Behavior Modeling in Commercial Games

David E. Diller
William Ferguson
Alice M. Leung
Brett Benyo
Dennis Foley

BBN Technologies
10 Moulton Street
Cambridge, MA 02138
617-873-8000
ddiller@bbn.com, wferguson@bbn.com, aleung@bbn.com, bbenyo@bbn.com, dfoley@bbn.com

Keywords:
behavior modeling, behavior authoring, computer games, training

ABSTRACT: *This paper presents findings from our ongoing survey of commercial game technologies for military relevance. In this paper we focus on those technologies related to behavior modeling and the authoring of behavior models and attempt to lay a foundation for behavior modelers to understand the gaming industry – its methods, goals and accomplishments. We extrapolate where the industry is going and what future synergies it is likely to contribute. We focus on behavior generation and other accomplishments that are relevant for training applications and especially for non-programmer authorship of content for training applications. We will argue that the gaming industry's modeling capabilities are currently of little theoretical interest but that pragmatically, the current accomplishments of the gaming industry can have a substantial positive impact on training and modeling and that this impact will increase over the next few years.*

1. Introduction

The computer game industry appears to be increasingly concerned with the development of sophisticated game characters. The industry has deep pockets from which to invest resources into game development and has achieved impressive advances and a huge player base. The military training community has long sought to leverage game technology. Furthermore, interest in game-based technologies and their actual application is on the upswing. Should behavior modeling researchers jump onto the gaming bandwagon? We will try to answer that question, particularly for those with an interest in developing models for use in training.

Commercial game developers have some of the same goals for the simulation of human behaviors as training application developers such as creating an immersive simulated world. Additionally, behavior creation is attracting increasing attention from game developers. As advanced, high-resolution graphics become commonplace, game developers are increasingly relying on "game AI" (i.e., behaviors of synthetic entities) to distinguish their game from competitors [1]. At the same time, game developers have become increasingly concerned with producing realistic and robust behaviors [2].

Game developers are interested in producing entities that are more adaptive to new situations, harder to game, less predictable, and more variable. Currently, the gaming community uses synthetic entities to play a range of roles and makes use of well-crafted scenarios to focus the user experience and highlight appropriate behavioral capabilities while downplaying behavioral imperfections and inadequacies. These needs and intentions for the development of behaviors in synthetic entities are shared with the training community.

However, unlike the behavior modeling community, game developers are interested more in the "illusion of intelligence" – by this we mean that behaviors in games are designed make synthetic entities appear to be intelligent, rather than actually intelligent [3] [4]. Paul Tozour, AI Programmer for Deus Ex 2 says regarding games:

> The whole point is to entertain the audience, so no matter what you do, you need to make sure the AI makes the game more fun. If a game's AI doesn't make the game a better experience, any notions of "intelligence" are irrelevant.[1]

[1] http://www.pcgamer.com/eyewitness/eyewitness_2002-09-18.html

There are many indirect reasons for behavior modelers to keep an eye on the game industry. Commercial games provide extensive examples of how simple artificial intelligence (AI) behavior can be used effectively in virtual worlds. Modelers who build training applications may be able to take advantage of the fact that trainees, like game players, will only expect a small slice of behavior from synthetic entities populating the training scenario.

A key similarity between the development of intelligent behavior in games and training systems is that in both cases, there is a need to facilitate cooperation between the content experts who describe and specify the desired behaviors and the programmers who implement them. In the game industry, programmers are responsible for the software and basic functionality (often called the game "engine"), while designers are responsible for the content and game-play, the game's goals, and the style of play. In order to enable game designers – typically nonprogrammers – to control the events and behaviors in a game and work semi-independently of the programmers, tools such as scripting languages are often employed. These tools allow behaviors to be defined in a higher-level, English-like language. This shared responsibility is similar to how, in the military human-behavior modeling community, a subject-matter expert often works side-by-side with a behavior modeler to develop appropriate behaviors. However, the game community appears to have found methods for enabling game designers to work with less dependence on programmers than is typically seen in military behavior model development.

Another relevant trend in the commercial gaming industry is the increase in content authoring by users. While players have "hacked" games for years (enabling them to, among other things, cheat a game by increasing equipment, abilities, or points), the gaming community is increasingly supporting game modification by providing software tools – often the same ones that were used to construct portions of the game itself – that facilitate game modification and new content creation. Games can be modified in a number of ways. Media (e.g., graphic & sound) files can be changed to modify the presentation of objects or to create new objects in the game. Entire "levels" (scenarios or maps) can be constructed by changing the map files representing the spatial architecture of the game. Rules of the game can be manipulated, and in some instances the behaviors of game characters themselves can be changed.

Web sites such as GameSpy's Fileplanet (www.fileplanet.com), Planet Unreal (www.planetunreal.com), and Neverwinter Vault (www.nwvault.com) provide a way for millions of gamers to share modifications, news, and interact in message forums. Game companies encourage these modifications because they increase game-play and sales. Game companies are now even holding workshops on how to modify their games [5].

