be a natural target, ethnic cleansing, guerilla wars, airplane hijacks and the likes, are also examples of asymmetric wars.

There are three categories of weapons one encounters in an asymmetric war. In the first category are weapons of mass destruction (WMD) such as long-range ballistic or cruise missiles. In the second category are cyber or cyber-based weapons, high-tech sensors, communications and weapons systems. A talented person with a PC and access to the Internet potentially can wreak havoc on the infrastructure of a country. The third category is the choice of the theater of operations; if one side attacks a power plant with a cruise missile, the other side responds with an attack on a metropolitan shopping mall with a suicide bomber. Such is the nature of asymmetric war.

Among various weapons of asymmetric warfare, the so-called "information operations" have a special relevance. What are information operations? In this context information operations can be defined as offensive, defensive and investigative actions taken in support of objectives that influence decision makers by adversely impacting the opponent's information systems while protecting ones own. These operations can be carried out essentially on four fronts: conventional air systems operations front, conventional battle systems operations front and the somewhat new and esoteric information warfare front and the economic warfare front.

Offensive information operations fall into three categories: attacks on infrastructure, deception and psychological operations. As much of the infrastructure is run by computers attacks on infrastructure necessarily involve the information warfare front. Activities on this front might include not only the detection and prevention of unauthorized intruders into our infrastructures but also methods of softening the enemy's defenses on these fronts.

Operations on the economic warfare front might include the creation of a database and the attendant data mining and knowledge management techniques to track the flow of money that finances terrorist activities, thereby creating nodes and links to other groups, peoples and their activities. Such a database would be drawing on worldwide information sources. After the creation of such a database, one can study the nature, relationships and links between agents of terrorism. Learning models, algorithms and training examples can be developed. The key to success in identifying and predicting threats lies in the quality of the paper trail left by financial transactions. Money is the lifeblood of most human activities and its trail is also very revealing in its scope. Organizations that use money as a medium of exchange are forced to leave paper trails in some form or another.

## II. Role of Artificial Intelligence

There is no well-developed theory to address the issues outlined in Section I above. The purpose of this paper is to explore the role of computer simulations of specially designed "war games" to study decision-making aspects of asymmetric warfare. In these interactive games human players will be pitted against believable software agents that come close to mimicking the capabilities of humans.

The goal in developing software agents is not so much in creating lifelike animations using physical laws and bio-mechanical modeling techniques. Rather, the goal is to achieve realism in cognitive modeling, a step beyond behavior modeling. The agents should react appropriately to perceived environmental stimuli and exhibit goal directed behavior. The cognitive models govern what an agent knows, how that knowledge is acquired, and how it can be used to plan actions. These agnets are vulnerable to common human foibles like emotion and stress. The objective is in achieving increased realism in the cognitive and emotional behavior of the game-playing agents and in capturing social situations. Finally the agents interact with each other to facilitate the simulation of group behavior. Such cognitive models are capable of directing the new breed of highly autonomous, intelligent agents that are beginning to find use in interactive computer games.

The design emphasis is on human-like behavior in a decision-making environment, not just on speed of the computer or the application of sheer computational power. Although IBM's Deep Blue defeated Kasparov, it is a fact that an expert human player who could hardly analyze more than a dozen variations per move stood up to the power of a computer's "millions of instructions per second." The next step is to endow the game-playing agent with the pattern recognition and decision making skills of a human!

The essence of conventional implementations of game playing on computers is search. The most straightforward way of selecting the best move is to explore all possible consequences (exhaustive search) of any action that can be taken in a given state. On a 3 x 3 board of tic-tac-toe, for example, with two players, this results in the need to explore 9! = 362,880 variations - not a formidable number for a computer. If one can think of the operations on a battle space as a board game resembling tic-tac-toe on a 100 x 100 grid, then 10,000! variations would result - surely a challenge even to the fastest of the computers.

It is true that AI search methods do not do exhaustive search; they are lot smarter than that. For example, inherent symmetries in the problem can be exploited to reduce the search burden. In complicated and realistic games this may not be possible. Other ingenious tricks and compromises are possible. In any event, the strength of classical search techniques hinges on our ability to perform a depth analysis and on the quality of static evaluation function we choose.

In minimax search, for example, player A associates a "value" to each possible state of the game and then seeks to minimize this value while player B seeks to maximize the same evaluation function. This approach suffers from two drawbacks:

(a) Assigning values to states is not a trivial exercise; needless to state that the search result depends on how these values are assigned.

(b) The assumption that B is a rational player whose value system is the same as that of A, and therefore always chooses the "best" defense as A interprets it.

