Overcoming Static Structures: Dynamic Buildings for Training

Jeff Lyons, Philip Sowden
Applied Research Associates, Inc.
Orlando, FL
jlyons@ara.com, psowden@ara.com

Julio de la Cruz, Hector Gonzalez
ARL STTC
Orlando, FL
julio.delacruz@us.army.mil, hector.gonzalez24@us.army.mil

ABSTRACT

Urban warfare continues to be critical in U.S. and allied military conflicts. As a result, government and industry have made great technological strides to support simulation and training in urban environments. However, the ability to simulate building damage with sufficient fidelity remains a significant capability gap.

On the battlefield, building damage events significantly impact tactics. For example, resulting rubble affects vehicle mobility, warfighters use explosives to form breach holes for building access, and building damage events incapacitate building occupants. Current training simulations, whether live, virtual or constructive, do not adequately model these effects.

Solutions from the analysis domain could be leveraged to address these shortcomings. These tools calculate building component damage, equipment damage, collapse, and rubbling effects. The DoD uses these models for weaponeering, but recent research has shown they can be adapted for real-time training.

However, challenges remain. Current training simulation tools do not support high-fidelity changes to their building formats. To support the aforementioned use cases, simulated buildings must be able to break apart, form breach holes, collapse, damage in specific areas, and rubble. Additionally, existing building formats lack the attribution required for high fidelity calculations (e.g., component material type, stud spacing, joint strength, etc.) Finally, distributed simulations are unable to communicate detailed building damage from node to node, resulting in uncorrelated buildings.

In this paper we discuss the state of the art of building damage in the training domain. We will discuss the entire problem space at a high level. We then propose solutions for two specific challenges. We propose the adaptation of specific analysis tools for the simulation domain. We outline our approach, as well as its challenges and limitations. We also present a methodology for communicating high fidelity building damage in a distributed simulation.

ABOUT THE AUTHORS

Jeff Lyons is an engineer at Applied Research Associates, Inc. He has a Bachelor of Science degree in mechanical engineering from Florida State University and a Masters in mechanical engineering from the Massachusetts Institute of Technology. Mr. Lyons's experience spans the mechanical engineering and software engineering disciplines. He has experience in structural analysis, CAD design, manufacturing support, and test support. He has worked in simulation and software design for military training and testing systems for over ten years. Jeff leads the development of the Real-time Physics Effects Library (RPEL), which aims to advance the state of the art in real-time casualty assessment.

Phillip Sowden is a software engineer at Applied Research Associates, Inc. He has six years of experience developing live training systems for the U.S. Army with a primary focus on real-time casualty assessment. Programs he's worked on include: LTEC, LT TES TCC, RPEL, AWES, and OneTESS. He earned a BS degree in Computer Engineering from the University of Central Florida.

Julio de la Cruz is the Chief Engineer for Synthetic Environments Research at the Simulation and Training Technology Center in Orlando, Florida and is responsible for the research and development of applied simulation technologies for learning, training, testing and mission rehearsal. Mr. de la Cruz has a Bachelor of Science degree in

Electrical Engineering from the City College of New York and a Master of Science degree in Industrial Engineering from Texas A&M University. He is a graduate of the U.S. Army School of Engineering & Logistics and alumnus of the Advanced Program Managers Course at the Defense Systems Management College (DSMC).

Hector Gonzalez is a Science and Technology Manager at the U.S. Army Research Laboratory, Simulation and Training Technology Center and supports research and development efforts in Synthetic Natural Environments for training applications. Some of the research and development topics being explored include real time physics effects of weapons and explosive devices on different environment elements; modeling of complex underground structures; the modeling and representation of dynamic synthetic environments; and Live, Virtual, and Constructive training applications. Mr. Gonzalez was a systems engineer with the NAVAIR Training Systems Division working on Common Acoustic Analysis databases, Submarine Piloting and Navigation trainers, and Combined Arms Command and Control Training Upgrade Systems (CACCTUS) program. He is DAWIA Level III certified in Systems Planning Research Development and Engineering (SPRDE) Science and Technology Management, and SPRDE Systems Engineering. Mr. Gonzalez has over 28 years of experience and holds a B.S. in Electrical Engineering from University of Puerto Rico.

Overcoming Static Structures: Dynamic Buildings for Training

Jeff Lyons, Philip Sowden
Applied Research Associates, Inc.
Orlando, FL
jlyons@ara.com, psowden@ara.com

Julio de la Cruz, Hector Gonzalez
ARL STTC
Orlando, FL
julio.delacruz@us.army.mil, hector.gonzalez24@us.army.mil

INTRODUCTION

Urban warfare continues to dominate ongoing U.S. and allied military conflicts. Government and industry have responded in kind, advancing the state of the art in urban operations (UO) training and simulation solutions. The U.S. Army has deployed several live, virtual, and constructive solutions that enable UO training.

