

Games for Training: the Good, the Bad, and the Ugly*

Ralph E. Chatham¹

The notion to use computer games for military training is almost irresistible. Unfortunately, unstated, unexamined assumptions about learning and the character of and driving forces behind commercial games can lead the way to unpleasant results if we don't keep the goal of proficiency "gains" ahead of blindly use of "games." I discuss a handful of lessons learned while developing and deploying two new training systems within DARPA's DARWARS program. Perhaps they will constrain the dark side of computer-based training from tainting other, training-based, efforts to deliver a wide range experiential learning delivered by light-weight simulations.

Military Training is Hard

Our military must train continually as personnel rotate and technology and missions change. Every person, every unit must not only refresh what they previously learned, but also absorb new lessons and techniques, many of which are exactly what our Soldiers and Marines did not sign up to do.

The situation is not hopeless. In the 1970s and '80s the Services, especially the Army, created what the Defense Science Board called a Revolution in Training² delivering an order of magnitude change to unit proficiency in less than three weeks. A unit as large as a Brigade, about 4,000 Soldiers, deploys to one of three fixed-site combat training centers (CTCs) where they: engage in battles with a better-than-real enemy, measure the engagements objectively, and then conduct no-holds-barred after action reviews (AARs) where everyone from the privates to the Colonels discuss what happened and what could be done better. It is effective, but expensive, incurring costs for travel, logistics, instrumentation, and maintaining the better-than-real opposing force (OPFOR).

Over two plus decades, this has changed Army culture. Nothing is done now without AARs up, down and across the chain of command. Lieutenants are pleased to hear and to act upon the advice of their sergeants and corporals. Army trainers view every training procedure as a route to an AAR; the faster you get there, the better. Unfortunately, large units visit a CTC only once every three years. Many units never do.

Can Computer Games Help?

The DSB recommended bottling this experiential training revolution and deploying it electronically for use by more people more of the time. This led the Defense Advanced Research Projects Agency (DARPA) to create DARWARS³. Early in the program, after restricting it only to PC-like hardware, the fallacious notion arose *by spontaneous generation* that DARWARS was a "games for training" program.

There are cogent reasons for this almost irrepressible bias towards thinking of games, from massive multi-player online ones down to first person shooters, as instant training devices. After all, they superbly teach people to play games. Shouldn't they, therefore, be good at training other

* This document was approved for public release, distribution unlimited by DARPA on 9/6/2006. A modified version was published in *Communications of the ACM July 2007* p37-43. An expansion of the subject will be published as two chapters in: *Development of Professional Expertise: Toward Measurement of Expert Performance and Design of Optimal Learning Environments*. K. Anders Ericsson, ed. Cambridge U. Press, probably early 2009.

things? The emerging answer is that this won't happen spontaneously. Mangling a metaphor: "a game unexamined yields an empty trainer."

Myths About Games for Training

I, too, entered this realm with unstated, unexamined assumptions. Training games, I expected, would automatically be:

1. CHEAP (to create, deploy, and maintain)
2. FAST (instant development and delivery)
3. EFFECTIVE (training automatically transfers to the real world)
4. TRAINER-LESS (give the user a disk and s/he would learn unsupervised)
5. UNIVERSAL (automatically accessible to anybody with a PC)

I was wrong. Unless we insist on proficiency "gains," not "games," these assumptions will prove false. DARWARS experience suggests, however, that, tempered with a dose of reality (and adequate funding), a middle course can be steered between the game fanatics and the developers of massive, expensive special purpose training systems. We now have solid evidence that lightweight training simulations at their best can be superbly effective. Unfortunately, "at the median they are awful."⁴ Thus we are not out of the woods; there is serious danger of drawing wrong lessons from the successes unless we document what works, why, what outside the game is needed, and whether learning transfers to the real world. So here are a few DARWARS lessons, the good, the bad, and the ugly that we learned adapting game and game-based technology to train.⁵

Tactical Language & Culture Training Systems (TLCTS)

TLCTS is not a game, but its core technologies come from games, as well as AI and learning theory. The project arose when I realized that the military could never create a massive corps of linguists. So I asked, instead, if we could put just a little gesture, culture and mission-specific vocabulary into the brain behind *every* trigger finger and steering wheel. Given other training demands, we couldn't expect much time with the student. Thus I charged the performer to define "tactical language," what militarily valuable knowledge could we stuff into foreign-language-impaired brains (like mine), and do the stuffing in two discontinuous weeks. The resulting Tactical Iraqi Language Training System comes on a single CD, free for government use, containing ~100 hours of training. Levantine Arabic and Pashto are also finished; French is coming soon.

