DRDC-RDDC-2014-P41

Stayin' Alive: What are Persistent Synthetic Environments?

Mr. Brad Dillman
VT MÄK
1012 Northgraves Crescent
Kanata, Ontario Canada K2M 0C9
+1 613-883-5156
bdillman@mak.com

Mr. Mark G. Hazen
DRDC Atlantic Research Centre
9 Grove Street
Dartmouth, N.S. Canada B2Y 3Z7
+1 902-426-3100 x176
mark.hazen@drdc-rddc.gc.ca

Mr. Bobby Chawla
Real-Time Engineering & Simulation Inc.
19 Emerald Meadows Drive
Kanata, Ontario Canada K2M 2L3
+1 613-963-0072
bobby.chawla@rteng.pro

Keywords: persistence, virtual worlds

ABSTRACT*:* *The concept of a persistent synthetic environment has existed for years and has gained credibility in the military M&S environment as serious games & massive multiplayer online games (MMOGs). Typical motivations include cost savings through re-use and reduced lead time to availability of simulations. The ideas of persistent synthetic environments (PSEs) are identified and defined. A PSE could exist in several layers of cloud services for users to connect to persistent infrastructure, models, data and storylines. Conceptual models are developed for various kinds of PSEs and their application to simulation problems, along with discussion of advantages, limitations, opportunities and alternatives. The life cycle of simulation infrastructure, models, data, scenario and simulated entities are discussed in the context of their persistence and requirements.*

1. Background

A common high-level requirement that shows up in project documents or briefs [1] (for example the Canadian Advanced Synthetic Environment (CASE) [2] [3] and the US Joint Distributed Continuous Experimentation Environment (DCEE) [4]) is for persistence. These requirements appear to be motivated from experience with commercial virtual worlds (vWorlds) and Massive Multiplayer Online Games (MMORPGs) or Online Role-Playing Games. In these commercial products, the requirements for users to enter into the experience with minimal overhead and quickly become immersed, make the games extremely popular. In the Modelling and Simulation world, and especially in the training community, the idea of having a system that students both can and want to use, with minimal overhead, seems extremely attractive.

Of these two technologies, vWorlds is the more general, while MMORPGs can be seen as a sub-set instantiated for a specific purpose. Bell [5] has proposed that a vWorld is a "synchronous, persistent network of people, represented as avatars, facilitated by networked computers." In this definition the notion of synchronicity includes concepts of location, space etc., the "world", and differentiates from collaborative work areas such as Wikipedia. The most well known example of a virtual world is Second Life [6], developed by Linden Lab and launched in 2003. Second Life is largely built and operated by its users and has a wide variety of environments. MMORPGs [7], on the other hand, have typically been developed as game systems and the environment is often non-modifiable by users other than through pre-defined avatar actions. One of the most well known examples is World of Warcraft developed by Blizzard Entertainment. Thus, MMORPGs are examples of vWorlds developed and constrained to a specific set of purposes, while a more general vWorld like Second Life imposes far fewer rules upon users.

However, from a systems engineering perspective this high-level requirement is problematic. As with

M&S terminology of ‘simulation’, ‘model’ or ‘terrain’, the word ‘persistent’ could have many interpretations. The requirement is often neither well defined nor the elements to implement them readily available. The aim of this paper is to explore the concept of persistence and to propose some initial system engineering tools for understanding the actual desired requirements for a synthetic environment.

2. Introduction

From the Oxford Dictionary the definition of persistent is: Continuing to exist or occur over a prolonged period or Continuing firmly or obstinately in an opinion or course of action in spite of difficulty or opposition [8]. This paper will use the former definition, although, there are certain elements of the latter definition seen in some proponents.