In asymmetric games, this may not be a valid assumption. One way to overcome this difficulty is to make the evaluation function of B different from that of A. Indeed modeling the opponent's evaluation function is in itself a research topic. A natural way to do this is to observe a player's behavior during the course of a game and use it in conjunction with any prior knowledge about the player.

There are other issues that need further attention. An action by one player may lead to alternative states - each with a different probability of occurrence. That is, the evaluation function will attain its value only with a certain probability. This forces one to consider the issue of using probability distributions to describe the consequences of a move. Classical game theory techniques can be invoked to some extent to address this problem.

## III. Merits of Learning-based Game Playing Simulations

The gist of the discussion in Section II is this. Unless the evaluation function predicts the state values reliably, the search has to be carried deep into the search tree with the attendant cost of computation. Ways to reduce this cost is through instruction, advice taking, pattern recognition and generalization; in short, via learning. What cannot be captured through precise rules can possibly be learned from examples.

There are several approaches to learning. One possibility is to keep the essence of tree-based search in tact and apply a layer of learning on the top such that when the learning parameter is set to zero, the method degenerates to one of the classical tree search methods.

Indeed, machine learning in all its facets has emerged as one of the main research areas in AI. Learning, and its spin-off field of data mining and knowledge discovery, is emerging as one of the fastest growing application areas. Mining documents on the WWW, detecting fraud in daily transactional activities [Fawcett and Provost, '97] and tuning the evaluation functions [Tesauro, '95] used in search methods are three relevant examples.

Gaining insights into the opponent's evaluation function is a fruitful area of research. Typically one views the evaluation function as one that is comprised of several components. Evaluation of these components and combining them in "some fashion" to reflect their relative importance appears to be a fruitful area to pursue. It is possible to visualize a library of routines that compute important properties of the current "board position" (to use a board game metaphor). The size of the territory controlled, the number of pieces available for action, the number of opportunities to act, etc. are some of the attributes of a board position. What is not known is how to combine these pieces of knowledge (weighted average, probability distributions, etc) and how to quantify their relative importance.

There are many styles of learning: book learning, learning from examples, learning from mistakes, learning from simulations and so on. As asymmetric war games are characterized by imperfect information (as in card games like Bridge) and random components (as in dice games like Backgammon), deep searches are not feasible, nor are they likely to be rewarding.

In supervised learning, the agent learns from examples gathered from past activities - either historical or simulated. In comparison learning, pairs of agents are pitted against each other (perhaps in a round-robin fashion) over a given collection of training positions. In reinforcement learning (RL), the agents are allowed to a sequence of moves to completion and then they were simply told whether they "won" or "lost" the game. The temporal difference learning corrects one of the weaknesses of reinforcement learning. In RL, one error in the final end game is sufficient to "lose" a game - disregarding a sequence of otherwise good moves. Effects caused by stress, fatigue or other emotional states of the game-playing agent can alter the final outcome. RL can take into account this "unfairness" by performing the weight adjustments on the evaluation function more intelligently. However, RL also suffers from a drawback; it strips the game playing agent the ability to adapt in a domain-dependent fashion, by taking advantage of the background knowledge of a given situation. This ability is crucial in complex domains.

Finally, logic-based techniques such as explanation-based learning (EBL) and inductive Logic Programming (ILP) do have the ability to endow learning capabilities to deliberative agents. In this scheme one uses logic to represent domain knowledge as well as individual and collective behavior and EBL and ILP methods for learning.

## IV. Simulated War Games

Conflict simulations are models of military confrontations and can serve as a testbed for studying the learning behavior of AI agents because of the following reasons.

(a) Availability of large amounts of crucial background knowledge.

(b) Diversity of the underlying models will pose a challenge to the generality and adaptability of the agents.

(c) Utility of intelligent computer opponents for military training and strategic decision-making.

(d) Scalability of the system.

In broad strokes the decisions made during conflict simulations are not too unlike the actions taken at a board game like Monopoly and Backgammon or a card game like Bridge. Whoever "controls more points" are essentially "in charge" of the situation.

The wargame taken up for study is Cyberwar XXI, a board game designed by Joe Miranda of Hexagon Interactive. The rules of the game call for actions and interactions of numerous agents at four levels of modern warfare: Battle, System, Economic and Information Levels.

- "Battle Level" is where conventional ground and sea forces clash.
- "System Level" is where air power is pitted against national infrastructure.

- "Information Level" is where cybernetic, intelligence and special operations forces conduct combat using computer viruses, electronic warfare, and media manipulation.
- "Economic Level" is where information about the financial transactions of the opponents are tracked to gain better insights on the participants or where actions like sanctions are used to coax the opponent to a different point of view.