It is clear that both the gaming and behavior modeling communities share an interest and desire in better behavior-modeling capabilities, and on the surface share some of same needs and intentions. However, it is less clear to what degree a mutually beneficial relationship can arise, given the differences in their goals, current practices, and audience. It is our goal to explore any potential synergies and to elucidate them to the advantage of both communities.

2. Behavior Generation

We examine the generation and authoring of game character behaviors from a psychological perspective, looking at how and whether game behaviors are based on the types of processes that underlie human behavior.

2.1 Components of behavior in games

The domain in which computer game Non-Player Characters (NPCs) must operate – although simpler than the real-world – often require NPCs to perform a wide range of functions and to integrate capabilities that include sensing the environment, reasoning about its spatial layout, planning and executing actions, as well as communicating and coordinating with other NPCs or players. This has been recognized by John Laird and colleagues [6][7], who suggests that computer games make an ideal testbed in which to perform research on "human-level" intelligence.

Given the niche oriented and specialized nature of much of the research in cognitive psychology and artificial intelligence, it is unsurprising that the vast majority of human behavior models are specialized for specific behavioral phenomena, and are not integrative across a range of phenomena or behavioral disciplines. Integrative models are rare, with most in existence based on cognitive architectures with long running research programs such as Soar [8] and ACT-R [9].

In this section, we examine whether, how, and to what extent behavior representations in commercial games include the same types of processes underlying human behavior. We explore game NPCs and their representations of memory, sensation and perception, motor control, decision making, learning, and language processing – the basic building blocks of human behavior.

2.1.1 Sensation and perception

The sensory mechanisms by which game characters "see" and in some cases "hear" the world range in complexity from the extremely simple to the surprisingly sophisticated and complex. Compared to the rich and compelling virtual worlds provided to human players, the world representations designed for synthetic characters are quite impoverished. Environments are typically reduced to simplified representation used only for navigation. Object detection occurs by performing collision tests radiating out from the NPCs location. Sensory routines often only perceive entities and events that can influence the NPC's reactions (e.g., *Halo*) – typically human players and the actions they produce. Routines are run periodically in order to gather relevant events. Vision is often represented by a two-dimensional view cone, and a line-of-sight operator is used to detect whether an object is visible to the NPC. Unfortunately, line-of-site detection does not take into account real-world cues such as motion detection, depth perception, and coloration or texture differences. Hearing is often implemented using even simpler models. For example, *Halo* models the distance that an NPC can hear a sound as a function of the volume of the sound scaled by a "hearing sensitivity" parameter. Smell is sometimes implemented using the same mechanisms as sound.

In contrast to these simple cases, the focus on stealth, evasion, and surprise found in *Thief* and *Thief 2*, required a more complex perceptual system. Multiple three-dimensional view cones are used to represent focal and peripheral vision differences – each with differing probabilities for the presence, location, and identity of an object. Furthermore, view cones are sensitive to additional factors such as motion or darkness levels. Objects relevant to NPCs have intrinsic visibility values, scaled by lighting levels, degree of motion, and exposure levels. Sound processing in *Thief* is also fairly sophisticated, complete with directional and attenuation information.

In order to develop NPCs that do not rely on preexisting navigational waypoint maps, researchers are creating methods based on techniques developed for robotic exploration (by which an NPC explores a region collecting sensor data), which are used to construct a high-level spatial map of the region (e.g., [10], [11]). Like all other mechanisms, the realism of sensory processes is subservient to game-play. In *Halo*, an NPCs ability to see a player is constrained so that the NPC can only see the player if the player can see the NPC. This is because players often felt cheated when a player is seen by an opponent they do not see. Additionally, the time required to sense and recognize an object was removed after testing showed people interpreted it as the NPC being broken [3].

2.1.2 Motor control

The kinds of actions that game characters are allowed to perform are often typical of human characters, although in science fiction or fantasy games the characters may have numerous special abilities. While the allowable actions may be human-like, often the constraints on these actions are not. In many instances characters are afforded superhuman capabilities in order to make game-play more interesting. For example, the female assassin character in *Halo* is able to jump higher than player characters, and access areas specifically built for her use and not available to the player. Game character reaction times have often been manipulated in order to provide challenging opponents.

2.1.3 Memory

Cognitive memory mechanisms in computer game characters are perhaps the least like their human counterparts. Typically, memory mechanisms are simply data structures representing the current state of the game, and data caches designed to reduce processing requirements for activities such as navigation and path finding. Memory limitations are not based on any psychological theory of human memory, but instead on the memory and processing constraints of the computing platform. Differences between short- and long-term memory, memory limitations due to decay and forgetting, and the processes by which memories are rehearsed, encoded, and retrieved, are not found in game characters. Most game characters do not retain memories of encounters with players in the game. Entities outside of sensor range are immediately forgotten. There are, however, some exceptions. For example, characters in *Titanic, Adventure out of Time* will react differently depending on whether they have previously met the player. Some strategy games such as *Civilization: Call to Power*, incorporate a memory of the number of units destroyed by a player at any given location and take this information into account when performing path planning [12]. Note, however, that these capabilities are more along of the lines of clever techniques to enhance game realism, rather than human-like memory mechanisms.