Moving from this definition a Persistent Synthetic Environment (PSE) would be a synthetic environment which has some aspect that has an ongoing existence over a prolonged period. An “environment”, for the purposes of this paper would include Core products (Elevation Vector Feature and Material data), derived products (Imagery, Maps) [9], climate (air temperature, winds, pressure, visibility, bathymetry, sea bottom types), entities, entity interactions, and consequences [10] of interactions. Using this definition, the question then becomes one of, what aspects of a synthetic environment need to be ongoing, and for what types of periods.

These requirements have some resemblance to requirements for “re-use” of simulation components and data. The primary driver for “re-use” has been the reduction in the cost to develop and operate synthetic environments. Coupled to this concept has been the development of standards for the interchange of components and data in order to broaden the market for “re-use”. Up until about 2005 there were many papers on reuse of simulation components; the decline in numbers of papers in the past decade is an indication of the acceptance of the concept [10]. It seems clear that “re-use” of simulation components and data can only occur if they have some persistence and availability.

However, simple re-use of components and data seems to fall short of the concept of PSE that comes from user experiences with vWorlds.

3. Characteristics of PSE

One attractive benefit of a vWorld is its well known yet almost unbounded conceptual model. In the absence of definition, the environment and entities in a vWorld are expected to behave like the real world. For example, the vWorld environment can be setup to consider real-world events, weather patterns, and scheduled flights.

The vWorld conceptual model can also extend to the passage of time itself. Whether users are actively engaged in the simulation or not, they expect simulated time will pass and the vWorld will exhibit dynamic changes not directly caused by users.

Examining the vWorld user experience there are a number of characteristics that appear to be common and not explicitly related to simple “reuse”.

1 Re-entrant – allow for entry and exit of participants/users

2 On-demand – available when desired with negligible set up effort and delay.

3 Asynchronous usage – not all users have to be in world at the same time.

4 Multi-user effects – in-world effects/consequences from one user that affect other users.

5 Sustained effects – in world effects/consequences that are sustained from session to session.

6 Constant time scale – usually real-time, but could be any time rate, but applied equally to all users:
   a Limited to the capacity of the slowest user if fair fight is a requirement.

7 Multi-player – action is not centered on a single player; instead a user is part of a dynamic (sometimes evolving) community of users interacting simultaneously.

8 Inter-user communication – between users rather than actors. A synthetic environment might have simulated communication between actors (e.g. DIS radio) but a PSE could have additional non-simulated communication between users such as chat for purposes of coordination of gameplay.

9 Spatial environment – concepts of distance and terrain make sense and are intuitive.

Some of these characteristics are the result of a shared, multiuser experience. Many distributed simulations already exhibit some of these characteristics (e.g. multi-user effects) and this does not distinguish them from a PSE.

The first three characteristics (re-entrant, on-demand, asynchronous[11]) are not typical of contemporary distributed simulations but apply more so to PSEs.

4. Framework of Persistent Synthetic Environment technology

Combining the ideas from “re-use” and vWorlds, a hierarchy of PSE implementations can be defined:

1 Network – the persistent network infrastructure, or infrastructure or Information as a Service (IaaS) required to construct a synthetic environment for some purpose. This is analogous to a Transport layer in the sense that it includes the security accreditation enclaves, communication systems, etc. required to support the use of a synthetic environment. An example might be the Joint Training and Experimentation Network (JTEN) [12] which is a network that has a persistent accreditation and does not require re-accreditation for every change to network or synthetic environment configuration.

2 Simulation tools – the persistent availability of simulations, tool chains, applications for data collection, data analysis and review, execution management and control. This level is close to some of the concepts for “cloud” based Modeling and Simulation as a Service (MSaaS).

3 Simulations – availability of single simulations or federations ready to be configured for a particular purpose.

4 Scenarios – availability of configured simulation or federation for a particular purpose; complete with physical environment, context, order of battle and mission set; ready to run.

5 vWorld – an ongoing scenario which allows for a combination of the characteristic from Section 3.

In almost every case, commercial vWorlds follow a Client/Server architecture, where the server models the entities, environment, and interactions and the client enables user interaction. This differs from the typical M&S “distributed” simulation model. where every federate can join and supply its own entities, interactions, and aspects of the environment.