Like Gaul, Tactical Iraqi is divided into three parts. A vocabulary and culture tutor listens to students' utterances and applies speech-recognition technology to assess progress. Students may also navigate through a set of arcade games by speaking Arabic commands: directions (left,



Fig. 1. Three parts of Tactical Language Training: a Skill Builder, an arcade game, and a 3D environment.

right, north, south, toward the river, *etc.*), place names, color names and military ranks. The more complicated the utterance, the more points. In listening modes the trainee responds to computer-delivered Arabic commands.

The University of Southern California Center for Advanced Research in Technology for Education, CARTE's answer to the "2-week" challenge was game technology. The game engine initially only drove the third part of the trainer, a Mission Environment where students talk their way in Iraqi Arabic through encounters with multiple characters and sticky situations in the performance of non-shooting military tasks. (In early versions a left mouse click yielded the pop-up: "you can't shoot; you must talk your way out.")

The decision to use an *existing* game engine was driven by cost and successful use of the Unreal engine in the *America's Army* recruiting game. We could get prototypes up and running quickly to learn if we *could* teach *some* language to everyone. It also kept the program alive by enabling us to show-off intermediate products to users, funders, and the media.

We found that shifting among training modes took too much time. This leads to "games for training lesson one: ***the game wants the whole computer,***" leaving no resources for speech recognition or videos of a talking head. CARTE traded some of these desirable features for rapid mode shifting. Now the game-engine acts like an operating system for everything.

Some game engine tools were quite useful; when students found it hard to recognize changing attitudes of virtual characters in the mission (game) environment, CARTE converted the "ammunition status bar" into a "trust meter." They also used *America's Army* resources to get started, although later dropped them because of copyright restrictions.

Other features were not useful. The programmer building the arcade games found that once one's avatar reached a new point in the Pac-Man-like maze it assumed a random orientation, rather than maintaining the direction of last motion. The documented routine that should have solved this didn't. Weeks of poking into the dusty, disused corridors of the Unreal engine eventually led to a work-around. The ***game's broad user community helped*** here.

A second problem was subtler. CARTE found that their new Pashto speech recognizer was not working well. Looking back at a new implementation of Tactical Iraqi, instituted when shifting from tools permitted by a research license to ones conforming to a newly purchased commercial license, they found that Arabic recognition had degraded, too. Months of searching traced this to an undocumented 8 kHz low-pass acoustic filter. It worked well for game communications; for a speech recognizing it was a disaster.

Thus: ***factors driving game engine development may not coincide with those driving a training application.*** Game developers stop fixing things when the *game* works. This leads to dangling, undocumented features appearing when one uses the engine for something other than playing the tested game.

Next: ***licensing issues.*** When you move from a low-cost research license to distribute a new training product for real, the game developer wants, justifiably, a cut of the revenue. The Tactical Language license alone cost more than does a typical whole schoolhouse training curriculum development. Games-for-training hopefuls balk at paying several hundred thousand dollars just for permission to distribute a game-based trainer. The Tactical Language license, however, was a bargain. It gives the government rights to use the engine in current and future tactical language

tutors, only requiring that the packaging proclaim “Epic technology inside,” and that Epic receives a percentage of new development costs.

Game from the Start: *DARWARS Ambush!*

Responding in 2003 to a DARPA Director’s challenge, I started a program to train the voice in the back of the head of every service person how to identify, prepare for, deal with, and recover from ambushes. I chose a development team that held training value as their first priority.