Without delving into the details of the game, suffice it to say that the players of the game in question are required to make several different types of decisions in different phases of the game. These decision points are summarized below:

- The selection of a InfoWar Squares, which in turn determines the number of Strategy Cards that are available to a player in each phase of the game. . The strategy cards give the players certain combination of assets and certain number of opportunities to use those assets.
- The selection of the prescribed number of (say M) strategy cards, from a deck of N, for each phase of the game. Given a limited number of strategy cards, each player has to decide on the optimum mix of assets to accomplish his/her goal.
- The selection of missions in each phase from the set of allowed missions, and deciding how many of the available resources to allocate to each mission.

These selections are to be made with the intention of "maximizing" one's own perceived "value" or utility. Although this perceived utility may differ from individual to individual, experience suggests that a "safe" way of playing the game is to work toward the goal of maximizing the overall InfoWar points one can control, in terms of gaining information dominance at the InfoWar level. So actions that maximize InfoWar point gain at the least loss in units are considered desirable.

## V. Treatment of Personality, Stress and Emotion

The AI engine is expected to simulate the effects of stressful inputs on emotional states of the players and the potential impact of these emotional states on the quality of decision-making. Critically, the simulation can capture not merely the actions of the real world players, but also can provide mechanisms for understanding their underlying maneuvers and objectives. It does so by quantifying factors such as political support and the "chaos" of transnational target audiences. The computer simulation can advance this understanding by its utilization of artificially intelligent, motivated "actors."

In the design presented here, these actors, also called "agents", are merely computer programs that draw upon conventional AI techniques like search, learning techniques such as neural nets and evolutionary programming techniques as well as methods of cognitive science to impart believable behavioral traits.

The Personality Engine (PE) is being designed to simulate the personality for a game-playing agent (see Figure 1). The PE works in two phases: (a) pruning options available to the agent before they are considered by the agent's static evaluation function. This is tantamount to an agent not even considering an option due to its emotional state. (b) modifying the weights assigned by the agent's evaluation function.

Personality is modeled using two of the major psychological theories that describe human personality: (a) Trait theory and (b) Needs-motivation approach. The structure of the PE consists of 4 main modules: Traits module, Needs-Motivation module, Physical module, and Learning module. The traits module emulates personality by assigning the agent a value within the range defined for each of a set of opposed traits and having these traits influence the agent's decisions. The needs-motivation module works by assigning the agent certain values of need for a number of defined factors (i. e., economic, religious, political, etc.). These values influence the agent's decisions by motivating it to satisfy its needs within the World State of the given game. The physical module models the physical state of the agent viewed as a human being. This feature will allow the agent's physical state (tired, angry, stressed, etc.) to influence its decisions. The learning module analyzes past game situations and predicts the opponent's personalities and strategies and uses feedback in the decision-making process.

## VI. Structure of the Agent(s)

An examination of the rules of the game reveled that the decision problem is fairly complex. As decision making by humans is not always rational, believable decision making behavior is not always rational behavior. This characteristic makes it difficult to depend on a rational agent or an agent that depends on systematic search methods to locate a goal state. Furthermore, given the potentially large number of players, the large number of options available to each and the fact that the "opponents" actions are not only hidden from general view but also they may include random actions makes the alpha-beta approach less attractive.

In addition to these considerations, there is a need to operationally decompose agent architecture in terms of some primitive capabilities. These constituent parts, when composed together, should give a variety of agent behaviors.

These considerations called for a design that is flexible, modular and scalable. Instead of having a centralized agent that does some sort of search to find the correct response, we decided to make the central agent very simple (mostly just a multiplexer) and delegate the processing to a bank of Advisors. The advisors would be comprised of relatively simple programs that compute a narrow aspect of the games, and each advisor would pass back to the agent an advice on what it thinks the agent should do. It would then be up to the agent to decide which advice to take (see Figure 2). This is not too unlike a couple of schemes published in the literature [Epstein, 94, Rahman and Fairhurst, 2000].

## Phases in a Turn

(A small sample from the design document)

1. **Chaotic Events**

   No decisions by AI

2. **Initiative Determination**

   No decisions by AI

3. **Mobilization (Selection of Strategy Cards)**

   Each agent selects certain number of strategy cards to use.

   3.1 Parameters: (number of cards that may be chosen, list of legal cards to choose from, game state that influences the value of the selected card)

   3.2 Considerations: (actions/impulses granted by the card and in which space, reinforcements received, limitations on the placement of reinforcements, cascading effects, Information Warfare cost incurred, history of opponent SC selection, etc.)