5. PSE Use Cases

In system engineering one of the more powerful techniques for analysis of requirements is the Use Case. This technique essentially asks the questions; for what, and how, will the system under consideration be used. Any sort of complete treatment of PSE’s is clearly beyond the scope of this paper, however, to illustrate the technique two simple military Use Cases will be examined: training, and concept development and experimentation (CDE). These are currently the two main users of military synthetic environments and are also where many of the high-level requirements for PSEs have appeared.

5.1 Use Case 1 – Training

Training is an extremely broad area with many different types of training occurring.

- part-task training – concentrates on sub-tasks and skills
- individual training – concentrates on individual tasks
- team/unit training – concentrates on team training within a community.
- joint training – concentrates on training of teams from multiple communities
- skill maintenance training – maintaining currency (e.g. Completion of flight hours)

To explore this further we look at a specific example. Fighter pilots must have a certain number of flight hours (in simulator or aircraft) in order to maintain currency. It is desired that these flight hours also provide the pilots with the opportunity to practice skills in a variety of air combat roles, coordinating with other aircraft and with communications with joint or coalition forces. It is up to the pilots to schedule flight or simulator time appropriately while also conducting their day-to-day functions.

Even for single aircraft training the simulator must be booked and configured by staff for the training event. For multiple aircraft training then coordination with another aircraft crew and multiple simulators is required. Using distributed simulation the simulators might be located at different sites. Extension of basic aircraft missions to interactions with other blue forces or human operated opposition forces requires coordination with the schedules for those units and linkages to their simulators. As the number of units/personnel increase the infrastructure and coordination required grows. Typically today joint or coalition training events can take over a year to plan and implement, and require substantial resources to implement.

This large resource bill conflicts with a fundamental tenet of training for multiple opportunities to practice skills, make mistakes, try new methods – to learn. It also conflicts with the underlying requirement to be able to schedule training around other activities.

Scheduling and control of what types of missions/skills are required in a scenario is a combination of the pilot and training authority. These learning requirements come from a shared review of results and opportunity to exploit or practice the skills under development.

5.2 SE Requirements from Use Case 1:

Without trying to be exhaustive the following synthetic environment requirements can be derived:

- an often on-going requirement for particular types of training, likely for the life of the system;
- ability to schedule around operations and other participants.
  - on-demand
  - minimal setup
- ability to interact with multiple organizations or training systems (likely at multiple sites)
  - distributed simulation
  - WAN – Security
  - coordinated scenario and control.
- feedback to trainees on performance
  - by instructors during simulation runs
  - After Action Review
- consolidation of learning by repeat application.
  - repeat scenarios (at least similar)
- variety of (tailor-able) experiences
  - tailored scenarios for the particular skills being trained, ability of instructor or participant to pick applicable runs for their training program
  - ability of instructor to modify scenarios to meet the skill level or participant during the run.
- practical cost/resource bills (within accepted budgets)

5.3 Use Case 2 – Concept Development and Experimentation (CDE)

Concept Development and Experimentation (CDE) is the process of developing new processes, procedures and equipment through an iterative process of experimentation. The experimentation requires the personnel to conduct their warfare tasks in multiple instances of similar scenarios with and without the condition being developed in order to obtain statistically significant evaluations of performance.

Staying with an analogous project to that considered in Use Case 1, consider the specific example of developing and validating a new air mission tactic such as anti-UAV defence. Multiple sets of aircrew are required to conduct missions using current tactics to provide a baseline, followed by the use of new tactics in similar circumstances. Actual runs are done in a randomized order with and without the new tactics.

The scenarios must include relevant mission elements that act and react in a realistic manner. This may require a combination of human and automated entities; for example actual UAV controllers to act as the opponents. Depending on level of tactic the process may include interaction with other services and higher level organizations. These participants may be in other locations and have their own schedules.