For example, live-domain training includes the Urban Assault Course (UAC), the Combined Arms Collective Training Facility (CACTF), and the Combat Training Centers (CTC). The virtual domain includes the Close Combat Tactical Trainer (CCTT) and the Engagement Skills Trainer (EST). The constructive domain includes the Joint Conflict and Tactical Simulation (JCATS) and OneSAF (Department of the Army, TC 90-1, 2002). These training solutions provide UO options, but with significant capability gaps: none of these trainers adequately model building damage effects.

On the real battlefield, building damage events significantly impact tactics. Building rubble impedes vehicle mobility in battle zones. Warfighters create breach holes for building access. Building damage events incapacitate building occupants. Current training simulations do not accurately convey these events due to the technical difficulties involved in making simulated buildings dynamic.

Figure 1. Live, virtual, and constructive simulations train warfighters for urban operations, but dynamic environment capabilities are limited.

Various domains have responded to the need for UO solutions. Game developers have added dynamic building events to their tools by creating their own custom building damage models. The fidelity of these models varies wildly. Some use very simple material attribution and first-principles physics to generate damage. Others use no real model at all, breaking down the building in a way that entertains, but is not realistic. Games don't consider civil engineering and physics-based models, which take into account building construction parameters (e.g., beam and column parameters, stud spacing and geometry, etc.). Without verification and validation (V&V), these early attempts at real-time building damage in games should be considered artificial—not at all based in reality. Even so, these gaming efforts are useful in that they have started to tackle the challenging problem of dynamically changing traditionally static structures in real-time. (We will not

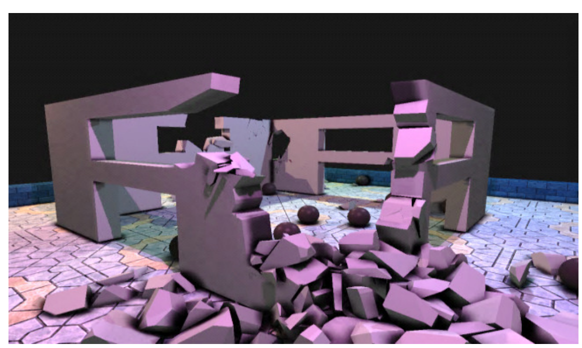

Figure 2. Some game and physics engines have added impressive real-time fracturing capabilities, but the fracturing models do not use engineering attributes and are not validated.

delve into implementation details for real-time environment modification in this paper, since implementation depends largely on the format being modified.)

Like the gaming community, the analysis domain has responded to the need for UO solutions. Analysis tools, used by the DoD for weaponeering and site protection, calculate building component damage, equipment damage, collapse, and rubbling effects. In recent years, these historically slow-running models have gone from calculating results in hours to finding results in minutes or seconds, sometimes less than a second. In addition, analysis tools have already gone through rigorous V&V, which can be costly and time-consuming. For these reasons, we believe training solution providers should consider adapting analysis tools for real-time use.

Our focus in this paper will be threefold: 1) We make the case that dynamic structures are critical to warfighter preparation for UO. 2) We discuss our approach for adapting analysis tools for real-time building damage calculation. Finally, 3) We tackle one final implementation challenge: communicating building damage in large-scale exercises. As simulations evolve to support dynamic environments, large scale exercises must adapt to keep their simulations' environments correlated at runtime. Current simulation network protocols do not support sharing runtime building changes, so sharing these across disparate simulations will be a significant undertaking.

WHY BUILDING DAMAGE IS CRITICAL TO TRAINING

Warfighters have been training for urban operations for many years. Unfortunately, military training systems lack the ability to dynamically change buildings and other structures. On the real battlefield, buildings change. They collapse and rubble, walls and studs fail, walls are breached, and power systems fail. Let's consider how some of these events could affect training results.

Rubble generated from building damage is a major impediment to warfighter mobility, particularly when operating vehicles. Few simulations model rubble, though rubble is a challenge urban warfighters often encounter. Without training that includes rubbling, soldiers are not ready to negotiate rubble piles.

When buildings are attacked, occupants are injured or killed. Simulations do not model these effects, so soldiers are not prepared to deal with the ramifications of building damage events. Trainees could even be lulled into a false sense of security in these simulated, indestructible buildings. These false perceptions cause negative training.

Room clearing is an extremely difficult skill to learn. One common danger with respect to room clearing is friendly fire, particularly stray bullets penetrating through walls. Enemies shoot through walls, hoping to stop the BLUFOR before they are in sight. Modeling *shooting through walls* requires dynamic buildings and higher fidelity models.

