

Simulation Without Programmers

Daniel Kaufman

Program Manager, Defense Sciences Office



DARPA 25th Systems and Technology Symposium

August 7, 2007

Teleprompter script

“ Good morning, Most program managers come to DARPA from careers in academia, the military, or the Defense Industry. My story is very different. I came from a corporate background in Silicon Valley. I closed about \$1.5B in venture capital deals, and then I worked for Microsoft and DreamWorks before I came to DARPA. So I come from a culture that looks at the world *quite differently* than most DARPA program managers do.

Let me illustrate. Last March I went to a party and met up with a bunch of my old VC and Microsoft buddies. The conversation got around to this conference, this speech, and what I was going to say to all of you. I told my corporate friends that I am first going to talk about some of DSO's accomplishments. They all nodded, and then I told them I am going to talk about what we're currently working on...this new program of mine called RealWorld that's going to completely change the way we build training software. And then I told them that I will discuss the problems we're facing, and the problems I think we're *going* to face in the future. One of my friends stopped me and said, in complete disbelief, "You're doing that *in public*? *Is your boss going to be there?*"

I said, "Yeah, sure...he knows. In fact, at the end, I'm going to tell the DARPATech audience that we have money available for anyone with a really great idea on how to help solve any of these problems."

That's when another friend looked at me, put his hand on my shoulder, and said, "Dude, you are so gonna be fired!"

And that, I think, really sums up the way DARPA views the world—and what a truly unique opportunity DARPA and DARPATech are for someone with my particular corporate background. For me, it's all about revolutionizing the business of military training and in-theater simulation software for our warfighters. I always laugh about my friend's remark about being fired.

You know what? I think that if we're successful in the DARPA programs I'm going to tell you about today, the entire business of software development will change. Because *the whole point* of the RealWorld program is to build simulations *without programmers!* I'm going to leave you with that teaser for a moment.

Let's go back to where we were two years ago. At the 2005 DARPATech, my predecessor stood up here and talked about his vision for what training was, and what software was going to be. And he made three statements, which I think were extremely powerful:

1. The amount of material that we're asking our warfighters to master is rising exponentially.
2. We can neither lessen the training burden, nor increase available hours.
3. We could take the existing commercial software – particularly game technology – and use it to rapidly build viable training tools cheaper and faster than what is possible today with current DoD software modeling techniques.

So two years ago we left you with those thoughts. And a lot of people stepped forward and helped. One of the programs we built was *DARWARS Ambush!*, a PC-based trainer that teaches Soldiers how to recognize, manage and recover from convoy attacks. *Ambush!* has been used by over 30,000 warfighters to train in both convoy operations and leadership training. So maybe you are thinking, “Wow. These sound like huge successes. DARPA should just declare victory and go home.”

That was a thought...

But we actually have become a victim of our own success. Having demonstrated that we *could* use existing commercial software to build training tools faster and cheaper, we opened the floodgates. And that's when we ran into a couple of key problems with our approach to these tools.

The first thing you find out is that there is an enormous need: our warfighters need far more training and simulation tools than we could possibly ever build. They need them for convoys, avionics, signal intelligence, force protection, disaster planning, medical care and the list goes on...There's just not enough time and not enough money.

The second thing you learn is that we don't build them quickly enough. In fact, one of my Air Force buddies said that “Most simulations I have are monuments to the way the aircraft *used to* be.” So this leads us to an interesting question:

If I can't build enough, and I can't build them fast enough, what am I going to do? I can't continue to do things the way we are!

Before I wanted to get going on this problem, I decided to do something I learned from DARPA—and from Dr. Tether in particular: I decided to go to my customer and say, “What is the tool you would really like? Not what you think I can solve or build, but what do you really like? How do I actually solve your problem?”

Wow: Another culturally shocking departure from my corporate roots!

I still remember the old software adage, “Beware asking the customer what they want, *lest they actually tell you!*” Well, I *did* go to see my customers – Special Forces, the Army, the Marines, the Air Force, and the Navy. I spoke with anyone and everyone who would talk with me – troops

just back from battle, the modeling and simulation folks in all the various labs, to acquisition professionals. I asked them this question, and their eyes lit up. And you know what? They all told me the same things:

- Build me a training and simulation tool that helps me do my job better.
- I want *one tool* that does training, planning, analysis and mission rehearsal.
- It needs to be specifically geared to the current situation *I'm* facing – not some generic, textbook problem in some generic, textbook location. We fight wars in Afghanistan, not “Afghanistania.”
- It needs to adapt to fit my needs as my needs change: from kinetic to non-kinetic; from rural to urban; from small conflicts to larger conflicts; from individual soldier to commander; from Iraq to wherever.
- Oh yes: I want it tomorrow.
- And I don't want to spend more than an hour to learn the system.
- Oh, oh...and one more thing: It must be deployable, preferably on a laptop and be usable by every warfighter, independent of job or rank.
- And no license fees or royalties.
- And please do not make it a proprietary system.

I went back and started thinking and talking with some of my old software colleagues. And they told me what you are probably all thinking: This is really, really hard...Maybe impossible.

Well, I'm unwilling to tell the young men and women who go into harm's way to defend this country—19-year-old kids who risk their lives on a daily basis – that a bunch of software geeks, like me, whose greatest daily danger is carpal tunnel syndrome, think that the warfighter's problem is just too hard for us to solve. A primary mission of DARPA is to actually help solve the *really hard* problems. It was obvious to us that we cannot answer our customer's needs if we continued to work the way we do now.

How do we cause a fundamental paradigm shift? How do we make things go faster? And the answer that suddenly came to us:

“Simulation without programmers.”