Scheduling and control of missions and scenario requirements are by the test authority. However, availability of aircrew and other human operators may constrain the scheduling.

Two key characteristics of the use case are:

- the amount of control required by the test team in order to limit variability and ensure that changes in the test metrics are due to the test condition; and,
- that each project is at least partly new so the synthetic environment infrastructure must be modifiable.

5.4 SE Requirements from Use Case 2:

Without trying to be exhaustive the following synthetic environment requirements can be derived:

- generally a CDE project requires a finite batch of simulation runs; possibly several iterative sets.
- ability to schedule runs around participant and other organization availability.
  - on demand
  - minimal setup
- consistent of experimental variables from run to run which may be separated by significant time
  - If participant repeat need different but equivalent scenarios

  - Strong control of simulated entities, generally limited changes at run time.
- ability to interact with multiple organizations or training systems (likely at multiple sites)
  - distributed simulation
  - WAN – Security
  - coordinated scenario and control.
- practical cost/resource bills (within accepted budgets)

5.5 Comparing use cases

Training only requires updates (modifications) to the synthetic environment when systems change, while CDE by nature almost always requires changes for each new project.

Both require control of the scenario but CDE requires more control at run-time to minimize confounding conditions while training may need to make changes at run-time to tailor events to the state of the participants.

Both are quite cost conscious and may require distributed environments and participants with varying schedules.

Both require availability of known configurations over significant time periods: training for the life of the system/process; CDE for life of the experiment series. Both have a requirement for on-going finite length scenarios. Construction of specific, controlled, scenarios are typical in both of these use cases, as is a finite scenario length.

5.6 Use Case Requirements for Persistence

Revisiting the PSE characteristics from above, Table 1shows the requirements from the two use cases for each of the PSE characteristics.

Table 1 – PSE Characteristics of Use Cases

PSE Characteristics	Training	CDE
Re-entrant	No	No
On-demand	Yes	Depends
Asynchronous usage	Depends	No
Multi-user effects	No	No
Sustained effects	Depends	No
Constant time scale	Yes	Yes
Multi-player	Depends	Depends
Inter-user communication	Yes	No
Spatial environment	Yes	Yes

From the results in Table 1 it is fairly clear that use cases for a vWorld with a single, persistent, long running scenario – a persistent storyline with re-entrancy and multi-user effects – are few and far between. Further maintaining long running storylines is likely to increase the human cost of the system and therefore likely to be cost prohibitive.

6. PSE Components

A second system engineering technique is to assume a non-complex system that can be broken into understandable components which can then be assessed with respect to the system objectives. Table 2 shows typical components of a PSE with examples, grouped into categories and matched against some PSE objectives.

The re-use of components listed in Table 2 for a given objective might vary slightly depending upon specific implementations. In general we see increasing re-use from left to right amongst the listed objectives. The re-use of components is considered with respect to four objectives listed as follows:

1 Persistent metadata: The first objective is to re-use metadata such as standards, designs and other documents [14]. Persisting metadata is a mature and well-understood activity, typically using a document control capability including people and technology.

   Somewhat paradoxically, metadata is re-used less as it is only required during design, construction and modification of the other components. Once the other components are tested and ready, the metadata is no longer immediately useful.

2 Persistent Simulation Assets: The second objective is to re-use specific assets such as hardware, software and data sets [15]. As with metadata re-use, configuration control for persisting software and data assets is necessary and well understood. Re-use of the assets implies the re-use of the human effort invested to develop it. This also applies to the effort to develop the metadata re-used in the first objective, but is not specifically mentioned here.

   Persistence of hardware assets is subject to failures and obsolescence. Availability of repair parts and skilled workers may degrade over time as technology progresses.

