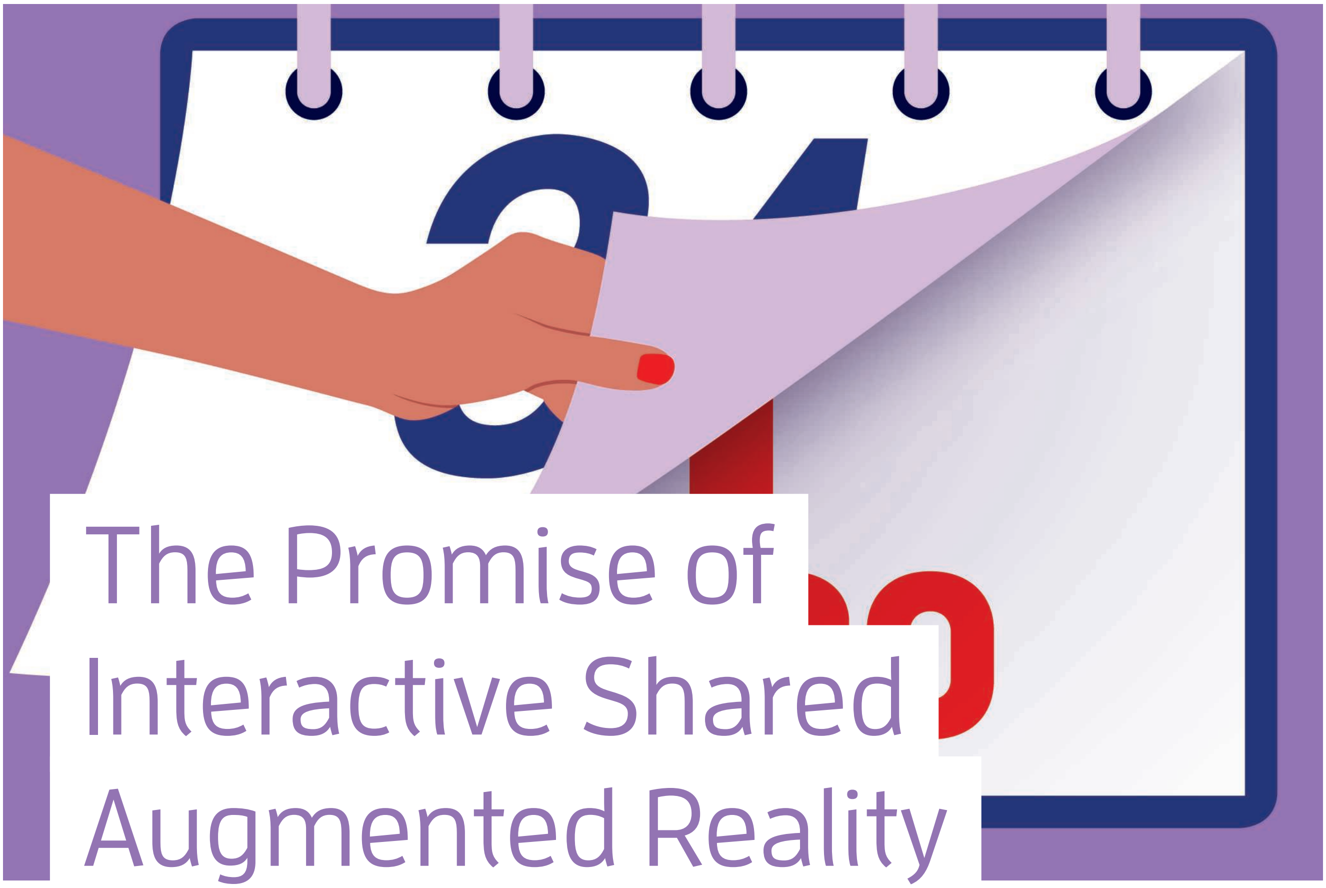

The Promise of Interactive Shared Augmented Reality

Loren E. Peitso and James Bret Michael, Naval Postgraduate School

Augmented reality (AR) is a game-changing technology that lets users see things they cannot otherwise see. Shared reality could be used, among other applications, to improve the safety of traffic systems. Despite current limitations, the future is bright for interactive shared AR.