   3.3 Heuristics: Do not select cards whose requirements cannot be met, select the cards with the intention of using them in a specific way, Consider look-ahead planning at this stage, etc.

   3.4 AI Ideas: Assess the value for each potential card combination and action/impulse sequence, develop a set of rules to guide the selection of cards

4. **Information Space Warfare**

   …

5. **Airsystems Space Warfare**

   …

6. **Airsystems Space Warfare**

   …

7. **Economic Conflict**

   …

8. **Reconstitution**

Head Agent

The AI Control Engine (Head Agent) is the main interface between the game and the agent (although the game's Database/Data structures may also be accessed by other components of the AI engine). The Head Agent receives requests from the main simulation loop whenever there is a need for decision-making assistance from the AI side of the game. This request should include the context (the stage of the simulation where a decision is to be made) of the simulation. Upon receiving this information the head agent will ask the bank of advisers for suggestions on what to do. For instance, if the head agent receives a signal requesting assistance in picking the strategy cards for the game, the head agent will pass this signal to all the advisers. The strategy cards will then be picked considering the suggestions of all advisors.

Warfare Agents at each Level

Following our initial design each agent will have a series of advisors for each task involved in the decision-making process. Therefore we can visualize the possible advisors by looking at the tasks the agent has to perform in order to make the overall decision.

For example, the IW agent will have to make the following decisions (tasks):

- Strategy card selection
- Play space selection
- Mission selection
  - Decide on targets
  - Decide on missions
  - Decide on Units to carry out missions

High-level advisors (Staff Advisors), at least three within each space (one corresponding to each of the other spaces). Possibly use for other type of staff advisors such as a learning advisor that examines past situation outcomes.

1. Looks at the list of advice, and deletes items that would be overly detrimental to their space (example: if one of the Battle-Level advisors suggests a card that would have a large negative impact on the Information-Level, the IW high level advisor to the Battle-Level sub-agent would delete that action from the proposal list)
2. Possibly looks at the list of possible actions before the low-level advisors remove any objectionable ones.
3. Game Information (Database and/or Data Structures).

Within this design there is great deal of flexibility, both in terms of the scope of problems it can handle, and in terms of development. By forcing the advisors to focus on small enough areas, they should be efficient enough to run within the lifetime of the universe. The combination of their advice (by using the trust values) will generate a fairly realistic (but probably not optimal) agent.