Table 2 - PSE Components

			objective			
Component	**examples**	**Category**	**re-use docs**	**re-use assets**	**on demand**	**vWorld**
conceptual model	DIS enums	standards	x	x	x	x
interoperability standards	DIS, HLA, WebLVC		x			
IEDM	FOM		x			
security	agreements		x			
framework	fed design, fed agreement	technical	x	x	x	x
network (inc. services)	JETN, CFBLNet			x	x	x
hardware	PCs, servers			x	x	x
interoperability services	RTI, gateway			x	x	x
simulators	SAF, CGF, flight sim			x	x	x
viewers	2D PVD, 3D stealth			x	x	x
logging services	data logger			x	x	???
replay	AAR			x	x	???
health monitor	ganglia			x	x	x
lobby	online game management				x	x
models	3D models, behavior	data		x	x	x
terrain	terrain files			x	x	x
scenario	sub hunt	scenario			x	
storyline	campaign					x
world	2nd life					x
developers	SLAs, standing offers	people		x		
support	IT, SME			x	x	x
operators	actors, game admins				x	x

3 Persistent Synthetic Environment: The third objective is to develop an on-demand PSE that could be configured and used with negligible configuration and set up delay [16]. This means re-use of assets, but also existing scenarios called up on-demand.

New simulation assets are required to provide an on-demand PSE, which organizes and automates the start-up and shutdown of simulations for groups of participants. New considerations apply also to on-demand availability of human resources including IT departments and possibly human actors.

A persistent environment may be asynchronous, where individuals or groups join a scenario, and interact in their own world [11], or could be synchronous where everyone appears in one virtual world.

The persistent environment may be a long-running scenario, but doesn't approach the complexity of the fourth objective:

4 Persistent Storyline: The fourth objective is to create an always-on vWorld which at least conceptually never ends. This has the same requirements as the third objective, with two differences:

First, the scenario doesn't end, so there is no possibility to reset to initial conditions. Destroyed simulated units, targets, etc., stay destroyed – unless there is some intervention, either automated or supervised, which replaces or restores destroyed entities. In MMORPGs this problem is typically solved by "spawning" new re-creations of players, monsters, etc. How this is handled in a simulation context must be carefully considered.

Second, the environment of the vWorld is also persisted. Terrain features such as bridges stay destroyed just as other simulated entities. But more profoundly, the future behavior of automated or non-participant actors should reflect those persistent environment changes as well. Simulated crowds would probably avoid bombed-out marketplaces. Logistically, neither friend nor enemy could cross a destroyed bridge.

A simulation can be adapted to efficiently manage the storyline and training objectives; similar to how MMORPGs sometimes handle persistence differently in different geographic regions in a vWorld. For example, one player group might raze a castle; subsequent visits by the same player group would not see the castle. Other player groups might see the castle until each group razed it themselves.

The prime difference between the on-demand PSE and the vWorld PSE is the re-use of a scenario versus the persistence of a scenario.

In the gaming MMORPG worlds most of the players' actions are being monitored and moderated by the other community members, but also by the game administrators. Game administrators (also known as game masters) are active characters in the game and they support other players that use the help chat channel, by receiving feedback on program errors or reports on some players' aggressive and rude behavior.

Therefore a certain amount of human interaction is required to maintain the systems, environment and scenarios, and moreover the characters continue to exist after the user signs off for the day. Virtual worlds require some "reality" that remains even when the user is logged off.

7. Discussion

It is important to examine the motivation for considering a PSE and relevant use cases to determine if the benefits warrant the associated effort and cost. The scope of persistence should be considered. Are changes persisted to a single user, to groups of users or to all users?

As with other systems engineering processes it is important to understand the goals and objectives before working on the requirements. The Distributed Simulation Engineering and Execution Process (DSEEP) [17] created a well-defined and practical process for creating simulations. It suggests a federation developer:

- Define Federation Objectives, including
  - Goals,
  - Requirements,
  - Constraints (budget, time)
- Perform Conceptual Analysis
  - Describe scenarios,
  - Develop conceptual model
- Design Federation
  - Select federates,
  - Assign responsibilities, etc.
- Develop Federation
  - Adapt existing federates,
  - Develop new ones
- Integrate and test
- Execute
- Analyze and Evaluate
- Provide feedback

The challenge in M&S is to model an environment to sufficient fidelity to suit the goals and requirements of a federation. For PSEs, the designers need to pay particular attention to:

1 Audience: Who will be using the PSE? What culture/ educational background, and intended purpose of the PSE. What critical mass of participants is required for a useful simulation run?