A highly simplified description of augmented reality (AR) is that it merges the user's view of reality with information overlays from virtual reality (VR). Most of us intuitively consider this to be a vision-related input system that is primarily useful for entertainment, training, maintenance, or design tasks. For instance, technicians use AR to help them assemble spacecraft.[1] We think of simulations as virtual worlds kept separate from our reality. AR and simulations both have a tremendous amount of unrealized potential for integration into day-to-day activities. The things we can do with integrated AR simulations are not necessarily new themselves; they have been predicted and written about for decades as being just around the corner. How does that seemingly old story turn into an article about the future?

Simulations that are distributed among multiple participants have faced a technical challenge that has, so far, proven to be impossible to overcome: the lag in communications between each participating system. This lag and the nature of how the vast majority of simulations are computed conspire to limit each participating simulation such that it works with a purely local state that can never dynamically agree with all of the other

Digital Object Identifier 10.1109/MC.2019.2951981
Date of current version: 15 January 2020

linked simulation participants. When every participant provably has different states for everything computed (which we will cover later), this eliminates the possibility of guaranteeing agreement in the computation of positions, actions, and interactions among them. That makes a set of simple ideas and use cases that have been described for decades into a grand challenge locked behind a seemingly impenetrable limitation: the speed of electrons through a wire.

This limitation has been so effective that research in the area of distributed virtual environments is a faint shadow of what it was in the late 1990s, when the original promises of VR and simulations were made, and the first steps into AR were taken. Every project that attempted to scale across a nonlocal network ran into the perceived insurmountable issue of latency, which prevented the consistency in the state that was necessary to have stable and verifiable computations across the network when real-time, dynamically moving objects were allowed. Multiple textbooks cite the problem as a proven, hard limitation.[2,3]

To solve this problem and unlock the true potential of the marriage between VR/AR and simulation techniques, we must manage these limitations and adopt techniques from other branches of the simulations tree. As we escape the box [which is literally a single box (system) serving a single user for a personal VR/AR experience], we can also move beyond the bias that VR/AR is simply visualization for a human. This opens up a much larger and more powerful realm of operations, entering into the grand challenge territory: network-wide sharing of dynamically changing information describing bits of physical reality that are hidden from other participants who do not even need to be human.

AR AND SIMULATIONS BOTH HAVE A TREMENDOUS AMOUNT OF UNREALIZED POTENTIAL FOR INTEGRATION INTO DAY-TO-DAY ACTIVITIES.

Author Michael conducted research on automated vehicle-control and safety systems at the University of California, Berkeley, in the 1990s. That research focused primarily on fully automated driving (such as automation of all driving tasks) for dual-mode vehicles (that is, equipped for both manual and automated control) operating in a cooperative vice-autonomous manner when traveling in lanes restricted to automated driving—one of several concepts involved in what is known as an *automated highway system* (*AHS*). Some of the main goals of the National AHS Consortium, of which Berkeley was a member, were to improve the safety and comfort of travel on highways while maximizing the utilization of existing highway lanes (for example, throughput) and minimizing the impact of vehicle operation on the environment (namely, reduction in nitrogen oxide emissions).[4]

The Berkeley-based team's concept of an AHS is an example of level 4 (high) automation in the Society of Automotive Engineers International taxonomy of automated driving. The occupants of the vehicle are not expected or even able to retake control of the vehicle in an emergency situation (such as failure of a vehicle's automated driving system to maintain a safe following distance from downstream vehicles in the automated lanes or to avoid a deer that wanders around a barrier into a restricted lane) nor would they be permitted to do so by the system. The researchers working on this project demonstrated the technical feasibility of having vehicles travel in platoon formations under high-performance conditions (in particular, close intra- and interplatoon spacing of vehicles at free-flowing traffic speeds of approximately 30 m/s), but those conditions are beyond the abilities of even professional drivers (in terms of reaction time or maintaining situational awareness) to safely retake manual control and operate their vehicles in the automated lanes.[5]

As Shladover[6] points out, lower levels of vehicle automation in the taxonomy, such as those that incorporate adaptive cruise control and automated lane-keeping (steering) systems but rely on the human driver to retake control of these driving tasks in an emergency, are commercially available today for use on existing non-AHS roadways with mixed traffic, that is, with vehicles operated under partial automated control interspersed with vehicles under full manual (human) control. Let's look at how a distributed AR simulation system could be used in a mixed-traffic setting to overcome the element of surprise and issues of latency for humans and systems performing driving tasks. In this case, AR extends the reality of the vehicle operators on local roads and highways in ways that are intended to be immediately useful and increase overall traffic safety.