Just think about that for a minute: how could I get *the warfighter* to *build his own simulations*. If he could do that, there are a lot more warfighters than simulation people, and they know exactly what they want. And they can build stuff a lot faster... so they could meet their own needs!

The warfighter comes first.

So what I needed to do is to build a bunch of tools that empower the warfighter, rather than the software developer. It means that the goal is not building powerful software. It's building software that empowers. The goal is to make the warfighter's life better – and if that requires building powerful software, so be it.

But we will be judged by the warfighter's effectiveness, not by the software features list. This, in turn, dictates a new approach to getting software requirements. The current waterfall development approach simply does not work. The 20 meetings to write the 100-page RFP to generate the 1000-page specification to find a product that will not be delivered for four years has consistently failed our military—and will continue to fail. You cannot ask someone to define every feature they will need in four years, when no one can accurately predict what his job will be in four years. Most of us can't tell you what we'll have for lunch next week! Instead, we need to build tools and capabilities so that *the warfighters create the applications they need when they need them.*

This is at the heart of my program, RealWorld, which you can see in the DSO booth: "Simulation without programmers." And this is where I'm going to need your help—to make RealWorld a Real-ity. The best way to tell you about the kinds of problems we have to solve is to tell you about the program itself.

So what is RealWorld? There are two simple ways to think about it. First: as Word is to a document, or as Powerpoint is to slides—I want RealWorld to be to building simulations. But that's a software-geek's view of it. Instead, let's look at it another way, since I promised you we put the warfighter first. Let's take our 19-year-old warfighter. He's out on patrol in a rocky canyon in Afghanistan and some OPFOR pops up and shoots at him because that's what an opposing force does. Our warfighter engages, the OPFOR vanishes, and our Soldier returns to base to be debriefed by his commanding officer. Our soldier gets out his laptop – and, *voila!* On the screen appears a scene that is an exact 3-D recreation of precisely where he was in that canyon. Not generic terrain – this is exactly his patrol and exactly his location. Within seconds, our soldier is dragging-and-dropping:

"This is where I was; this is where my buddy was; this is where George was; this is where the HMMWV was, this is where the sniper was, and this is where we got shot, sir."

Notice that I said *he* does it. There's no software guy; there's no writing down specs. *He* does it, and within seconds it's right on his laptop screen and it's exactly correct. If you think about it, in that one small instance, four very important things have taken place: RealWorld has become an after-action review tool, a mission planning and briefing tool, a mission rehearsal tool and a training tool.

Imagine recording this whole sequence, and then sending it back by e-mail to Ft. Polk and Ft. Lewis, and Twentynine Palms, or anywhere else. And instead of trying to tell a kid back

at a U.S. training base, “Look—here’s a 100 pages of doctrine that explains how you are supposed to handle an IED, and here’s a Powerpoint slide, and here’s a satellite map, and here’s a contour map,” *we put him right there!* Now training takes on a whole new meaning.

Our stateside Soldier is not working with, “Here’s a square: imagine that’s you, and imagine the bad guy is this circle over here.” We’re saying, “In 90 days, you’re going to be there. Work with this simulation and figure out what you would do. Because if we have not gotten that sniper—who really does exist—in three months, odds are he will still be out there and it will be *your* job to go get him.” To make all this happen, we have lots of very hard problems to solve in RealWorld. We need your help. *How* can you help? Let me give you a few ideas:

I need to be able to ingest DTED and DEM, photos and LIDAR, and CAD files in order to generate 3-D buildings and terrain. Quickly and accurately, and easy enough to use in the field. Can you help me do that? Do not give me solutions that I need a programmer for. And this is only what we have available right now. Our vision for the future includes access to future data sources, including making a market for those sources. If you have solutions, come see me. But let’s up the ante a little more. OK, a lot more! Let’s talk about the vision of where we want to go with this.

Once we can quickly, easily, and accurately replicate our geo-specific space in RealWorld, it will become critically important to be able to quickly, easily, and accurately model the cultural-specific human space. We need to introduce *human factors*. I want crowds of people behaving as they would in the real world. I want them to have multiple, conflicting goals. I want a promising individual, whom I’m approaching in the simulation, to think, “I’d like to help this soldier, but I want my family to stay safe. I can’t be seen talking to him.”

Imagine this scene: I have a shepherd and his flock. And you’re driving this convoy and he just crossed in front of you. Imagine that, instead of just driving around him, this time you could get out and talk to him in a culturally appropriate manner. You take off your sunglasses, you say the right things. And you finally get around to asking, “How’s business?” He says, “I’m moving my sheep. Some of these fields are dangerous because of left-over mines.” What if I were then to go help clear that field? How does that change dynamics with his village? How does all that play out? How do I start to learn the cultural space and share insights? But be very careful! There’s a real hard, nasty DARPA question lurking in here.

When we replicate the physical world, it’s pretty straightforward to verify our accuracy: either the building looks right or it doesn’t; it blows up correctly, or it doesn’t blow up correctly; the bomb does ‘this’ amount of damage, or it doesn’t. We could iterate and iterate, and, eventually, people would check it out and say, “That’s close enough – it works; looks good.” But in the space of human factors, how do we know what models to use? How do we know people are acting the right way? What are our standards, our metrics for accuracy?

So, in the true spirit of DARPATECH, I want to leave you with a challenge: craft me an experiment where we can take a number of social models and theories, implement them in

software, and test them against known datasets so we can see how valid these models are. Do this, and we will revolutionize the social sciences. This is the path we want to pursue with RealWorld. If you have ideas come by and see me, I'll be at the DSO booth!

Now it gives me great pleasure to introduce my good friend and colleague Dr. Michael Callahan.