2 Time zones: How often will people be interacting, at different times of the day (e.g. work-hours, evenings, another time zone)? What is simulation time if logging in from a different part of the world? Are 24/7 support staff required?

3 Reliability: How often is the infrastructure expected to be available? Are scheduled downtimes acceptable?

4 Funding model: A PSE requires not only setup, but also funding for continuous upkeep of the systems for the life of the project.

5 Project Lifespan: What is the expected lifespan of the project or a particular scenario requirement?

A major consideration for PSEs is how the environment and entities will persist and change over time. Some changes might be structured or initiated by administrators or automated simulations. Other changes might originate from the users themselves. For example, a user group might plan and construct a simulated village in a vWorld. This free-form change would not require administrators or automation. The changes occur within the conceptual model and bounds of the PSE.

A PSE might need to change continuously in the case of asynchronous usage, e.g. changing the environment for higher resolution terrain, or adding new interaction types/consequences. Because the timing of a user's interactions might not be known in advance, these types of changes have to be handled in a sophisticated manner so there is minimal loss of realism in the vWorld.

Alternatively, a PSE might change stepwise, between sessions. Each new session might depend on the previous session plus changes introduced by administrators. Such a stepwise change is easier to specify and implement than a continuous change, but it would impact all users simultaneously upon its introduction.

Verification, validation and accreditation (VV&A) poses new challenges to the PSE developer. The VV&A must account for persisting environment and scenario changes, as well as changes to simulation assets themselves. In the case of a vWorld which changes continuously is generally always available, it might be necessary to perform VV&A periodically or even continuously to maintain confidence in the PSE.

There are different levels of interaction. A commander gives orders from HQ, and goes home for the evening, trusting that the individual units will follow through. Creating a persistent environment for a pilot requires a more hands-on approach, Esp. for practical training value.

Persistent systems differ from "long running" scenarios that might last for a week, or even a month. In case of a finite length scenario, applications, and resources (e.g. disk space) only need to be capable of handling a fixed, computable load. If there is a memory leak in an application, the issue can be resolved by simply adding more RAM. Persistent operations require more rigor.

In an ideal world all programs would be error free, and there would be no corruption of data. Danner [16] points out an Event Analysis tool that is setup at every location, and allows post-event analysis. For persistent systems, the software will be running for long periods of time, and therefore require more sophisticated error/anomaly tracking tools, redundancy, and fallback systems. If there is a failure other applications can take over modelling via ownership transfer. The cause of failures and tracking of the error/anomaly can ideally be handled without (noticeable) user interruption.

8. Conclusion

This paper has examined the concept of persistent simulation environments (PSE) that often appears in high-level requirements statements for future military training and concept development systems. The idea of persistence for simulation environments has its roots both in the online gaming industry and in the need to reduce cost by enhancing re-use.

This examination of persistence and re-use shows that there is a spectrum of implementations that can be deemed to be persistent; ranging from persistence at the transport layer through to the full 2nd Life-esque 24/7 online environment. The fundamental question a simulation environment designer must address is what type and level of persistence will actually fit the use cases for the application, in a cost-effective manner. Current simulation systems engineering processes (such as DSEEP) provide the mechanisms to work through these issues.

The two most common military use-cases for persistent simulation environments (training and concept development) were examined to determine the general level of persistence required. In both cases the requirement for scenario control and repeatability indicate that a fully persistent environment would not be appropriate. Instead, the primary persistence attribute is that of availability. Thus, the persistence is in the scenario setup, infrastructure, network etc., but not in the particular effects generated in a particular simulation run. Using a game analogy this is similar to a World of Warcraft dungeon instance that always has the same story thread and is always available for play but the results of one run of the instance have no effect on the next.