2.1.4 Decision-making

The most common representations for decision-making processes in game characters are finite-state machines (FSMs). Character behaviors are modeled as a finite set of states with transitions between them in the form of a directed graph. The character resides in only one state at a time, with transitions between states driven by the conditions and actions occurring in the game. FSMs are simple, easily understood, can be graphically represented, are computationally inexpensive, and have compact representations. There are a number of extensions to FSMs which make it possible to develop more complex behaviors, including Stack-based FSMs, Fuzzy State Machines, Hierarchical FSMs, and Probabilistic FSMs. However, current FSM techniques do not scale well and their propositional representation can make behaviors difficult to represent.

Rule-based approaches to behavior have also been implemented in games (e.g., *Age of Kings*). Typically, in these systems a set of if-then expressions are used to encode simple condition-action pairs. In most cases FSMs, and rule-based approaches are represented in game-specific script or code, rather than within a general purpose implementation.

Another behavior representation technique seeing some use in the game community is that of smart terrain or smart environments. In smart environments, the objects in the world contain information on how game characters can interact with them. Section 3.4 describes smart environments in more detail.

Recently, a few games have developed character behaviors using goal-directed reasoning techniques. In this technique game characters have a set of goals and actions are chosen in order to best satisfy the most relevant goal or goals. *No One Lives Forever 2* implements a simple goal-directed system in which characters have multiple goals and choose which goal to pursue given the current game conditions. However, the type of behavior executed in response to the goal is hard-wired – no plan of action is developed to satisfy the goal.

Finally, there are a small number of games which have used neural networks or genetic algorithms to specify behaviors. We describe several of these in the following section.

2.1.5 Learning

A number of AI learning techniques have been employed in computer games, including decision trees, neural networks, and genetic algorithms. While many of these techniques are used during the game-development process, few games actually ship with the learning mechanisms turned on for fear that the game will learn poorly and provide a less than desirable game-play experience. There are notable exceptions, however. *Black & White* has learning as a central component in the game. In *Black & White* a player controls a Creature that learns from the player's actions and any positive or negative feedback provided by the player. The Creature, based on the Belief-Desire-Intention agent architecture, learns using a variety of learning algorithms, including decision trees based on Quinlan's ID3 [13] and neural networks.

Relatively few games have used genetic algorithms. A notable exception is *Cloak, Dagger, and DNA*, which used genetic algorithms to evolve opponent strategy. Between battles with the player, the player can allow the opponent's DNA strands to compete and evolve through a series of tournaments.

2.1.6 Communication

In most games, communication between players and NPCs is quite limited, with the dialog consisting of a few scripted phrases. A slightly more complex technique used to enrich game interaction is the use of dialog trees, in which there is a tree of game character statements and possible player responses. The player typically picks a response from a menu interface, which in turn dictates the game character's response, thereby walking through the conversation tree. Terminal leaves in the tree may result in actions (other than communication) by the NPC. Some games (e.g., *Baldur's Gate*, *The Elder Scrolls III: Morrowind*), make the dialog sensitive to situational context – what has been done in the game, with whom the player has talked, current teammates, etc. Games with speech recognition capabilities for giving simple orders to NPCs are just beginning to emerge[2] (e.g., *SOCOM: US Navy SEALS*, *Tom Clancy's Rainbox Six 3*) with more rumored to be released shortly (e.g., *Unreal Tournament 2004*). In addition, there are now companies producing speech recognition Software Development Kits (SDKs) for game platforms.

3. Current Behavior Authoring Capabilities and Limitations

Increasing interest in user generated content and AI for game behavior has already led to some behavior authoring tools. As the game world develops more complex NPC behaviors, the accompanying tools will give non-programmers ways to specify behaviors which are currently only possible to create through software programming. We describe these tools and techniques – their strengths and limitations, and ways they might contribute to behavior modeling for training applications.

3.1 Parameterization

[2] *Seaman* by Sega was an early game employing speech recognition technology

Parameterization is a simple yet significant approach to behavior specification. Some games have easy-to-use, graphical authoring tools that allow the user to modify the behavior of characters by setting personality or manner parameters. This kind of naïve parameterized AI provides meaningful handles to control behaviors which are intelligible to the lay user. It is not necessary to possess human behavior modeling or programming expertise in order to use these tools. For example, in Operation Flashpoint – a game involving synthetic soldiers operating in military units, an author can pick a policy which is the behavioral goal for a whole unit. When creating a new scenario, the author picks an overall policy which is the behavioral goal for the unit (such as engage at will, seek and destroy, or hold position) as well as a manner (such as cautious, aggressive, reckless). The author cannot actually edit the behavior model, only the input parameters. However, these types of parameters make intuitive sense to a training designer or SME, and such tools might satisfy some military customization needs.

Role-playing games have long used a set of numerical ratings for categories such as intelligence or strength to calculate character capabilities. This kind of behavioral parameterization has some similarity to Button Theory and the five-factor model of personality. Button Theory is a training system where students interact with a computer tutoring system through a limited set of responses (buttons). Although each button represents a simple expression of mood, question, or action, pushing the buttons provides enough expressiveness for the system to respond with tailored learning [14]. The Big Five personality dimensions and the five-factor model are popular psychological approaches to studying personality traits [15]. There are alternative sets of factors and dimensions in use, but the relevant idea is that relatively few traits are needed to describe important variations in human personality. To the extent AI can generate behavior from traits, this suggests that game AI relying on relatively few character parameters should still be able to generate a wide spectrum of behaviors.