Dubbing the program *DARWARS Ambush!* I then set them free constrained only by these goals:

1. Be guided by a respected early adopter. Col. Robert Brown the commander of the 1st Brigade, 25th Infantry enthusiastically adopted us and immediately marched my contractors^{6,7} into the mud during his CTC rotation.
2. Build for the soldiers, who bear the brunt of the ambush, not the Lieutenant Colonels.
3. Provide for simple, rapid field authoring. Teach what happened yesterday, not waiting for months for contractor changes with attendant cost and confusion.
4. Make OPFOR be the squad next door, not software. They would learn as much executing an ambush as being ambushed. Inter-squad competition might insure the tool would be used continually, rather than played once and put on the shelf.
5. Use games to make training compelling (but put training inside)
6. Deliver in 6 months.

BBN’s team exceeded those goals, creating a networked, multi-user, PC-game-based, convoy-ambush training tool that “allows Soldiers and Marines to experience lessons learned by others and to construct their own scenarios based upon actual experiences. Trainees move about in a shared, immersive, first-person perspective environment where they carry out mounted and



Figure 2. Soldiers at Fort Lewis training with DARWARS Ambush! summer 2004 (Jason Kaye of the Northwest Guardian, used with permission)



Figure 3. Screen-shot from DARWARS Ambush!

dismounted operations, operate ground and air vehicles, use small arms and vehicle-mounted weapons, and communicate over multiple radio nets.”⁸

This approach limited the training developer’s opportunities to generate continuing revenue.

The game engine, *Operation Flashpoint*⁹, costs users about \$10 per seat, and the six-month deployment goal tacitly assumed that the government-developed product would be free for users

to copy and distribute. I had envisioned that BBN would do the rapid authoring in response to field requests. However, they took my vision one step further by embedding training to show users how to author scenarios themselves. In consequence, Soldiers make all but the most complicated changes – with no cost but their own labor. Similarly, homegrown OPFOR can adapt to any new roles without developer expenses.

DARPA, and others, gave BBN occasional time-and-materials contracts to add features, scenarios, or terrain. The major continuing business provides web and phone, as well as on-site train-the-trainer support, but this, too, doesn't represent an enticing source of continuing revenue.

I don't recommend, however, that training developers sell game-based training on a per-seat basis, maintaining control over changes and charging for upgrades. It irks users when they can't make even minor alterations. Moreover, there are too few users to keep the per seat costs low enough to fit within unit budgets. What they *can* afford is their own labor.

Probably the best hope for a good product, satisfied users, and modest developer profit is what DARWARS did: provide up front money from a central source. Then pay the developer, or some other entity, on a time and material basis, to deliver user support, set-up and train-the-trainer services in chunks small enough to fit within a unit's budget.

WHY A GAME

The “deliver in six months” requirement demanded an existing simulation engine. We chose Operation Flashpoint as a clear winner out of eight or ten alternatives because it:

1. could handle large-scale outdoor terrain
2. had tools adaptable to field authoring
3. allowed users to enter and operate vehicles
4. delivered adequate realism
5. had flexible multi-player capability

Flashpoint was also a good choice because, being several years old, it didn't require a top-of-the-line PC with the latest video and memory cards. It had been well tested in similar applications by a gaming community that had also created a host of simulation resources and objects available for the cost of acknowledging their creators.

Flashpoint was also a bad choice for reasons we recognized from the start: limited support for AAR and voice communications. These arose because our ***training developers had no source-code access***. Also, late in the process, a lengthy negotiation over licensing requirements erupted that took time and energy away from distribution and support activities. Part of the problem came from a mismatch between the business model for making money with a commercial game, which risks much money early in hopes of making it back on sales, vs. government acquisition practice where profit comes during a paid development period.

The licensing issues were finally resolved with an agreement linking licensing fees to actual development costs. The legal wrangling was brightened when BBN let their attorneys play, and the technical staff got to shoot the lawyers.

DISTRIBUTION BY DIFFUSION

DARPA sits outside normal military acquisition channels, so while our early adopter quickly embraced *Ambush!*, we had no mechanism for further distribution, or even to deliver the message that it existed. Tactical Language was in the same boat. So we engaged the press, visited units with demonstrations and conducted word-of-mouth campaigns. Knowledge and demand for both products spread rapidly and for the next six months I received at least one message daily asking how to get the products. Mendicants received software in the next day's mail. This would not have worked had we charged the units for the government-developed tools.

