

A Persistent Simulation Environment for Autonomous Systems

Benjamin N. Kelley¹, Ralph A. Williams²
Analytical Mechanics Associates
NASA Langley Research Center, Hampton, VA, 23681

Jason L. Holland,³ Otto C. Schnarr⁴
NASA Armstrong Flight Research Center, Edwards, CA, 93523

and

B. Danette Allen⁵
NASA Langley Research Center, Hampton, VA, 23681

The age of Autonomous Unmanned Aircraft Systems (AUAS) is creating new challenges for the accreditation and certification requiring new standards, policies and procedures that sanction whether a UAS is safe to fly. Establishing a basis for certification of autonomous systems via research into trust and trustworthiness is the focus of Autonomy Teaming and TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR), a new NASA Convergent Aeronautics Solution (CAS) project. Simulation Environments to test and evaluate AUAS decision making may be a low-cost solution to help certify that various AUAS systems are trustworthy enough to be allowed to fly in current general and commercial aviation airspace. NASA is working to build a peer-to-peer persistent simulation (P3 Sim) environment. The P3 Sim will be a Massively Multiplayer Online (MMO) environment where AUAS avatars can interact with a complex dynamic environment and each other. The focus of the effort is to provide AUAS researchers a low-cost intuitive testing environment that will aid training for and assessment of decisions made by autonomous systems such as AUAS. This presentation focuses on the design approach and challenges faced in development of the P3 Sim Environment is support of investigating trustworthiness of autonomous systems.

¹ Software Engineer, Crew Systems and Aviation Operations, MS 233.

² Scientist, Crew Systems and Aviation Operations, MS 233, AIAA Senior Member.

³ Software Engineer, Simulation Engineering Branch.

⁴ Electronics Engineer, Sensors and Systems Development Branch.

⁵ NASA Senior Technologist (ST) for Intelligent Flight Systems, MS 233, AIAA Senior Member.

I. Nomenclature

<i>Agent</i>	=	actor in the simulation
<i>AUAS</i>	=	Autonomous Unmanned Aerial System
<i>DDS</i>	=	Data Distribution Service
<i>h</i>	=	geocentric altitude above the oblate earth
<i>h_{gd}</i>	=	geodetic altitude of the position of the vehicle projected to the surface of the WGS-84 ellipsoid along the normal to that surface
<i>MMO</i>	=	Massively Multiplayer Online
<i>P2P</i>	=	Peer-to-peer
<i>R</i>	=	distance from the center of gravity of the vehicle to the center of the Earth in feet
<i>R_⊕</i>	=	radius of the Earth at the equator (6378137 meters)
<i>R_p</i>	=	radius of the Earth at the pole (6356752.3142 meters)
<i>R_{local}</i>	=	local radius of the Earth at the vehicle geocentric latitude in feet
<i>UAV</i>	=	Unmanned Aerial Vehicle
<i>UAM</i>	=	Urban Air Mobility
<i>xAI</i>	=	eXplainable AI
<i>λ</i>	=	geocentric latitude of the vehicle
<i>λ_{gc}</i>	=	geocentric latitude of the position of the vehicle projected to the surface of the WGS-84 ellipsoid along the normal to that surface
<i>λ_{gd}</i>	=	geodetic latitude of the position of the vehicle projected to the surface of the WGS-84 ellipsoid along the normal to that surface

II. Introduction

Modern Unmanned Aerial Vehicle (UAV) systems have paved the way for systems that demand autonomous capabilities. While UAV systems have been around for a long time, these legacy systems [1] lacked decision-making capability and thus required a human in the loop for most tasks. As technology advancements continue to drive towards cheaper, lighter and smaller computer systems and sensors, the benefits to unmanned and autonomous systems continue. We are at the cusp of a new era for Unmanned Aircraft Systems (UAS) as operational autonomous systems are realized in aeronautics.

The strategic priority of autonomous systems has risen over the past decade in large part due to military applications of UAS. It is one of the U.S. Air Force's top focus problems. As part of USAF vision for science and technology 2010-2030, they proposed as their second Grand Challenge "to explore, develop, and demonstrate technologies that enable current human-intensive functions to be replaced, in whole or in part, by more highly autonomous decision-making systems and technologies that permit reliable V&V to establish the needed trust in them. [2]" This is not only a challenge for the military but for civilian use of Autonomous UAS (AUAS).