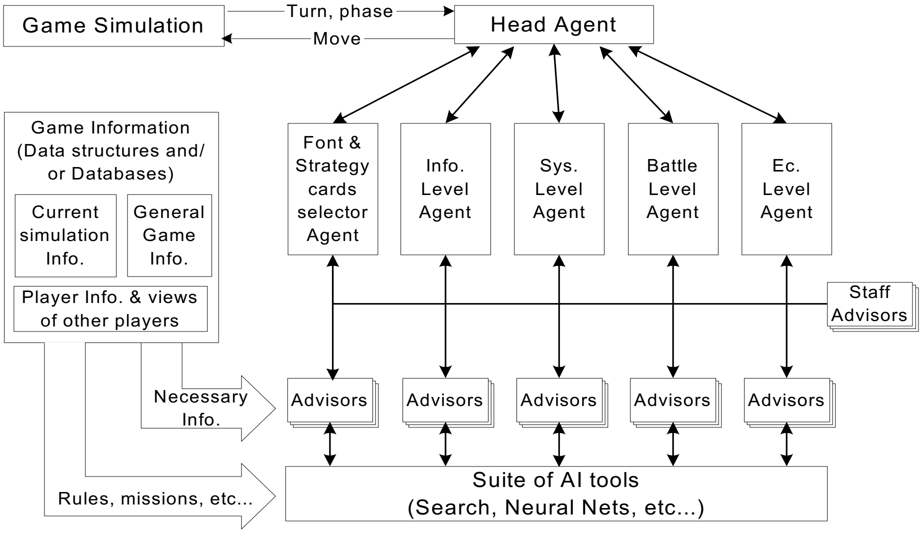

**Fig. 1.** Agent architecture diagram.

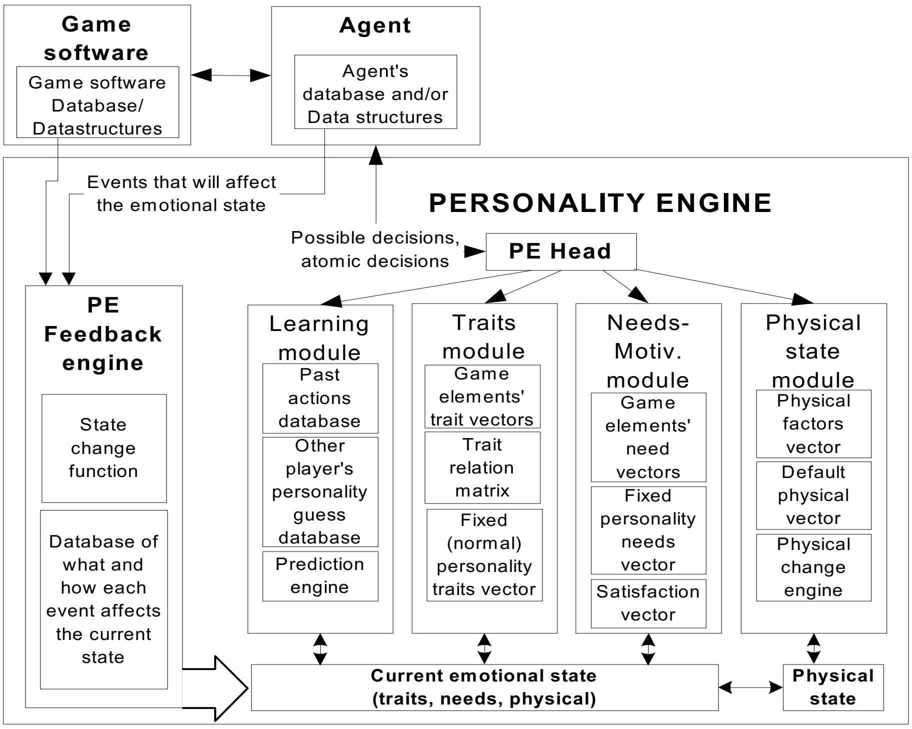
Game software
Game software Database/ Datastructures
Agent
Agent's database and/or Data structures
Events that will affect the emotional state
PERSONALITY ENGINE
Possible decisions, atomic decisions
PE Head
PE Feedback engine
State change function
Database of what and how each event affects the current state
Learning module
Past actions database
Other player's personality guess database
Prediction engine
Traits module
Game elements' trait vectors
Trait relation matrix
Fixed (normal) personality traits vector
Needs-Motiv. module
Game elements' need vectors
Fixed personality needs vector
Satisfaction vector
Physical state module
Physical factors vector
Default physical vector
Physical change engine
Current emotional state (traits, needs, physical)
Physical state

**Fig. 2.** Personality engine architecture diagram.

## Acknowledgement.

Thanks to Tony Zalewsky, Joe Miranda of Hexagon Interactive for introducing Cyberwar XXI. My students Chris Marsh, Daniel Castaneda and Leland So are working on various facets of the design. Discussions with Sunil Vemuri of Media Labs, MIT helped us to develop a focus. Daniel Castaneda is designing the personality engine.

## References

Alonso, E and D. Kudenko, Machine Learning Techniques for Adaptive Logic-based Multi-agent Systems: A preliminary Report, University of York, York, U. K., ??

Atkin, M. S., Westbrook, D. L., and Cohen, P. R., "Capture the Flag: Military Simulation Meets Computer Games," In Papers from the AAAI 1999 Spring Symposium on Artificial Intelligence and Computer Games, 1999.

Epstein, S. L., "For the Right Reasons: The FORR architecture for learning in a skill domain," Cognitive Science, 18(3): 479-511, 1994.

Fawcett, T. E and F. Provost, "Adaptive Fraud Detection," Data Mining and Knowledge Discovery, vol. 1, no. 3, pp 291-315, 1997.

Funge, J. D. AI for Games and Animation: A Cognitive Approach, A. K. Peters, Natick, Mass. 1999.

Furkranz J. and M. Kubat (Ed.) Machines that Learn to Play Games, Book Manuscript under review, 2001.

O'Brien, Kevin, "Intelligence gathering on asymmetric threats," Jane's Intelligence Review, pp 50-55, October 2000.

Rahman, A. F. R. and M. C. Fairhurst, "Multiple expert classification: a new methodology for parallel decision fusion," Intl. Journal of Document Analysis and Recognition, Vol. 3, pp 40-55, 2000.

Stern, A. "AI Beyond Computer Games," In AAAI 1999 Spring Symposium on Artificial Intelligence and Computer Games, Technical Report SS-99-02, AAAI Press, 77-80, 1999.

Tesauro, G. "Temporal difference learning and TD-Gammon," Comm. ACM, Vol. 38, No. 3, pp 58-68, 1995.

Wilson, I. The Artificial Emotion Engine: Driving Emotional Behavior, http://artificial-emotion.com