MOTIVATING EXAMPLE: THE DIFFICULT-TO-SEE BICYCLE

Bicycles often share the roadway with motor vehicles, as shown in Figure 1. Figure 2 depicts an unsolved problem for both human drivers and individual autonomous vehicles: the element of surprise in hazardous situations, such as a bicycle entering an intersection from the side while blocked from view by an obstacle. It is easy to imagine a truck or bus as the obstacle, but a fence, building, group of pedestrians, or tree can also create a visual barrier. When the bicycle enters the intersection from behind the obstacle, there may be a critically short time for the bicyclist, motorists, and/or self-driving automobiles to react to avoid a collision. The uncontrolled placement of real-world visual obstacles hides critical information from multiple participants at the intersection.

Whether the driver of vehicle A is human or an automated control system, the time to sense the environment of the intersection; make a decision; and activate the accelerator, brakes, and/or steering (specifically, reaction time) may be longer than that required to avoid a mishap. We believe that augmenting the automobile operator's individual reality into a collective shared one can dramatically reduce the occurrences of surprises and, in our scenario, potential mishaps.

SHARED REALITY

What is this shared reality? It is a form of AR in which the augmenting information comes from a VR that itself is a distributed simulation. The base environment from a data/map system is dynamically combined with information obtained from tracking the behaviors of real-world objects via the participants' sensor arrays. Each participating vehicle shares its view of the world with other local vehicles, and this is merged into a common and consistent shared representation of the local world. We consider the result a form of shared reality. In an individual vehicle, the operator has access to information from both onboard sensors and network sharing. The accurate and consistent creation of this shared reality is precluded by the network latency problems in current distributed simulations.

FIGURE 1. A bicycle crossing a busy multilane roadway intersection.

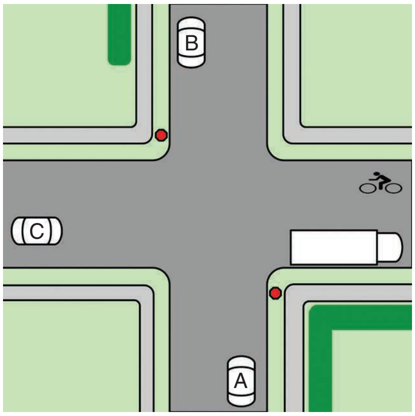

FIGURE 2. The crossing bicycle is hidden from vehicle A but visible to vehicles B and C.

Back in our difficult-to-see bicycle scenario, the shared reality is built by communications between vehicles on the road or even fixed sensors that may be part of a smart intersection-management system. Although vehicle A in Figure 2 has a significant portion of its sensor field masked by the truck and tall hedges, other sensor-equipped autonomous vehicles (B and C) have a clear line of sight and can not only detect the presence of the nonnetworked bicycle and rider but also accurately determine its relative position and velocity. When the bicycle's track information is passed along to other vehicles, the receiving vehicles' decision systems cross-check and integrate this information into their local simulation. This is possible because fixed information from the map/data system is fused with sensor readings of fixed local landmarks, such as traffic signal support posts and the parked truck. The common fixed-location information allows each vehicle in the network to determine high-confidence absolute position, track information for themselves, and integrate the relative track and position of the non-networked bicycle. Now, the vehicles' decision-making systems each have a shared reality that is larger and/or more precise than what is possible from own-platform systems alone.

This shared reality is projected in each automobile to augment the platform operators' AR scenes. If the driver of vehicle A is human, an AR bicycle warning graphic can be projected to represent the bicycle and rider hidden behind the truck, and it can be moved so that it follows along the same bearing angles as the actual bicycle. If vehicle A is operated as an autonomous system, what-if scenarios and decision processes for determining how to safely approach and navigate the intersection can progress by considering the bicycle's track, even without prior own-vehicle sensor confirmation. Either way, more relevant information presented to the operator generally enables globally better decisions, especially when the additional information unmasks otherwise hidden threats.