The National Research Council (NRC) of the National Academy of Sciences identified the priority of autonomous research for civil aviation in their June 2004 "Autonomy Research for Civil Aviation: Toward a New Era of Flight" stating:

"The development and application of IA [Increasingly Autonomous] systems for civil aviation are proceeding at an accelerating pace, driven by the expectation that such systems will return significant benefits in terms of safety, reliability, efficiency, affordability, and previously unattainable mission capabilities. [3]"

More specific to persistent simulation, *Modeling and Simulation* was called out as one of the top priorities research recommendations to

"develop the theoretical basis and methodologies for using modeling and simulation to accelerate the development and maturation of advanced IA [Increasingly Autonomous] systems and aircraft. [3]"

NASA's Aeronautics Research Mission Directorate (ARMD) places a high priority on research in Increasingly Autonomous (IA) systems assurance. So much so, that "Assured Autonomy for Aviation Transformation" was named as a Strategic Thrust in the NASA ARMD strategic implementation plan [4] detailing NASA's vision of aeronautical research for the next twenty-five years. One of the tenets of the strategic implementation plan is research required to realize Urban Air Mobility (UAM) where UAM is defined as "a safe and efficient system for air passenger and cargo

transportation within an urban area, inclusive of small package delivery and other urban Unmanned Aerial Systems (UAS) services, which supports a mix of onboard/ground-piloted and increasingly autonomous operations. [5]” As we stand at the cusp of a new era, a major challenge we face to accredit and certify these autonomous systems for use in commercial aviation airspace.

NASA’s Autonomy Teaming and Trajectories for Complex Trusted Operational Reliability (ATTRACTOR) project aims to build a basis for certification of autonomous systems to determine an autonomous systems via an increased understanding of trust and trustworthiness of autonomous systems as well as the human-machine teams responsible for safety critical and time critical decision-making. The research includes autonomous systems in the context of mission planning and execution with analyzable trajectories, enabled by explainable artificial intelligence (eXplainable AI or XAI), efficient multi-agent interaction, and persistent modeling and simulation. The persistent modeling and simulation environment as a tool for researchers to test their systems in a dynamic environment that complements the UAM airspace.

III. Background

Simulations often play a critical role in the accreditation and certification process due to their potential low cost, ease of set up, rapid data collection, and ability to test multiple scenarios and repeat results. Historically, flight simulation focused on pilot/crew training or research on flight dynamics by modeling with a high fidelity the aircraft’s performance. These types of simulators are intended for a single pilot or crew interacting with a virtual environment, thus requiring limited or no connectivity across multiple systems. Alternately, multi-agent simulators have been developed to model air traffic patterns, such as Brahms simulation and development environment multi-agent system model [6] used in the evaluation of the Departure Sensitive Arrival Scheduling (DSAS) concept.

For each simulation, it is important to characterize the focus of the research the simulation will support. A key focus of our research is the interaction of multi-agents in a simulation environment verses actual flight dynamics of the models themselves. The multi-agent environment ATTRACTOR is aligned to the Massively Multiplayer Online (MMO) gaming community rather than the traditional high-fidelity simulations used for training and flight research. Gaming entertainment simulation systems have had multiplayer options for a long time. For example, the release of Microsoft Flight Simulator 98 included a multiplayer option. Utilizing the Unity™ Gaming Engine as a common simulation environment and open standards such as Object Management Group (OMG) Data Distribution Service (DDS), the ATTRACTOR project’s persistent simulation environment provides a flexible, scalable, and dynamic environment that researchers can utilize to test autonomy-enabling algorithms and methods in shared simulation environment.

The Autonomy Incubator (AI) at NASA Langley developed an autonomy framework based on the OMG DDS publisher-subscriber architecture. This framework, named AEON (Autonomous Entity Operations Network) [7], has been vital to the success and rapid development of work performed in the Autonomy Incubator. The AEON framework proved successful for real-time autonomous flight, the need for a simulated test and evaluation environment was identified as a must-have capability for follow-on efforts. If an AI member is working on a new trajectory generation algorithm, for instance, the only way to perform an end-to-end component test is to perform the test on a real vehicle in the lab. Although testing on a real vehicle is vital, a simulation environment to safely evaluate and improve software capabilities in advance of flight tests would facilitate a more dynamic and agile approach to testing new components in the AEON framework.