Designers will also have to grapple with the trade-offs between the cost of setup for new scenarios and the cost of modification of scenarios to maintain currency, relevancy and participant interest. It is likely that the applications and testing tools required for continuous availability require more rigor and analysis.

In the end there is no one all-encompassing implementation of a persistent simulation environment (PSE) but this paper has provided an initial framework to help designers to identify the scope of the actual requirements.

9. References

[1] N. Harrison, R. Lestage: "Science and Technology Support to Concept Development and Experimentation" RTO NMSG Symposium on Modelling and Simulation to Address NATO's New and Existing Military Requirements, held in Koblenz, Germany 7-8 October 2004 NATO RTO-MP-MSG-028, http://ftp.rta.nato.int/Public/PubFullText/RTO/MP/RTO-MP-MSG-028/MP-MSG-028-10.pdf Accessed May 2014

[2] Lieutenant-Colonel J.L.D. Lachance: "PROJECTING POWER: Alternative Futures for Canada's Air Force in 2020", Department Of National Defence Canada, 2011. Downloaded from: http://airforceapp.forces.gc.ca/CFAWC/eLibrary

/pubs/Projecting_Power_2020.pdf, June 2014

[3] R. Thompson: "The Canadian Advanced Synthetic Environment" – Creation of a distributed simulation environment to support Air Force MA&S activities", JSMARTS Vision Workshop, Ottawa, Canada. Downloaded from: http://admmatapp.dnd.ca/cosmat/dmasp/downloads/ModellingSimulation/Presentations/Thompson.ppt

[4] Ceranowicz, R.W. Dehncke, T. Cerri: "Moving Toward a Distributed Continuous Experimentation Environment", Interservice/ Industry Training, Simulation, and Education Conference 2003, Orlando, FL, USA. Downloaded from: http://www.alionscience.com/pdf/Dist_Environment.pdf

[5] Bell (2008): Toward a Definition of "Virtual Worlds". Journal of Virtual Worlds Research, Vol. 1, No. 1. July 2008. Jvwresearch.org

[6] L. Perez: "Introduction to Second Life," 2008, downloaded from www.coedu.usf.edu/we/TeachingResources/secondlife.pdf,

[7] N. Yee: "The Psychology of Massively Multi-User Online Role-Playing Games: Motivations, Emotional Investment, Relationships and Problematic Usage", The Psychology of MMORPGs: Emotional Investment, Motivations, Relationship Formation, and Problematic Usage. In R. Schroeder & A. Axelsson (Eds.), Avatars at Work and Play: Collaboration and Interaction in Shared Virtual Environments (pp. 187-207). London: Springer-Verlag. Downloaded from: www.hci.iastate.edu/REU09/pub/Main/723/yee-psychology-mmorpg.pdf

[8] http://www.oxforddictionaries.com/definition/english/persistent

[9] Lemmers, A. Gerretsen, S. Skinner: "Missionland: The Creation of a Virtual Continent for Mission Simulation" National Aerospace laboratory NLR. NLR-TP-2011-157. http://reports.nlr.nl:8080/xmlui/bitstream/handle/10921/122/TP-2011-157.pdf?sequence=1 Accessed May 2014.

[10] Lemmers, A. Gerretsen, M. Dahlberg, M. Aasen: "Re-using Real World Data for the Fictitious Missionland Continent" National Aerospace laboratory NLR. NLR-TP-2011-337. Aug 2011. http://reports.nlr.nl:8080/xmlui/bitstream/handle/10921/139/TP-2011-337.pdf?sequence=1 accessed May 2014.

[11] J. Threlfall, M.W. Freeman, J. Roderick: "Persistent Learning Capability: It's Your MOVE", I/ITSEC 2011

[12] S. Croom-Johnson, W. Huiskamp, B. Möller: "Security in Simulation – A Step in the Right Direction", 13F-SIW-009. Fall Simulation Interoperability Standards Organization September 2013.