DETERMINISTIC AND REPEATABLE DISTRIBUTED REAL-TIME SIMULATION

How does the scenario described avoid the constraints that prevent the effective creation of shared reality with current techniques? The short answer involves finding a way to hide the communications latency whenever possible and having an efficient way to quickly update the participants during the remaining times when hiding the latency is not possible. Over the years, many parts of the solution to this problem have been attempted in isolation; however, none of them have solved enough of the issue to be selected for widespread use. Going back to the requirements and crafting a coherent system out of the inspirations of the past may finally provide a different result. Here, a mini deep dive is required to clarify these last statements.

NEWTON DOES NOT TELL US WHEN THE MATH MUST BE EVALUATED, BUT HE DOES IMPLY THAT WE MUST MAINTAIN A CHAIN OF CAUSALITY.

The shared reality is a distributed simulation system with hard real-time constraints. Our latency problem has been poked and prodded by simulation and VR domain researchers for years, and, unfortunately, they determined that communications latency categorically prevents both a dynamic and shared state across the network. Singhal and Zyda[2] formally describe the problem as the consistency-throughput tradeoff and provide an informal proof. The consistency part relates to the states at all nodes in the network agreeing at the same time. The throughput part says that when the state is allowed to frequently change, the communications delays necessarily cause delays in updating the state of all of the other nodes. The faster the dynamic changes, the worse the throughput-required part of the problem. Why the throughput portion goes bad so quickly is of interest to us to help avoid the problem in the future.

Whenever a node on the network broadcasts a change in something known only to it, the remainder of the network remains oblivious until that broadcast information event is received. This guarantees that each simulation node on the network has continued computing based on what is now some bit of stale, incorrect information. Given the traditional implementation, in which simulations have fixed time steps and the whole environment is updated synchronously, the time steps need to be short to avoid a whole host of technical problems. Computing many short time steps before that broadcast change arrives creates an intractable amount of work to recompute, and the higher the rate at which those changes enter the simulation network, the more divergent and intractable the problem becomes.

This has enabled a commonly held, but unfounded, belief that distributed simulations will always be somewhat different at each node and, therefore, not fully deterministic or repeatable as a system. It is accepted as inevitable that communications latency creates artifacts, limiting the utility of distributed simulation techniques to assist in applications that require