Finally, it is common for building power systems to fail in battle. Simulations, particularly virtual and constructive, do not model power failures. The ability to train with lights on or off would improve warfighter readiness.

As all of these examples demonstrate, without dynamic building events warfighters are not fully prepared for the many contingencies that occur in urban warfare. Modeling these events will better prepare our warfighters for these and other contingencies.

REAL-TIME BUILDING DAMAGE CALCULATION

Until recently, no one had tried to calculate physics-based building damage in real-time; the tools were too slow and cumbersome for real-time. Analysis tools typically have tightly integrated user interfaces (UI) and other limitations (e.g., restrictions on building geometries with non-orthogonal walls) that make them unusable for real-time training. Fortunately, in recent years, some of these tools have addressed these limitations, removing UI dependencies and significantly speeding up processing times. Air Force Research Laboratory's (AFRL's) Modular Effectiveness Vulnerability Assessment (MEVA), Defense Threat Reduction Agency's (DTRA's) Integration Munitions Effectiveness Assessment (IMEA), and DTRA's Vulnerability Assessment Protection Option (VAPO) have all moved toward modularity, quick runtimes, and increased flexibility. This section outlines our efforts adapting one of these tools for real-time training and simulation.

Calculating Building Damage in the Analysis Domain

The process for calculating building damage is similar for most weaponeering analysis tools. Figure 3 shows the process for MEVA (Applied Research Associates, 2006). First, MEVA performs a penetration calculation, which causes penetration holes and returns a munition's detonation location. The tool calculates any resulting breach holes. MEVA modules propagate air blast pressure through the building, by constructing a grid inside the building and calculating the pressure for each cell (see Figure 4). Algorithms propagate air blast pressure through the grid until pressures drop to a configured minimum. MEVA calculates component failure for every building component touching a cell with pressure. Finally, MEVA determines if the building should collapse due to inadequate support.

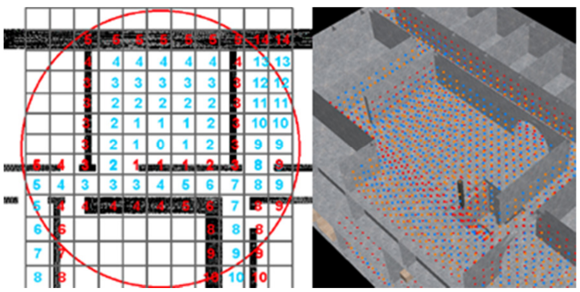

Figure 4. MEVA grids building space and propagates pressures through rooms and hallways, which can be applied in two (left) and three dimensions (right).

Using Building Damage Calculations for Training

Building damage calculations from the analysis community provide V&V'd, realistic results. However, to be useful for training, building damage events must also look realistic. This requires dynamic visual effects for rubble, component fracture, and building collapse. Analysis tools often provide building visualization before and after the building damage event, but they do not show dynamic movement or changes to the building's geometry between end states. Fortunately, we can infer dynamic visual effects from the analysis tool results. We integrated a building damage analysis model (MEVA) with a real-time effects framework, the Real-time Physics Effects Library (RPEL), to deliver visual effects in real time. The remainder of this section describes our approach.

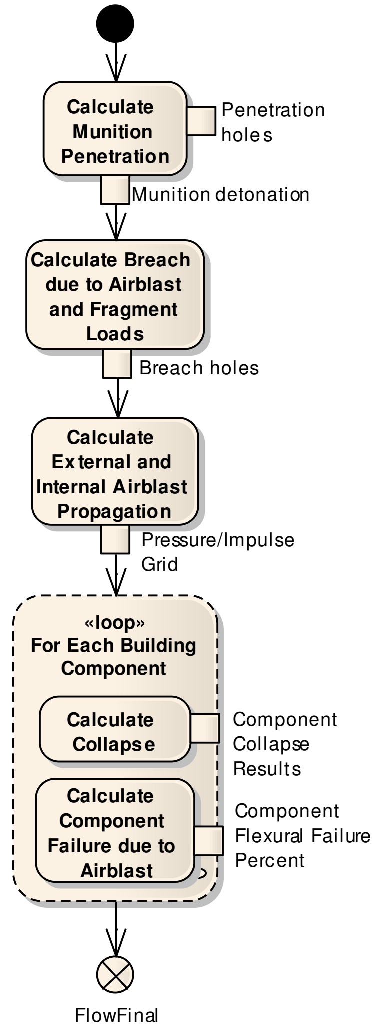

Figure 3. MEVA building damage calculation results include penetration and breach holes and component flexural failure percentages.