3.2 Programming/scripting

Authoring tools for programming and scripting behavior span a wide range of expressive power and ease of use. Some games have graphical tools for doing limited scripting; these types of specialized tools will be discussed separately below. In general, they give the author the ability to generate only very particular kinds of behavior and limited logic. In other games, significant amount of game logic are defined outside of the game's core software code-base. The external modules are often written in a scripting language, and an interface for script editing may be embedded in the game software. Games vary in how much character behavior can be controlled through scripting. The *Quake* and *Unreal* families of games provide extensive scripting capabilities through the script languages QuakeC and UnrealScript, respectively. These game developers have made the game code publicly available, with only the graphics engine code proprietary. This has made these games popular testbeds for academic research [16]. The use of scripting languages as an authoring tool gives users or game designers the power to modularly expand and extend behaviors. However, it requires that authors develop skill at script programming. Additionally, game-specific scripting languages will often lack programming development tools for debugging, making it difficult for an author to create large or complex development.

Another programming approach to behavior authoring is to interface with game software through API's. This approach can allow authors to use a programming language completely external to the game to write their own synthetic entities with arbitrarily complex behavior models. The research community has made extensive use of this capability, developing additional tools for interfacing software agents with games (e.g., [17] [18]). While powerful, the use of full-fledge software programming to author behavior is only useful to programmers. Most training system developers and subject matter experts will need to look elsewhere.

3.3 Specialized scripting techniques

3.3.1 Trigger and rule systems

A trigger system allows authors to define simple conditions that result in specific responses. Typically, the condition would be an interaction between a character and an object or location in the game environment. Rules are more complex but have the same test/result structure. The test usually consists of simple conditions and logical combinations of conditions. The result of a rule firing can be a simple action or a change in policy.

An example trigger:

- If an enemy unit comes within sight of my castle guards send out cavalry units to attack them.

An example rule:

- If target city has city walls and my attacking force doesn't include a battering ram, do not attack and instead, build a battering ram.

The authoring of triggers is structured so that it is often supported with graphical tools that prevent syntax errors and can preempt many simple semantic problems (e.g., preventing type miss matches by only allowing blanks to be filled in with certain types of objects). Rule authoring is usually supported by text-based languages. This allows for many of the strengths and shortcomings of programming languages. Rich behaviors can be expressed but rule interaction errors are common and subtle. The required expertise to author rules is therefore much higher.

3.3.2 Dialog trees

A dialog tree is a simple tree-based representation of possible conversations between a player and an NPC (described above in Section 2.1.6). Bioware[3] has developed a number of role-playing games (e.g., *Planescape Torment*, *Neverwinter Nights*) which make extensive use of conversational interactions through dialog trees. Games such as *Neverwinter Nights* provide graphical conversation editors for constructing dialog trees.

3.4 Smart environments

Behavior models are usually created from the point of view of an intelligent agent operating in a world with inactive objects. The game world has occasionally employed the opposite paradigm, utilizing smart environments to produce interesting behaviors. Such games populate a complex world with simple agents. Corresponding to this paradigm, tools for such games enable the author to create or edit objects in the game environment in order to influence agent behaviors. For example, a tool could support drawing invisible paths or waypoints onto a game map to specify routes for characters to follow. Although the smart environment model appears to lack psychological validity, behavior authoring tools that utilize this approach can be used effectively to produce reasonable behaviors.

The Sims game uses the smart environment model, and there is a corresponding authoring tool available to in-house developers. In this game, players create dollhouse characters by specifying values for five personality parameters. The game animates the characters with default behavior driven by a set of eight needs whose relative importance depends on the personality parameters. A character's current primary goal is visible to the player, who can use this information to predict the character's likely behavior. In the game world, everyday objects, such as chairs or food, advertise their ability to satisfy particular needs. Characters move toward objects based on a match between current needs and the advertised object effects, with geographic vicinity and randomness exerting some influence. When characters interact with objects, the object takes control of character behavior and executes a script. The authoring tool used by the game developers facilitates the creation of new objects for influencing character behavior. The player is also able to directly control the characters, causing them to exhibit short-term deviation from their default behaviors. This "semi-controlled" entity model may be interesting for training applications, particularly if it could be combined with machine learning. With the semi-controlled model, a trainer could interfere in a running simulation and direct NPC behavior if the default behavior was undesirable.

3.5 Machine learning

Some games employ machine learning or neural nets to "teach" agents behavior. Tools for training game agents have been used by game developers in-house, although the trained agents behavior specifications are then frozen before the game is published. One example of a behavior authoring tool that utilizes machine learning is *Black & White*. In this game, the player plays a god and trains a Creature by providing positive and negative reinforcement for behaviors. The Creature might be rewarded for helping to carry building materials, or punished for eating a villager. A benefit of using such tools is that agent behavior can be refined incrementally without explicitly enumerating rules and exceptions to rules for behavior. However, this authoring technique is limited by its dependence on the specific data representations used in a game. For example, it would be impossible to train an agent to act a certain way towards your enemies if there is no data representation indicating whether a character is your enemy. Another limitation is that these tools can require excessive input for training, making the process impractical. Machine training can also produce unintelligible learned behaviors, and unintended behaviors may be difficult to predict or correct. In other words, the entity may appear to learn one thing but actually learn something else, such as a special case of the desired behavior.