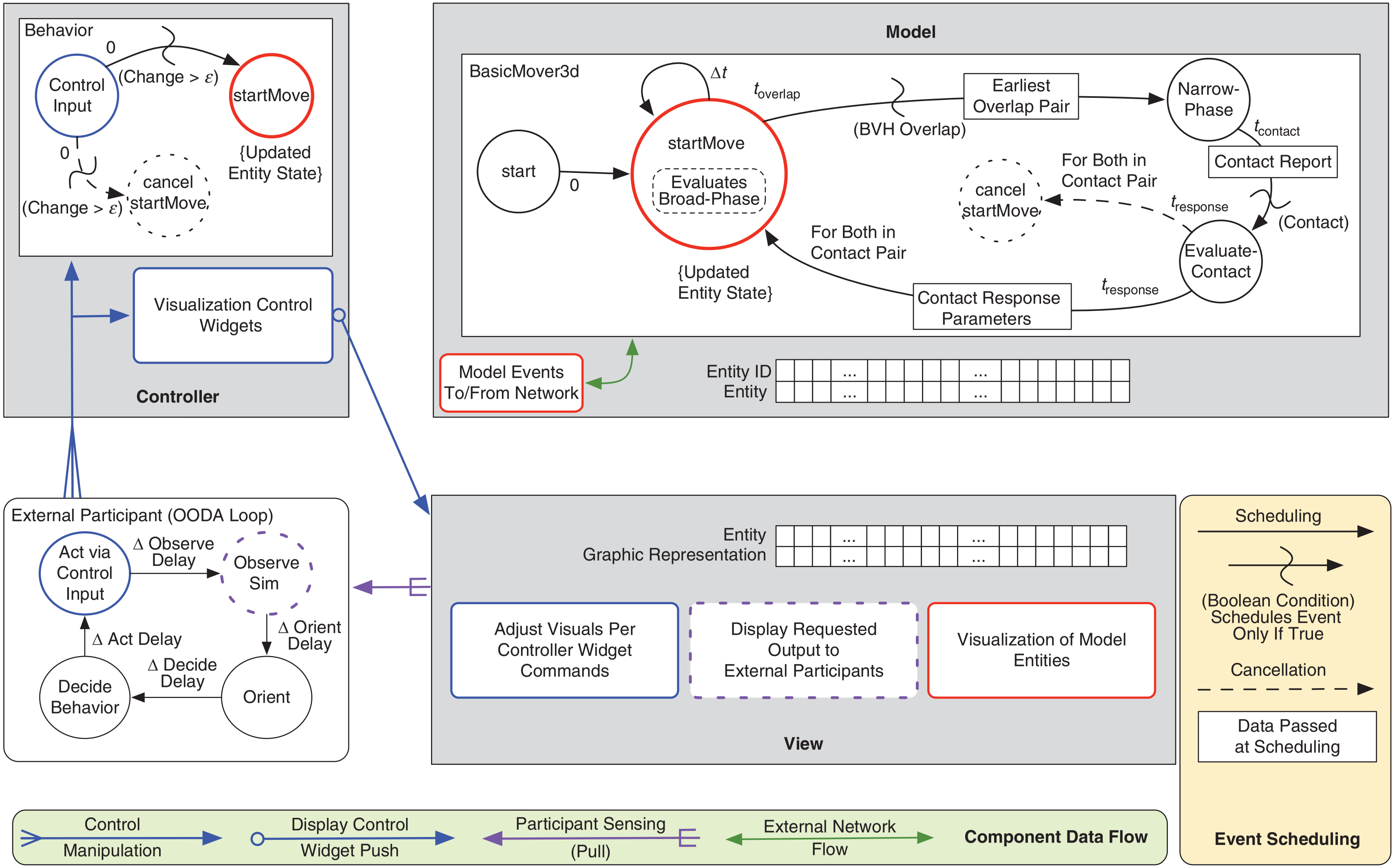

FIGURE 3. An architectural design for a network-distributed asynchronous discrete-event based physics simulation.[7] OODA: Observe Orient Decide Act; BVH: bounding volume hierarchy.

high reliability or are safety critical. However, we did not write this article only to run into a wall of "don't get to the future" due to latency. We will get to the future by cleverly rebuilding the networked simulation system to hide communications latency to the maximum extent practical and heal the divergence in the limited places where latency cannot be hidden. The latency-hiding technique allows a distributed simulation to efficiently manage an effectively consistent shared state, and this enables the real-time shared reality to be built from data that are verifiably accurate and reliable. A distributed simulation system, high-level architecture incorporating latency hiding is shown in Figure 3, and a prototype system implementation is in development at the Naval Postgraduate School.[7]

To enable latency hiding, we first need to sever the reliance on fixed short time steps, which can be achieved by asynchronously scheduling events on an as-needed basis driven by adaptive-step motion-evaluation methods. This kicks off a cascading sequence of simulation system design and technique choices that are best illustrated by the children's book *If You Give a Pig a Pancake*.[8] Each step of the cascade requires picking a component that enables extending the length of the required advancement intervals and not compromising accuracy or fidelity.

JUST-IN-TIME DELIVERY, SELF-HEALING, AND ENGINEERING TOLERANCE

Coupling adaptive-step numeric methods, data structures originally designed for geographic information systems research, and asynchronous discrete-event scheduling techniques from the analytical simulation domain enable leveraging Sir Isaac Newton's first law of motion: "Every body persists in its state of being at rest, or of uniform motion in a right line, unless it is compelled to change that state impressed thereon."[9] However, this is done in a novel manner: Newton does not tell us when the math must be evaluated, but he does imply that we must maintain a chain of causality.

If we optimistically precompute the expected outcome of interactions that will impress new forces on an object, we can transmit an event containing that result over the network so that it arrives just in time to each of the other distributed nodes in the network. We get latency hiding by precomputing and sending just early enough such that the transmission latency results in the just-in-time delivery. Events created by an action outside the simulation software constitute a form of surprise to the system, which we will deal with in a moment. Everything else that is not a surprise is manageable within this precomputed, just-in-time delivery paradigm. An added benefit is that, should network latency jitter create a late arrival despite these efforts, the distributed participants will calculate an update themselves and then compare the locally computed result with the late-arriving event when it occurs. This comparison provides the opportunity for a cross-check between systems, and if the state is not effectively consistent, we have evidence of a problem, whether a bug, sensor failure, or potentially malicious action. Here, the term *effectively consistent* describes a consistent state across all nodes of the network but allows short-duration, limited-magnitude deltas in state during surprise-resolution self-healing intervals.