Today sites at Army and Marine bases across the country and overseas have upwards of 100 computers dedicated to training with *DARWARS Ambush!* Over 20,000 soldiers, Marines, and airmen trained with these tools in 2006. Had we tried to introduce the products from the top, we would still be awaiting approval. **Bureaucracy is slow, but a government charge card in the hands of a Major is fast.**

The services have begun to pay for user support and central maintenance of the software. This is timely, since DARPA, as the Attention Deficient Disorder child of defense research, abandons programs once we have taken the technical excuse from those who say something cannot be done. Service organizations are justifiably resistant to adopting the support burden of something pushed on them from the outside. Unplanned costs must come out of an already tight budget. Thus distribution by diffusion can only succeed with strong user demand, and demonstrable training performance.

WHY DARWARS AMBUSH! SUCCEEDED.

First, it met an urgent, clearly perceived need. Once users tried it, they discovered that *training was built into the product*. It was not just an environment in which training could take place, the disk included: training manuals, tactically representative scenarios with descriptions of how to use them for training, and train-the-trainer information. Military billets are often gapped¹⁰, so we cannot expect one user's enthusiasm to be transferred to the next. At one training facility the new director found a high stack of CDs of game-like things all claiming to train. Only the *DARWARS Ambush!* disk told him *how* to train with it.

When distribution requires user acceptance, the tool must appeal to the end customer. What they adopt with their own money and effort must work well. Our gaming contractors insisted that we *test intensively before release* far beyond normal military training software testing. Commercial testing practice weeded out bugs that would have prejudiced users against us.

User authoring: we didn't deliver an application that lost user interest once the students had run through the 24 provided scenarios. Instead, we gave ordinary users tools to change scenarios themselves and create new ones to meet changing circumstances. Lessons from deployed units are incorporated into training within days. Commander-identified training needs are met with user modifications. One unit created the terrain of their home base then used *Ambush!* for a disaster relief exercise. They simulated tornado damage with a platoon of virtual tanks, knocking down trees, flattening cars and damaging buildings. Others used *DARWARS Ambush!* to produce a training video of a real battle, which the commander then used to prompt discussion of his tactical intent and explore tactical options with his battalion.

This flexibility for users to invent and implement new ways to use the tool without a contractor between them and their tactics, techniques and procedures was the most important lesson of the

project. When users can do this with no delay and no costs but their own labor, they embrace the trainer as their own, not resenting it as imposed from above. They never asked us if it trained well; they saw its value in their own AARs.

Finally, *we listened the users* and engaged them *early and often* in the development cycle. We took draft systems on the road every month to discover what needed fixing. But, because we listened, we found the alpha version trained units about convoy ambushes even as they showed us how to improve. Game platform use allowed us to get vital feedback early.

THERE IS NO GOLDEN DISK

Both *DARWARS Ambush!* and the Tactical Language tutors showed us that *there isn't yet a trainer-less trainer, but they do dramatically reduce the need for human trainers*. The mere acquisition of a disk of training software seldom results in effective training. *Ambush!* reduces quickly to a free-for-all unless it is used in a setting with an instructor, training goals, and enforced AARs. Tactical Language and Culture tools, while usable by motivated individuals, delivered much more useful training in a group setting. One pair of cooperating novice students taking turns on the same computer got more out of it than did most students with similar motivation who had a computer to themselves. Human instructors also insure that artificialities of the simulation do not lead to negative training.^{11,12}

Reviewing the lessons

I'd end this essay with a bang, not a whimper, but the word-count gods must be propitiated. So you only get a table of lessons, ... and this: thanks to Dr. Lewis Johnson of CARTE and Mr. Bruce Roberts of BBN who each created from vague ideas real training that is saving lives today. (2997, 2998, 2999, whew.)