3.6 Third-party tools

Recognizing the growing demand, a number of third party vendors have developed tools designed to make authoring easier and faster. These authoring tools are designed to be used by game developers and training application developers. One advantage of this type of third party behavior authoring tool is that they are usually intended to work with multiple games, whereas tools produced by game companies are more likely to be game-specific. Some of these tools include integrated development capabilities such as debugging and source control support, making it easier to develop complex modifications.

[3] http://www.bioware.com/

Recently, there has been the introduction of AI "middleware" into the commercial game market. There are currently a handful of products (e.g., AI..implant[4], DirectIA[5], RenderWare A.I.[6], SimBionic[7]), that provide tools and authoring environments for developing NPC behaviors independently of any specific game engine or environment. Many of these products provide tools for developing low-level behaviors of virtual characters in a game world – path finding, complex animation and body motion control are two common capabilities. More complex, higher-level behaviors are typically constructed using finite state machines or decision trees with graphical authoring systems to facilitate development. It remains to be seen whether or not these products will become commercially viable.

Some human behavior modeling systems like Soar or ACT-R, offer more complex models which lay claim to varying degrees of psychological validity. Tools for authoring behaviors using these modeling systems have not been developed specifically for game use. Currently, using such modeling systems requires significant programming skills.

3.7 Implications for the modeling community

Behavior authoring tools for games are becoming more widespread and important. The behavior modeling and training community will benefit by familiarity with and keeping up-to-date on developments in these authoring tools. A direct benefit is that training applications may be able to take advantage of simple behavior models, easy-to-use graphical authoring tool, and still produce realistic enough behavior. Second, if deeper models are needed, some game tools for behavior authoring might be extensible or compatible with more complex, psychologically valid models. Third, modeling researchers may want to use game authoring tools to customize game-based experimental testbeds for examining both human and model behaviors. Fourth, existing behavior authoring tools may indicate what sorts of modification capabilities are most useful to authors, and can be used a guide when developing models that allow these sorts of modification.

Game behavior creation techniques emphasize close coupling between the virtual world and synthetic agents. Generally, the two are developed in parallel, so that the game AI takes advantage of the world representation and the virtual environment compensates for weaknesses in the AI. In contrast, modelers have an interest in developing models which are robust and less dependent on their environment, and training system developers may need a level of realism in their scenarios which preclude certain environmental compensations. However, a subset of the synthetic behaviors needed for training applications can be provided by the types of simple, psychologically invalid models used in games. For some of these behaviors, authoring tools developed for the gaming world already exist. As AI and authoring both increase in importance in the game world, modelers should see the emergence of more useful behavior authoring tools.

4. Future Developments

Current trends in the commercial computer gaming industry suggest that the future of computer games will provide increased utility for training and behavior modeling.

Game companies are increasingly encouraging and facilitating user-generated content to keep their games fresh and interesting. Companies have found that allowing users to author content dramatically increases the overall amount of content available, increases customers' investment in the game, and generates grass-roots publicity and enthusiasm for a game. We anticipate that this movement towards increased capabilities for user-generated content will result in more authoring tools, making it easer to customize games for training applications. However, while these tools may contribute to easier customization, they are likely to fall short of satisfying certain needs, including the ability to produce psychologically valid behavior models. These limitations may be addressed through cooperation with game companies to produce more customized tools and game modifications. In fact, this approach has already been used on a number of projects with promising results. [20]

With game companies increasingly employing intelligent synthetic players as a means of distinguishing their game from their competitors, we will most certainly see continued improvements in game AI. There is already a trend away from traditional finite-state-machine and scripting approaches to more complex, but robust techniques such as goal-directed reasoning. While it is likely that the development of deeper AI techniques will have potential benefits to training applications, there are requirements that may not be addressed by game development efforts. Behavior authoring middleware tools have recently become available, but it remains to be seen the degree to which they will catch on. Centrally, while NPC behavior is valued for entertainment, characters in training scenarios must

[4] http://www.ai-implant.com/
[5] http://www.directia.com/
[6] http://www.renderware.com/ai.asp
[7] http://www.stottlerhenke.com/products/index.htm

often react with realistic responses, regardless of the entertainment value.

Computer gaming is at the forefront of multi-user collaborative interactions. In and out-of-game communication tools for text messaging or VoIP are now available and will continue to become more prevalent and inexpensive. Games have developed support for handling the logistics of game distribution and updating, multiplayer on-line sessions, and user-content distribution. Integrated, computer-based military training systems (such as DARPA's DARWARS project) will need similar capabilities for training application distribution, scheduling players and sessions, matching participants, using persistent user records, and tracking player availability status. Game concepts such as "lobbies" (discussion areas) for coordinating multiplayer sessions will also be useful.