Self-healing removes state divergence created by surprise-injected events that cannot be precomputed, and these occurrences are fully subject to the effects of latency. This is where fixed time-step simulations run into intractable rework problems, but the longer-duration events and richer data structures used to represent the motions of objects limit the rework to manageable levels. In many cases, rework is completed simply by truncating the end time of a single object's predicted future motion. In the general field of distributed asynchronous simulation, this recomputation is called *rollback*, and solving it in the general case is a messy and inefficient affair. By leveraging the laws of physics and integrating the real world, the time advancement of the system is tightly constrained, limiting the greatest threat that creates nasty rollback

SELF-HEALING REMOVES STATE DIVERGENCE CREATED BY SURPRISE-INJECTED EVENTS THAT CANNOT BE PRECOMPUTED, AND THESE OCCURRENCES ARE FULLY SUBJECT TO THE EFFECTS OF LATENCY.

situations. Having a motion-tracking data structure that can efficiently identify the minimum subset of objects that must be reworked is another key to avoiding unconstrained rollback, so much so that we believe the overall constraints are strong enough that this version should be called *rework* instead of *rollback*. (There are no claims that these techniques solve the general case of the rollback problem.)

Maintaining this healed, effectively consistent state does not violate the consistency-throughput tradeoff; we simply limit the effects of unavoidable latency into a small corner of the total design space and define the means to depart from that corner in an efficient manner. This is additionally supported by modeling the real physical world at the human scale, which is a sparsely populated space where objects tend to be far apart with respect to the size of the contact tolerances. This also helps limit the magnitude of changes with respect to how long and the rates at which divergences may grow, further preventing unconstrained rework requirements. Collectively, these constraints do not provide a universal simulation solution. Modeling the motions of snow crystals in the interior of an avalanche or the gas dispersion and mixing within the cylinder of an automobile engine are poor choices for this type of distributed simulation.

Another contributing enabling aspect is that effective consistency is not absolute binary consistency. Absolute binary consistency is problematic, especially when not every participating platform uses identical hardware. For navigating real-world objects at the human scale, precision or absolute matching in the 21st least significant binary digit is not strictly necessary. In the bicycle example, the critical threshold when considering contact between objects is, realistically, the thickness of automotive paint, approximately 4 mil, or 0.1 mm; we want our driving systems to calculate and execute with far larger safety buffers during planning and movement. Should we merely scratch the paint of the automobile and bicycle, we will not be happy, but we will not have transferred enough energy to have caused a catastrophic crash.

When using a comparison threshold that has its most significant digit in the eighth significant binary digit, numeric issues, such as binary round-off error, become far less critical as those effects are primarily felt in the low-significance digits. This also insulates the simulation algorithms from requiring true binary lockstep precision between participants to consider the state effectively consistent. This tolerance-based systemic consideration honors the fact that the dynamic physical world is not mathematically perfect and keeps with the truth of simulation models expressed by George Box: "Essentially, all models are wrong, but some are useful."[10] The more differing platforms that may participate in an effectively consistent and useful shared-reality simulation, the better.

Routine time-of-arrival artifacts in the networked computation state are eliminated by the state healing process, providing the basis for more effective reasoning and recording of event streams, which, in turn, can provide definitive audit trails of which vehicle knew what exactly when. A divergent state not healed by normal operation can be treated as an error signal, alerting other platforms either that they need to provide a safety buffer for a potential hardware failure or that there is an attempt to maliciously manipulate the shared state.

In this manner, a distributed network of vehicles operates in a collective environment that is deterministic in managing its shared data. The effectively consistent state across the

ABOUT THE AUTHORS

LOREN E. PEITSO is a senior lecturer in the Naval Postgraduate School's Department of Computer Science. His research interests are how to employ network distributed simulation as the glue between multiparticipant virtual reality, augmented reality, and command and control systems. Peitso received his M.S. in modeling, virtual environments, and simulation from the Naval Postgraduate School, Monterey, California. Contact him at lepeitso@nps.edu.