[13] W. Riggs, K.L. Morse, R. Brunton: "Emerging Solutions for LVC Asset Reuse", 11S-SIW-034. Spring Simulation Interoperability Standards Organization April 2011.

[14] E. Andrew, C.D. Turnitsa, A. Tolk: "Software Reuse for Modeling and Simulation", 11S-SIW-042, Spring Simulation Interoperability Standards Organization April 2011.

[15] B.P. Danner: "Cross Domain Solution (CDS) Certification and Accreditation for Persistent, Simulation Training" Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2009

[16] Institute of Electrical & Electronics Engineers. "Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)", IEEE Std 1730-2010 (Revision of IEEE Std 1516.3-2003), vol., no., pp.1-79, Jan. 24 2011

10. Further Reading

[17] J. Braman, A. Jinman, G. Trajkovski: "Towards a Virtual Classroom: Investigating Education in Synthetic Worlds" (2007) http://www.aaai.org/Papers/Symposia/Fall/2007/FS-07-04/FS07-04-004.pdf accessed May 2014

[18] Chadbourne, D. Clark: "Building, Using, Sharing and Reusing Environment Concept Models"(2006) Visitech Ltd Alexandria Va. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA444524

[19] R. Kneebone, S. Arora, D. King, F. Bello, N. Sevdalis, E. Kassab, R. Aggarwal, A. Darzi, D. Nestel: Medical "Distributed simulation – Accessible immersive training" Teacher, Vol. 32, No. 1, pp. 65-70, 2010, http://informahealthcare.com/doi/pdf/10.3109/01421590903419749

[20] S. Strassburger, T. Schulze, R. Fujimoto: "Future trends in distributed simulation and distributed virtual environments: results of a peer study" In Proceedings of the 40th Conference on Winter Simulation (pp. 777-785). Winter Simulation Conference, December 2008.

[21] T. Manninen: "Virtual team interactions in Networked multimedia games - case: 'Counter-Strike' - Multiplayer 3D Action Game" In Proceedings of PRESENCE2001 Conference, (Philadelphia, USA, May 21-23, 2001). Downloaded from http://homes.chass.utoronto.ca/~tkennedy/Courses/38H3/Papargyris.pdf June 2014.

[22] M. Macedonia: "Virtual World Technologies and Applications" Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) (2010) training session slides.

Author Biographies

BRAD DILLMAN is a principal systems engineer at VT MÄK, Cambridge, MA, USA. Mr. Dillman has a B.Eng. from the Technical University of Nova Scotia and 24 years' experience in real-time and embedded distributed systems research and development. He currently works on distributed simulation technology and applications. From 2008 to 2012 Mr. Dillman was the project manager for the virtual VICTORIA Capability Evaluation Laboratory at DRDC Atlantic Research Centre.

MARK G. HAZEN is a senior scientist with Defence R&D Canada in Dartmouth, NS, Canada. He has been researching the use of distributed synthetic environments since 2001 and has 28 years of experience in using simulation to support defence research and analysis. He is currently the Canadian National Leader for TTCP Joint Systems Analysis Technical Panel 2 on M&S Technology.

BOBBY CHAWLA is the Founder of Real-Time Engineering and Simulation (RTeng.Pro), a Professional Consulting firm specializing in Real-Time Software Development, Modelling and Simulation. Prior to establishing RTeng.pro, Mr. Chawla worked for the R&D department of QNX, a Real-Time OS; the Simulation & Training division at xwave; and the M&S group of CAE. He has worked on various military M&S projects over his 20-year career and is known for his problem solving, and out-of-the-box innovative solutions. Mr. Chawla holds a degree in Mechanical Engineering from McGill University, Montreal, Quebec, Canada and a Master's of Science in Automation from University of Waterloo, Ontario, Canada. If not at work, you might find Mr. Chawla piloting a Cessna around the Ottawa area.