Rubble and fragments from damaged building components should separate from the building and fly out according to the damage event forces. Additionally, collisions with other shapes in the environment must be detected and the effects of those collisions must be represented realistically. RPEL uses a physics engine for this functionality. When a building is loaded, RPEL adds its components to the physics engine's environment. When a building damage event occurs, RPEL and its associated physics engine perform flyout and collision calculations by applying gravitational and air-blast forces to components.

Building damage events can produce holes in building components due to munition and fragment penetration. MEVA represents these holes as shapes; often they are circles with a given center and radius. For training, we must cut these holes through the component's geometry. Additionally, RPEL can generate rubble from holes in the building and fly it out to provide visual realism. RPEL's physics engine represents rubble as geometric primitives (e.g., cubes or spheres) to simplify collision calculations, but rubble could be displayed as more jagged rubble-like shapes.

MEVA provides building component damage as a percentage. To be useful in a training context, damage results should dynamically affect the component's visual representation. To look realistic, damaged building components must be capable of fracturing and separating from the building.

We deduced building component fracture patterns from the component's failure mode, geometric properties, material properties, and connection properties. We categorized component failure modes into either flexural failure (e.g., failure due to blast pressure) or buckling (e.g., failure of vertical supports due to increased load), allowing us to pre-determine the component's fracture pattern.

Figure 5 shows typical fracture patterns. Pre-fracturing a building (i.e., fracturing components at load time) provides a significant performance advantage over fracturing during a damage event. We were able to pre-fracture building components, since all of the information to determine component fracture patterns is known at load time[1].

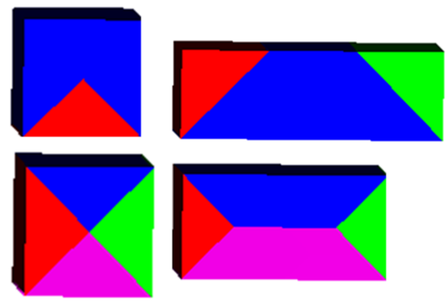

Figure 5. Typical fracture patterns. Patterns can be predicted before a component is damaged, so we pre-fracture them before a simulation even begins.

Analysis tools may not directly calculate when a component should be separated from the building. MEVA, for instance, does not calculate when connections between components have failed. Based on guidance from MEVA developers, we used a 100-percent damage level to indicate that a component should be separated from the building. We also generated fragments (i.e., pieces produced by the applied fracture pattern) from failed building components and flew them out. MEVA gives the impulse magnitude and direction applied to a component. When we determined the component should be separated from the building, we flagged it as dynamic in the physics engine and applied a portion of the impulse to the component's fragments. More research is required to characterize realistic rubble zones from failed components.

Figure 6 shows a visualization of real-time building damage using MEVA results from our integration effort. The figure shows a simulation of the detonation of a vehicle-borne improvised explosive device inside a five story building with concrete slab construction. Red components have totally failed and separated from the rest of the building. We fragmented these components and flew them out from the building based on pressure impulses. The smaller fragments are rubble generated from the explosion.

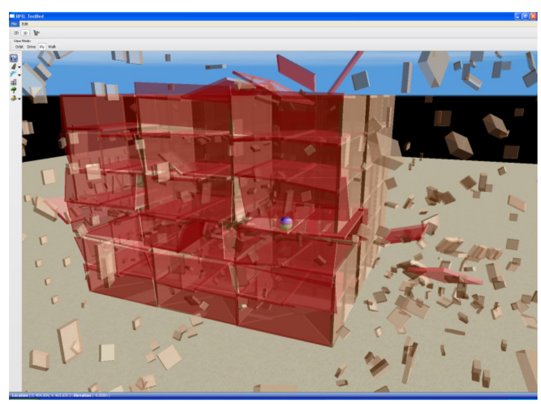

Figure 6. Building damage visualization showing damaged components separated from a building and fragment flyout

Our effort included visualization of the damage as a means of verifying our results. With a few modifications, a similar visualization method could be used for training. Most notably, the application of representative textures and fragments with jagged edges would greatly enhance realism.

[1] We assumed that flexural failure, the most common failure mode, occurred. A higher fidelity implementation would take into account the possibility of any other, rarer failure modes.

COMMUNICATION OF EFFECTS

In the DoD test and training community, most simulations interact and interoperate with other simulations. Disparate simulations interoperating with one another form a large scale simulation exercise, sometimes referred to as a federation. Federations can simultaneously include federates[2] from the live, virtual, and constructive domains.

For real-time building damage to be useful in these large scale exercises, results must be communicated over a federation. In this section we explore the challenging problem of communicating building damage results across a federation to disparate, incompatible federates. Before we discuss how to communicate these effects, we must consider how simulations represent buildings.