JAMES BRET MICHAEL is a professor in the Naval Postgraduate School's Computer Science and Electrical and Computer Engineering departments. His research interests are on the design, assurance, and sustainment of trusted artificial intelligence, distributed, and embedded systems. Michael received his Ph.D. in information technology from George Mason University, Fairfax, Virginia. He is a Senior Member of the IEEE. Contact him at bmichael@nps.edu.

networked participants provides a robust starting point for validation and verification of the system participants' software and interaction behaviors. Without an effectively consistent shared state, systematic and repeatable verification and validation of dynamic system-dependent, decision-making algorithms is impossible, and canned piecemeal situational tests are all that are left. Playback of recorded event streams can provide a DVR-like audit trail of which vehicle knew what information and when.

REALIZING THE PROMISE OF INTERACTIVE SHARED AR

Fulfilling the potential of interactive shared AR requires the adoption of an engineering approach to integrating the enabling technologies into working systems. Getting past engrained thinking about perceived technical showstoppers is a first step. For example, the described manner for managing the consistency-throughput tradeoff for interactive shared AR systems is strengthened by the fact that engineers have been able to develop distributed systems that provide for acceptable levels of consensus in the presence of failures, despite the theoretical implications for distributed computing of the Fischer, Lynch, and Paterson theorem.[11]

To be effective at improving traffic safety, engineers must treat interactive shared AR as part of complex sociotechnical systems. Nancy Leveson[12] argues that system safety is a control problem in which the control system enforces safety constraints through technical and social controls. Social controls, such as laws and policy, regarding the use of interactive shared AR must be considered. In addition, ethics considerations abound. For example, in the human-to-machine interactive shared AR used in safety-critical systems, does the human have a choice in what safety-related data he or she shares with others (machines or humans) at the roadway intersection? Similarly, in machine-to-machine interaction, is it acceptable for an engineer to design an interactive shared AR system that shares safety-related data only with users who are using technology developed by the engineer's company? These and many other sociotechnical issues remain to be explored.

The traffic safety example is the proverbial tip of the iceberg in the leveraging of shared AR. Wherever people or automated decision systems have to operate in the presence of information hidden and/or indirectly provided to some of the participants, shared-reality techniques can eliminate some of the difficulties in generating an appropriately synchronized environment and providing the augmented display layer to improve the decisions made. C

REFERENCES

1. E. Winick, "NASA is using HoloLens AR headsets to build its new spacecraft faster," *MIT Technology Rev.*, Oct. 9, 2018. [Online]. Available: https://www.technologyreview.com/s/612247/nasa-is-using-hololens-ar-headsets-to-build-its-new-spacecraft-faster/
2. S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*. New York: ACM Press, 1999.
3. A. Steed and M. Oliveira, *Networked Graphics*, 1st ed. Boston, MA: Morgan Kaufmann, 2009.
4. M. Novak, "The National Automated Highway System that almost was," *Smithsonian*, May 16, 2013. [Online]. Available: https://www.smithsonianmag.com/history/the-national-automated-highway-system-that-almost-was-63027245/
5. J. B. Michael, D. N. Godbole, J. Lygeros, and R. Sengupta, "Capacity analysis of traffic flow over a single-lane automated highway system," *Intell. Transp. Syst. J.*, vol. 4, nos. 1–2, pp. 49–80, 1998.
6. S. E. Shladover, "The truth about 'self-driving' cars," *Sci. Amer.*, vol. 314, no. 6, pp. 52–57, 2016.
7. L. Peitso and D. Brutzman, "Defeating lag in network-distributed physics simulations: An architecture supporting declarative network physics representation protocols," in *Proc. 23rd Int. ACM Conf. 3D Web Technology*, June 2018. doi: 10.1145/3208806.3208826.
8. L. Numeroff, *If You Give a Pig a Pancake*, 1st ed. New York: Harper Collins, 1998.
9. I. Newton, *Philosophiae Naturalis Principia Mathematica*, vol. 1, 1st ed., London: Royal Society, 1687 (Transl.: in A. Motte and J. Machin, Eds., *The Mathematical Principles of Natural Philosophy*. London: Royal Society, 1729, p. 19).
10. G. E. P. Box, "Science and statistics," *J. Amer. Stat. Assoc.*, vol. 71, no. 356, pp. 791–799, 1976.
11. M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, 1985.
12. N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: MIT Press, 2016.