By considering the trends occurring within the game industry we have identified a number of potential benefits for continued synergy with the game development community. We expect that training applications and modeling research will benefit from these advancements. At the same time, the fundamental difference in goals between games and training mean that game-based technologies and tools are likely to require some modification in order to satisfy training and modeling research requirements.

5. Appendix A: Fallacies about the Gaming World

There is great potential for military training to benefit from advances in the commercial computer game industry. However, advocacy for broader application of game-based concepts or technologies must be based on an accurate understanding of the gaming world. When planning for how aspects of commercial games may be extended for purposes beyond entertainment, decision makers should not be misled by several common misconceptions about the current state of the computer gaming industry.

Fallacy 1: The US is leading the way in game playing.

The size of the US and European interactive entertainment markets were comparable in size, at about $8 to $9 billion in 2001 [21]. Japan was smaller at $6 billion. However, this actually suggests a stronger penetration of game playing among those of prime game-playing age, since Japan has smaller, older population than the US. In Japan in 2001, 29% of those age 10 or older, and 68% of those aged 10-24, report playing home video games [22].

Although the overall computer game playing hardware and software market in the US is relatively large, the US is a secondary audience for massively multiplayer online game's (MMOG's) compared to Asia. In fact, the current broad success of MMOG's in the Asian market is often cited as justification for predicting that the US MMOG market has significant possibility for growth. One factor that has hindered the growth of the US on-line game market is this country's lower rate of broadband internet, compared to Korea or Japan. [23]

The MMOG market is dominated by Asia. The two most popular Korean MMOG's, *Ragnarok On-line* and *Lineage*, claim somewhere between 2 and 6 million players each. The most popular US MMOG, *EverQuest*, has somewhat less than a half million players. *Final Fantasy*, a Japanese MMOG recently launched in the US, has approximately the same number of players worldwide. [24] Four of the largest five digital content companies are Japanese, while only one is American [25].

Fallacy 2: The computer game industry is larger than the movie industry.

Although total retail sales of commercial computer games in the US are comparable to total box office sales of movie tickets, total computer game revenue is still smaller than total movie industry revenue (which includes video and DVD sales and rentals). Similarly, the nearly $10 billion in computer games retail sales in 2001, while comparable to the sales of leisure or entertainment books, was less than the $26 billion total sales of published books or the nearly $14 billion in retail music sales [21].

It is forecast that computer game playing will grow faster than established entertainment industries such as movies, books, or music [21]. Mainly, this is because the age range of game players is expected to expand as the current generation of dedicated computer gamers ages.

Fallacy 3: Everyone plays computer games.

While a number of surveys have reported on America's video gaming tendencies, it is prudent to consider that many commonly quoted statistics have been generated by processes which have never been publicly examined [26]. It is likely that industry proponents will tend to overestimate the popularity of computer games by means such not differentiating between regular, occasional, and infrequent players.

Sales of console computer game units are somewhat easier to quantify. One estimate is that by 2005, 70 million US households will own a modern console unit

[27]. This is approximately 24% of total US households [28]. Another statistic is that 50% of Americans play computer games. This necessarily means that a large percent of Americans never or rarely play computer games. In particular, computer games are less popular with women and people over 35 [29].

There are several dangers in over-stating the relative size and importance of the US computer games market. First of all, a narrow focus on the US will miss the largest MMOG markets, hindering efforts to understand the social phenomena and leaning potential in these games. Secondly, over-selling the relevance and universality of computer games in this country can only result in future disappointment if planners rely on these fallacies to predict how fast the general US public might accept new game-based applications. It is important to consider the current demographics of computer game playing in predicting target audiences for other game-based applications.

Fallacy 4: MMOGs are authorable.

While *The Sims On-line* is widely recognized for player authored contributions to game content, these contributions are limited to the addition of somewhat superficial objects such as wallpaper or clothes. Although MMOG's such as *Second Life* and *There* are encouraging player defined content, these particular games focus more on socialization in a virtual world rather than typical narrative game scenarios or interactions with non-player characters. Thus, user generated content for these games does not include important aspects of game authorability such as scenario creation or non-player character behavior authoring. MMOG support of authorability that significantly affects character or game behaviors is currently minimal, though interest in user generated content is increasing [30].

Fallacy 5: MMOGs feature a persistent world.

Although MMOGs employ the concept of a persistent world, only character aspects such as personal possessions or abilities are entirely persistent. Character initiated changes to the game world are generally of short-duration. This limitation prevents the game world from drifting too far from planned scenarios.

Fallacy 6: MMOGs enable tens of thousands of players to interact in the same world.

The very "massiveness" of MMOG is deceptive. Large-scale MMOGs actually divide players into different game world shards so that in most cases less than a thousand players are really sharing the same game world. Again, there are currently initial efforts to remove this limitation.

Many proponents for broader application of game-based technologies have identified potentially useful characteristics in MMOG. At the same time, there are a number of limitations in MMOGs such as their support of game authorability, persistent in-game effects, and simultaneous player participation, which are not universally acknowledged. It is important to understand these limitations, since less limited implementations of these game characteristics may be desirable for broader applications. While MMOG authorability, persistence, and massiveness are potentially attractive for training or other military applications, planners must be aware that these characteristics are currently limited, and that the commercial game industry is just starting to push these limits.