Building Representation

The DoD test and training communities have made little effort to make building formats consistent or interoperable. With static building structures, there is little need to do so. As long as static structures are correlated across federates, simulations can represent buildings in any way that meets their own requirements. Simulations get consistency by generating correlated databases pre-runtime.

Table 1. Simulations typically use native building formats that are not interoperable with one another

Simulation/ Game	Type	Building Format
CCTT	Virtual	Insurgent buildings
JSAF	Constructive	CTDB
OneSAF	Constructive	UHRB
VBS2	Virtual, Game	P3D buildings
Unreal	Virtual, Game	Unreal buildings
MEVA	Analysis	STMG format
IMEA/VAPO	Analysis	IMEA buildings
RPEL	Framework for adapting analysis tools to real-time LVC	Collada physics

As we move to dynamic buildings, this lack of consistency in building representation presents a significant challenge. Table 1 illustrates the magnitude of the interoperability problem. Each building format contains the data its associated simulation requires. For example, IMEA, VAPO, and MEVA require engineering attributes to perform their damage calculations. This attribution is not present in any of the other formats. UHRBs contain data that enables SAFs to move intelligently through the building. A format like P3D takes a completely different approach, having been optimized for game performance.

The architecture, engineering, and construction (AEC) building industry has struggled with similar format compatibility problems. It is worth considering how their struggle might apply to our domain. In the late 1990s and early 2000s, the AEC industry saw the emergence of a design technology poised to replace computer-aided design (CAD). AEC industry leaders called the approach “building information modeling,” or BIM. As Smith and Tardif (2009) explain, “The geometry of a building represents only a small percentage of the total body of useful information about that building. A genuinely comprehensive building information model would encompass not only geometry but all of the information about a building that is created throughout its useful life…The building information would be accessible to many different types of users—building owners, operators, constructors, facility managers, portfolio managers, and even emergency responders—through user interfaces that are accessible and familiar to each.”

[2] The term federate refers to an individual simulation participating in a federation.

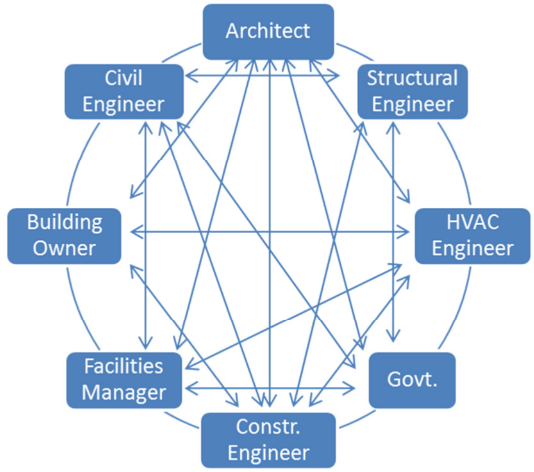

Figure 7. Before BIM, building stakeholders communicate in an ad hoc fashion.

As illustrated in Figure 7, before BIM, AEC professionals had no mechanism to facilitate communication. BIM had the potential to revolutionize the AEC industry, but its adoption was almost derailed by its own proponents. Early BIM enthusiasts proposed a "single building model" that could be "developed and sustained throughout the life cycle of a building facility" (Smith and Tardif, 2009). AEC professionals quickly discovered the single building model approach was impractical. Who would have stewardship for the model? Who would maintain it? Who owns it? What about security, reliability, and liability? To address these concerns, the industry scaled back the vision to a more practical approach: using BIM for information exchange. In this way BIM streamlines communications and business processes that already exist, as opposed to inventing a new way of doing business. For these reasons, in recent years the AEC industry has focused on achieving BIM through the development of a standard interface exchange format. This approach was more evolutionary than revolutionary, which helped it achieve widespread adoption.