<p align="center">Table 1. A Few Games-for-Training Lessons. <i>(Reader beware: there is vague correspondence of thought along a few rows, but not all)</i></p>		
<i>The Good:</i>	<i>The Bad:</i>	<i>The Ugly</i>
User authoring drives user acceptance.	Many things, like human behavior, are hard to author.	There is no golden disk, no “trainer-less trainer” that will compel trainees to use it by themselves. Humans must still be around to ensure effective training.
Existing game engines permit early prototyping.	Game engines are only tested for gaming. Use them for something else and beware of undocumented features.	Often you can’t get access to the source code. America’s Army resources were later dropped to avoid copyright problems.
Games with broad user communities can provide resources and technical solutions.	Use of some resources, <i>or even the game engine itself</i> may be forbidden by network security concerns. DARWARS bypassed this by using only detached networks, but that approach is not always available.	Government contracting and commercial game business models clash. It is hard to find a business model to give training developers incentive to support light-weight simulations after they are built, especially if user authoring is, as it should be, required.
Older games provide good-enough simulation for training on low-end military computers.	Game reality usually comes from smoke and mirrors.	Training must still be built into the product. You must listen to users early and often.
Broad licenses of older games can be negotiated to enable wide distribution at minimal cost to the end user. Amortized over all users this is a bargain.	Finding cash up front for the license deters developers with small budgets. Moreover, a single commercial game can cost \$4M to \$40M. Good training, based upon good content does not come cheaply, either.	Up-front money notwithstanding, licenses are often worth it, but negotiating a license is often a pain.
Games can provide a stable, tested simulation engine.	Game engines don’t want to share (the computer).	Distribution by diffusion demands commercial game testing practices, <i>before initial delivery</i> .
Light weight simulations can reduce the need for human trainers.	There is not yet a trainer-less trainer	Comments to the left are either good or bad news depending upon which half of the glass you focus on.

¹ Co-Chairman, Defense Science Board task forces on *Training Superiority and Training Surprise* and *Training for Future Conflicts*. A former submarine officer, now Program Manager for Training Superiority, Defense Advanced Research Projects Agency, Dr. Chatham is also a Ph.D. physicist. However, not making house calls, he sees no need to advertise this fact outside of footnotes.

² Training Superiority and Training Surprise 2001 and Training for Future Conflicts 2003. Reports of the Defense Science Board. Available at www.acq.osd.mil/dsb/reports.htm

³ DARWARS = DARpa’s universal persistent, on demand training WARS

⁴ This is an anecdotal assertion. Still, cast your mind back to the last web-based company-wide training you took and ask if it does not seem justified. Thanks to Bill Ferguson for the observation.

⁵ There was much more to the DARWARS Program than discussed here. See, for example: a summary of my speech to the 2005 DARPATECH Symposium at <http://www.darpa.mil/darpatech2005/presentations/dso/chatham.pdf>. There are more lessons, but the word-count police restrained me.

⁶ BBN Technologies, Total Immersion Software, and Jason Robar, Inc.

⁷ A great deal of the credit for the success is also due to the *pro bono* contributions of Mr. Dan Kaufman, now Program Manager at DARPA. His new goal is to create user authoring tools that empower the Soldier to create his or her own *DARWARS Ambush!*-like mission rehearsal tool, but in a geo-specific world, and do it in a few days. When successful, his program, *RealWorld*, will put *DARWARS Ambush!* out of business.

⁸ Roberts, Bruce; Diller, David; Schmitt, David, "Factors Affecting the Adoption of a Training Game." *Proceedings of the 2006 Interservice/Industry Training, Simulation and Education Conference*.

⁹ Operation Flashpoint™ is published by Codemasters and was developed by Bohemia Interactive Studios.

¹⁰ When a billet is "gapped," one person leaves, often months, before his/her relief arrives.

¹¹ When challenged with this potential for negative transfer of training at the first *DARWARS Ambush!* Users Conference in February 2006, several users of *DARWARS Ambush!* strongly countered that it only would happen if the trainers failed their students in the AAR. I didn't have to say a word.

¹² In fact, elsewhere in DARWARS, we have seen strong evidence of positive transfer of mental skills learned in computer simulation to combat-like situations. See Widerhold, Brenda; Widerhold, Mark; Physiological Monitoring During Simulation Training and Testing, Final Report, Virtual Reality Medical Center report on contract DAAH01-03-C-R-301, July 29, 2005