Fallacy 7: The opponents in shooter games rely on complex AI for intelligent behavior.

Most commercial games rely on relatively simple AI and use "tricks" to make opponents who are appropriately difficult adversaries. In some games, the machine opponent has access to more game knowledge or better weapons than the human players. Other games use some random machine opponent behavior to keep the players guessing. Human players may perceive such random actions as being part of an intelligent plan.

Fallacy 8: Better AI will result in better games.

More complex AI that strives for human behavior validity can be counterproductive. Human behaviors, such as hesitation or misperception, when simulated by a game AI, can be perceived by real humans as just "broken" AI. Similarly, NPCs based on human behavior models can result in too much predictability.

There has been much interest in how game AI is applied to generating teammates, by-standers, or adversaries in the player experience. For military training purposes, such machine-generated characters could be used to make training more efficient and effective. Scrutiny of commercial game AI shows both current limitations and future promise for applications beyond entertainment. Commercial games demonstrate that it is possible for players to experience compelling game play with artificial opponents or teammates that are based on relatively simple AI. Thus, while it is a misperception that commercial game AI is particularly intelligent, it is true that players can perceive characters based on relatively simple AI as being compelling. A key lesson from the game industry is that perceived realism is more important than actual realism.

Fallacy 9: Commercial games are fast and easy to develop.

One potential benefit of using game-based technologies for non-entertainment applications is the possibility of leveraging existing games to produce new applications relatively quickly. Sometimes this optimism is based on a misconception that computer games can be cranked out by a lone programmer during a few marathon sessions.

While early games were sometimes the results of such efforts, modern games generally require a substantial development team. As games become more complex, development effort is likely to increase. The typical game development cycle now ranges from 18-24 months, with some games clocking 36-48 months of development time. It is difficult to shorten this time, even if a game is a sequel or reuses an existing game engine [31]. This should serve as a caution to planners that new game-based applications are also likely to require significant effort to develop. On the other hand, the very fact that development effort per game has been increasing means that there may be an increasing market motivation for game companies to develop reusable components.

Those interested in behavior modeling applications such as training in emulated environments will benefit from understanding the interactive entertainment industry and modern computer game technology. The commercial games industry has been able to offer immersive, compelling experiences to a growing percentage of people. While not a magic wand, creative use of existing and emerging game-based technology has the potential to improve training efficiency and effectiveness.

6. Appendix B: Resources

The following table includes a list of web sites about games, game AI, and non-entertainment uses of games as well as a list of books on the use of AI in games.

Web Sites
http://www.aiwisdom.com
http://www.digra.org
http://www.dodgamecommunity.com
http://www.educationarcade.org
http://www.gamasutra.com
http://www.gameai.com
http://www.gamedev.net
http://www.generation5.org
http://www.gdmag.com
http://www.seriousgames.org
Books
S. Rabin: “AI Game Programming Wisdom” Charles River Media. 2002
S. Rabin: “AI Game Programming Wisdom 2” Charles River Media. 2004.
J. D. Funge: “AI for Games and Animation: a Cognitive Modeling Approach” A K Peters, Ltd. 1999
M. Buckland: “AI Techniques for Game Programming” Premier Press 2002.
L. Dorfman & N. K. Ghosh: “Developing Games that Learn” Prentice Hall PTR 1996
M. DeLoura: “Game Programming Gems” Charles River Media 2000.
M. DeLoura: “Game Programming Gems 2” Charles River Media 2001.
D. Treglia: “Game Programming Gems 3” Charles River Media 2002.
A. Kirmse: “Game Programming Gems 4” Charles River Media 2004.

7. References

[1] S. Woodcock: “Game AI: The state of the industry” Game Developer, Vol. 7, pp. 24-32, 2000.

[2] S. Cass: “Mind Games” IEEE Spectrum. Vol. 39, pp. 28-33, 2002.

[3] C. Butcher & J. Griesemer: "The Illusion of Intelligence: The Integration of AI and Level Design in Halo" Presented at 2002 Game Developer's Conference, http://halo.bungie.org/misc/gdc.2002.haloai/talk.html

[4] B. Scott: “The Illusion of Intelligence” AI Game Programming Wisdom (ed. S. Rabin), pp. 16-20, Charles River Media, 2002.

[5] Unreal University: “EPIC GAMES ANNOUNCES UNREAL UNIVERSITY, NOVEMBER 8 & 9, 2003, AT NORTH CAROLINA STATE UNIVERSITY” http://www.unrealtournament.com/news/view.php?id=905

[6] J. E. Laird: “An exploration into Computer Games and Computer Generated Forces” 9th Conference on Computer Generated Forces and Behavioral Representation. May 2000.

[7] J. E. Laird & M. van Lent: “Human-level AI’s Killer Application: Interactive Computer Games” In Proceedings of the National Conference on Artificial Intelligence. AAAI Press, Menlo Park, Calif., 2000, pp.1171-1178, 2000.

[8] P. S. Rosenbloom, J. E. Laird, & A. Newell. “The Soar Papers” Cambridge, MA. The MIT Press, 1993.