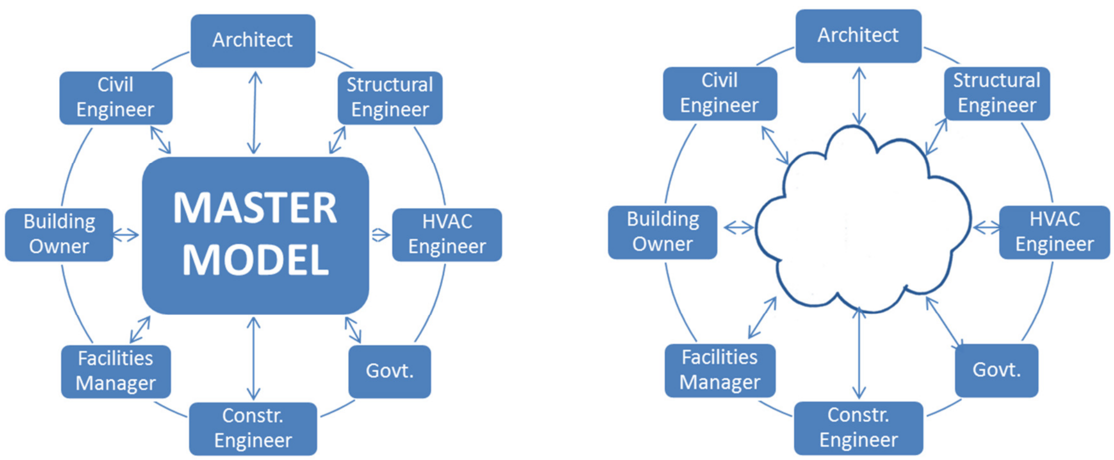

Figure 8. AEC industry's initial BIM approach (left) advocated for a single master model. This was abandoned in favor of a more pragmatic approach: using BIM to facilitate information exchange (right).

How applicable is BIM to the DoD simulation and training domain? Replace the stakeholders in Figure 7 and Figure 8 with typical simulation federates: OneSAF, JSAF, VBS2, etc. BIM facilitates the communication of building models; we wish to do the same, except this time as part of a federation. Consider the following definition:

"Building information modeling is nothing more—and nothing less—than a systems approach to the design, construction, ownership, management, operation, maintenance, use, and demolition or reuse of buildings. A building information model is any compilation of reliable data—in single or multiple electronic data formats, however complete or incomplete—that supports a systems approach in any stage in the lifecycle of a building" (Smith and Tardif, 2009).

A building's lifecycle includes military operations conducted inside the building; by this definition, UO simulation and training are within the scope of BIM. More importantly, there is significant overlap between the building data required to conduct BIM for AEC professionals and the data required to create dynamic, intelligent buildings for simulation and training. Therefore, we should take a close look at what we can leverage from (and possibly contribute to) the BIM movement.

Possible Communication Mechanisms

As discussed earlier, we cannot realize the benefits of real-time building damage until effects can be communicated across a federation. The Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) protocols are typically used for federation communication. Typical protocol data units (PDUs) associated with these protocols include entity information and state, weapon firings and detonation events, logistic events, simulation management, and radio communications (IEEE, 2012). There are no PDUs for, nor are the protocols designed to handle, dynamic building events.

Thus we must explore new communication options. We will consider three approaches.

1. Communication of dynamic building event PDUs containing alpha-numeric parameters
2. A shared, single master model
3. Communication via a standard interchange format

Approach 1: Events Containing Alpha-numeric Parameters
The approach of communicating dynamic building event PDUs with alpha-numeric parameters is most similar to the DIS/HLA paradigm. When we communicate entity state over DIS/HLA, we identify the entity by ID and list parameters describing its state. Typical parameters include location, health, and supply status. Individual entities belong to higher echelon groups, which also have IDs and define the entities contained therein.

Just as we break up higher echelon groups into individuals, we could similarly divide a building into components. Walls, beams, and columns are typical building components. When a simulation event affects a building component, we would communicate this change via a PDU. For example, a building damage event might create a breach hole in a wall component, cause superficial or minor structural damage to the wall, or completely obliterate the wall and remove it from the building. All of these changes could be communicated with parameters describing the nature of the event. It would be up to the federates to incorporate these parametric events into their building representations.

Participating federates must identify which building components should be affected by the events. Therefore, they need to modify their building formats to incorporate matching building component IDs. This would require the database modeling tool to generate and add these correlated IDs to all the correlated databases in an exercise. Some building formats do not support tagging building components with IDs. Or, components might have an ID format that is incompatible with others. In these cases, database formats would need to be augmented to include correlated component IDs.

Sometimes, the hierarchical structure of buildings will not match from federate to federate. For example, one simulation might represent an entire external wall as a single component, whereas another might break that external wall into multiple components based on the internal rooms it encloses. Changes to sub-components could not be correlated across these two simulations.

Similarly, some high-level simulations do not model individuals, representing them as part of a higher level echelon or group. These simulations do not process individual entity events. These disparities can lead to correlation and fair fight issues. A federation can mitigate these issues as long as it ensures the low-level simulations aggregate individual events appropriately.

Alternately, federations could force simulations that do not represent buildings with enough detail to ignore building component damage events. This would analogous to how high-level simulations group individuals. Unfortunately, this mitigation strategy would be less effective for buildings; simulations that do not break down buildings into components are not necessarily high level, low-fidelity simulations. Rather, they represent buildings in a format optimized for their typical use cases.

The alpha-numeric-event paradigm has been proven and is in widespread use in DIS/HLA simulations. Therefore, despite the issues discussed, it warrants further study and prototyping to assess feasibility for use in communicating building damage events.

Approach 2: Single Master Model
A "brute force" approach would be to compel all federation participants to operate on a single building model. By sharing this single model, all federates are guaranteed to have the same "ground truth." This single master model approach would be similar to the original approach of the AEC industry. (See Figure 8.)

Just as it was for AEC, this simplistic approach is impractical (and likely impossible) for many reasons. Simulations of different types must optimize building formats for their requirements or runtime performance will suffer. For example, a game engine should not be redesigned to operate with a format designed for a SAF, or the game itself would not meet its own runtime and performance requirements. In any case, it would be impossible to compel training solution providers to use a particular format. As stated earlier, the AEC industry has moved away from implementing BIM with a single master model for similar reasons. Therefore, we advise against this approach.

Approach 3: Standard Interchange Format
After years of research, experimentation, and industry-wide collaboration, the AEC industry has decided to pursue the communication via a standard interchange format. Their own research has shown this approach is less costly (Fuhrman, 2006). An ideal standard interchange format should be:

- Simple and structured logically. The data model should make sense and be understandable. It should be easy for simulation professionals to work with, which will help foster industry adoption.
- Consistent. Special cases should be minimized.
- Compact. Depending on the use case, a small memory footprint may be required due to hard disk constraints or network transfer requirements.
- Composable. To accommodate the many possible use cases we are targeting, the system must be flexible enough to include or exclude element types on a case by case basis.
- Complete. To address many possible use cases, the breadth of the data model must be comprehensive.
- A recognized standard. Standardization will also help accelerate industry adoption.

The AEC industry has adopted a BIM implementation called Industry Foundation Classes (IFC). IFC is a neutral and open specification not controlled by a single vendor or group of vendors. The specification is a registered ISO standard supported by 150 software applications (buildingSMART, 2013).

IFC defines a data model that contains several hundred entities, including building elements such as walls, geometry elements such as extruded solid areas, and basic building blocks such as Cartesian points. IFC specifies three file formats. The IFC-SPF format is an ASCII format in which each line consists of a single object record. IFC-XML is the XML variant of IFC-SPF, and IFC-ZIP is a ZIP compressed format of IFC-SPF.

IFC effectively covers the breadth of data required for building representation. However, the AEC designers did not design the file formats or the data model for compactness, efficient data transfer, or efficient export/import. The design emphasis for AEC is lossless, complete interchange; fast data transfer is a secondary concern at best.

For large simulation exercises, fast data transfer is vital. Therefore, IFC file formats are not a viable option for our domain. The Layered Terrain Format (LTF) and the associated Layered Synthetic Environment Runtime (LaSER) could be the answer. PEO STRI's OneTESS program originally developed the format and runtime services with very tight processing and on-disk requirements. Army Research Lab's Simulation and Training Technology Center (ARL STTC) then generalized the technology for reuse across live, virtual, and constructive domains. Because STTC intended LaSER and LTF for multiple domains across many programs, they took special care to ensure simplicity, consistency, composabilty, and compactness (Peele et al, 2011).

STTC is now pursuing recognition of LTF and LaSER as international standards. They are also optimizing the technologies for rapid network transfer using Google protocol buffers (Google, 2012). LTF and LaSER are evolving to meet all the aforementioned criteria. To ensure completeness, LTF developers should consider the breadth of IFC's data model.

Dynamic building events, once received, must be incorporated into federates' environments. We call this process assimilation: environment changes must be assimilated into all participating simulations. We will not discuss this problem in depth, but we will mention STTC's research involving the SHared Architecture for Dynamic Environments (SHADE). Through SHADE, STTC provides a set of tools that enable dynamic environment events. SHADE implemented assimilators for different simulation types, including OneSAF. Assimilators are custom-built for each type of simulation participating in an exercise; there is no way to avoid the assimilation process, since simulations use custom formats. Fortunately, SHADE offers a framework for assimilation and provides examples. Alternately, federates may decide to implement assimilation on their own.

OVERALL SOLUTION

So far, we have discussed the need for dynamic building events in UO, and we have discussed several disparate technologies that fill parts of the capability gap. Ultimately, we must bring these tools together to solve the static structure problem.

Figure 9 shows how a federation could leverage these technologies to achieve shared dynamic building events. Existing constructive simulations already share detonation events over distributed simulations (Figure 9, top left). A building damage model adapted from the analysis domain would receive this detonation event (Figure 9, bottom). In near real-time the model would calculate the resulting building and package the results in LTF or a similar, standardized format. The federation would then send the damaged building PDU to interested federates. Through assimilators, the participants would incorporate the damaged building into their runtime environments.