[9] J. R. Anderson: “The rules of the mind” Mahwah, NJ: Erlbaum 1993.

[10] R. W. Hill, C. Han, & M. van Lent: “Applying perceptually driven cognitive mapping to virtual urban environments” AI Magazine, December, 2002.
[11] B. Best & C. Lebiere: “Spatial processing requirements for navigation and interaction with cognitive agents in a 3-D interior space” Symposium on Psychological Space at the 5th International Conference on Cognitive Modeling. 2003.
[12] K. D. Forbus, J. V. Mahoney, & K. Dill: “How Qualitative Spatial Reasoning Can Improve Strategy Game AIs” IEEE Intelligent Systems, 1, pp. 25-30, 2002.
[13] R. Quinlan: “Induction of Decision Trees” *Machine Learning*, *1*, pp. 81-106, 1986.
[14] M. Jona, et al.: “Button Theory: To taxonomy of to student-teacher communication for interface design in computer-based learning environments” Proceedings of the Thirteenth Cognitive Science Conference, Chicago, IL, pp. 765-769, 1991.
[15] R. R. McRae & P. T. Costa Jr.: “Updating Norma's "adequate taxonomy": Intelligence and personality dimensions in natural language and in questionnaires.” Journal of Personality and Social Psychology, Vol. 49, pp. 710-721, 1985.
[16] M. Lewis & J. Jacobson: “Game engines in scientific research” Communications of the ACM, 45, pp. 27-31. 2002.
[17] A. Khoo and R. Zubek: “Applying inexpensive AI techniques to computer games” IEEE Intelligent Systems, 17, pp. 48-53, 2002.
[18] G. A. Kaminka, M. M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. N. Marshall, A. Scholer, S. Tejada: “GameBots: a flexible test bed for multiagent team research” Communications of the ACM 45, pp. 43-45, 2002.
[19] J. E. Laird, M. Assanie, B. Bachelor, N. Benninghoff, S. Enam, B. Jones, A. Kerfoot, C. Lauver, B. Magerko, J. Sheiman, D. Stokes, S. Wallace: “A Test Bed for Developing Intelligent Synthetic Characters” AAAI Spring Symposium on AI and Interactive Entertainment, March 25-27, 2002.
[20] H. Kennedy. Computer games liven up military recruiting, training *National Defense*, November, 2002.
[21] M. Pachter & P. A. Saunders. Interactive Entertainment Industry Report, “Content is King: an in-depth look at interactive entertainment software”, Wedbush Morgan Securities, May 2002.
[22] Japan Statistics Bureau, http://www.stat.go.jp/english/index.htm
[23] http://boston.internet.com/news/print.php/1479321, summarizing *The State of Massive Multiplayer On-line Games 2002: A New World in Electronic Gaming* sponsored by Zona, http://www.game-research.com/statistics.asp quoting Net Profile Research, LG Ad 2000
[24] Bruce Woodcock, 2003 MMOG charts, http://pw1.netcom.com/~sirbruce/Subscriptions.html
[25] Bloomberg Data compiled by Square Enix
[26] http://www.game-research.com/art_what_women_want.asp
[27] Forrester Research: “Game Consoles Connect”, January 2002.
[28] 2000 US census http://www.census.gov/
[29] IDSA, ESA http://www.theesa.com/pressroom.html
[30] http://www.gamespy.com/amdmmog/week8/index.shtml
[31] http://www.gamasutra.com/features/20040109/fristrom_01.shtml

Author Notes

This material is based upon work supported by the Office of Naval Research (ONR) under Contract No. N00014-03-C-0161. We thank Dr. Harold Hawkins of ONR for his support and assistance. The opinions expressed here are the authors' and do not necessarily reflect the views of the Navy or the Department of Defense.

Author Biographies

DAVID E. DILLER is a Senior Scientist at BBN Technologies in Cambridge, MA. His current focus includes cognitive modeling, mixed-initiative agent based systems, and simulation-based training applications. Dr. Diller holds a Ph.D. in Cognitive Science and Psychology and a M.S. in Computer Science from Indiana University.

WILLIAM FERGUSON is a Senior Scientist at BBN and Principal Investigator for the ONR game behavior survey project. He currently also serves as technical lead for the DARWARS project -- a DARPA program to develop large, deployable simulation-based instructional systems using technology and ideas from the commercial gaming world. Mr. Ferguson has worked for over twenty years in industry and academia on Artificial Intelligence, story based instruction and intelligent tutoring environments.

ALICE M. LEUNG is a scientist in BBN's Distributed Systems and Logistics Department. She has been developing military inventory and medical materiel planning software using distributed agent technology. She has also investigated small-scale simulation of information transfer economics. Ms. Leung holds a

PhD in Polymer Science and Technology and a B.S. in Chemical Engineering from MIT.

BRETT BENYO is a Software Engineer at BBN Technologies in Cambridge, MA. His area of interest is in multi-agent systems with a focus on human to agent system interaction and adaptability.

DENNIS FOLEY is a Lead Technical Writer at BBN Technologies. For this project he has served primarily as an analyst of gaming technology. At BBN he is a technical-communication specialist, producing documentation and web sites, for example. He is a long-time wargamer and military-history buff.