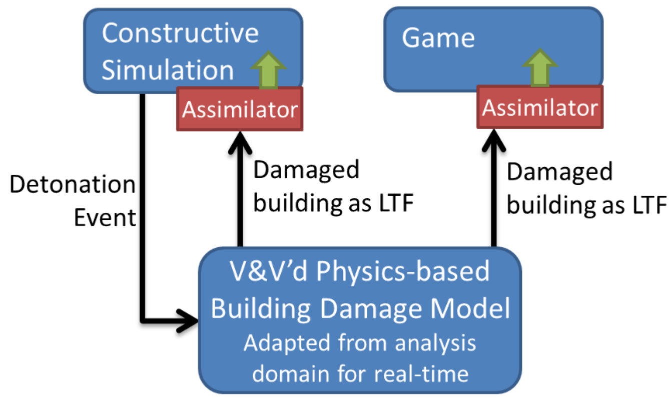

Figure 9. A building damage model, LTF, and assimilators work together to enable dynamic building events in a federation.

While achieving this vision on a large scale is ambitious, we've proven the utility and functionality of the individual parts. Much work remains to achieve the overall solution, and widespread adoption will be challenging. However, the potential gains in training capability will be well worth the effort.

CONCLUSIONS AND FUTURE RESEARCH

Training for urban operations is critical for today's warfighter. Dynamic building events, a significant part of UO, are impossible with today's training systems. The technical challenges involved in achieving dynamic buildings are substantial, but several promising technologies have emerged. With more research, we can advance these technologies and bring them together.

Though games and simulations have made significant technical strides toward real-time building modification, with very few exceptions these dynamic capabilities work in standalone mode. Game and simulation developers should explore ways of sharing these runtime environment changes as part of a larger simulation. To facilitate and standardize this sharing, we propose that developers consider communicating via a standard interface format.

STTC is pursuing LTF standardization. In addition to standardizing, we propose the data model be compared to IFC's. The BIM community has invested sizeable resources to ensure IFC's comprehensiveness, and we should leverage this investment.

Under the SHADE project, STTC successfully developed assimilators for incorporating environment changes from external sources. Game and simulation developers should consider these assimilators to increase their dynamic building capabilities.

In this era of tight DoD budgets, we cannot afford to re-implement solutions already created for other services and domains. Test and training communities should leverage the analysis community's significant investment into fast-running, verified and validated models. We should adapt these solutions, not re-invent them.

As we advance the technology of the piece parts, we shouldn't neglect the whole. We can begin by prototyping ways to bring enabling technologies together to address the overall static structure problem. This way, we will learn if our vision and implementation plans are flawed, need minor course corrections, or are right on track. The end result of these efforts will be warfighters better prepared for the urban battlefield.

ACKNOWLEDGEMENTS

The authors would like to thank the technical experts that provided feedback on this paper, including James Kainz and James Nichols of Applied Research Associates, Inc. and Chuck Campbell of Edgewise Technologies. We'd also like to thank Heather Harms for reviewing the paper.

REFERENCES

Applied Research Associates. (2006). *Modular Effectiveness Vulnerability Assessment (MEVA) (Version 7.0, Beta 1.0) Final Report: Software User's Manual*. Eglin AFB, FL: Scarborough, Eric, & Watts, David.

buildingSMART, (2013). *International home of openBIM*, Retrieved June 6, 2013, from http://www.buildingsmart-tech.org/

buildingSMART, (2013). *Industry Foundation Classes Release Specifications (IFC2x Edition 3)*. Retrieved June 6, 2013, from http://www.buildingsmart-tech.org/specifications/ifc-releases/summary

Fuhrman, Andy. (2006). OSCRE Update: The OSCRE Standard for Property E-Commerce. *American Institute of Architects Building Connections 3*

Google, (2012). *Protocol Buffers*. Retrieved June 6, 2013, from http://developers.google.com/protocol-buffers

Headquarters, Department of the Army (2002). *Training for Urban Operations, Training Circular (TC) 90-1*, Washington, DC

IEEE Computer Society (2012). *IEEE Standard for Distributed Interactive Simulation—Application Protocols, IEEE Std 1278.1-2012*, New York, NY

Peele Jr., Gregory, & Adkison, James, & de la Cruz, Julio, & Borkman, Steve. (2011). Building a Compact High-Fidelity Runtime Database Engine Layer by Layer. *IMAGE 2011 Conference*.

Smith, Dana K., & Tardif, Michael. (2009). *Building Information Modeling*, Hoboken, New Jersey: John Wiley & Sons, Inc.