IV. Design Approaches

When designing the initial network communication architecture for the ATTRACTOR simulation environment, it was prudent to consider approaches used by other past and current simulation and MMO environments. While each approach has advantages and disadvantages, priority was given to design approaches that met the project requirements of high availability, serves world state to agents in a timely manner, and requires little to low maintenance by administrators.

A. Client-Server

The prevailing MMO network architecture design is a client-server approach. In this design the client layer communicates with a centralized server that manages player interactions, world state, etc. This design simplifies the client layer communication as it has a single channel to send and receive information from and does not need to manage multiple communication connections. For this design to scale, the single centralized server can be replaced with a load balancing server farm. This, however, adds the burden of managing and maintaining the server farm and demanding that the servers keep a synchronized player and world state. In this design approach, a server/server farm

act as a single point of failure. If it goes down for any reason (i.e., power outage, network congestion, scheduled maintenance, etc.) the players are unable to participate in the game. This added burden when scaling a client-server design can be prohibitive for smaller development teams.

ATTRACTOR’s simulation architecture avoids the strict client-server approach because of these scalability and resource limitations and adopting a more flexible and dynamic architecture utilizing peer-to-peer design. A traditional client-server application does not allow for the flexibility required in the simulation. The simulation, for instance, should be self-contained –able to execute in a single-player fashion without any network connectivity at all.

B. Peer-to-Peer

An alternative network design is a headless system where players communicate directly with one another over a peer-to-peer (P2P) network. Unlike a client-server design, clients must synchronize and manage the world state amongst themselves. This added logic can make the client application significantly more complex but eliminates the single point of failure of a centralized server allowing for players to join, leave and interact more freely. The distributed nature of a P2P network helps facilitate the scalability of the system. However, since each peer must manage multiple channels of communication, individual peer bandwidth becomes a concern. Unless a multicast technology is incorporated into the P2P communication, the bandwidth used by each peer grows as the number of participants increase. Another complexity of a strict P2P network is where the world state persists if there are no participating peers.

The ATTRACTOR simulation environment utilizes components of a P2P architecture because offers the flexibility required. A single instance of a simulation can execute alone or with participating peers with little difference.

C. Hybrid

A strict P2P approach may not be possible in many cases due to the complexities surrounding discovery of peers in a P2P network. Approaches that rely on multicasting may not be a viable option depending on network requirements of the client application and the limitations of the available network. One hybrid approach is to have a simple client-server network which helps facilitate peers in discovering one another.

ATTRACTOR’s simulation design hybridizes a traditional server and P2P peer into a persistence server. This persistence servers monitor the peers’ communication and keeps a record of the world state. When a new peer joins the network, the world state can be obtained from another peer if available or a persistence server.

V. Current State of Persistent Simulation Environment

The current state of the Persistent Simulation environment consists of a working Unity™ environment integrated with RTI™ DDS capability. The DDS Services provide agent announcements and agents’ current state such as position, orientation, and velocity. Work has also been accomplished to establish connectivity between centers, where agents from each center interact in the same environment, has proven DDS is sufficient for multi-center interaction in the same environment.

An extensible “mod-ing” pipeline for adding additional functionality and agent models is being designed. This will provide researchers with the tools that they need to build their own agent packages to “plug and play” in the Persistent Simulation. The initial simulation space modeled in the Unity™ environment is the Autonomy Incubator Complex at NASA LaRC as seen in Figure 1. Proof of concept for integration of external systems also has been demonstrated using the Vicon™ motion capture system at NASA LaRC.

A. ATTRACTOR Framework

The ATTRACTOR simulation framework is designed to achieve the following high-level goals:

- Platform agnosticism
- A high level of modularity with loose coupling to achieve reusability and flexibility
- Abstraction of network details and boilerplate from researchers and third parties
- Distributed communications that enable software to run anywhere
- Interoperability via well-defined communication interface based on standards
- Easily extensible
- Interoperability with existing AEON software

Although the Unity engine provides a multiplayer application API, we choose not to develop with it because it utilizes a traditional client-server design approach. Instead, we developed the Unity™ application acting as standalone single-player game. The ATTRACTOR Framework manages the network communication via the RTI™ Connex

DDS middleware. Each standalone single-player simulation upon startup, uses the DDS middleware to a) announce its entry to the simulation b) discover other agents in the simulation and c) publish its agent state and other sensor data.

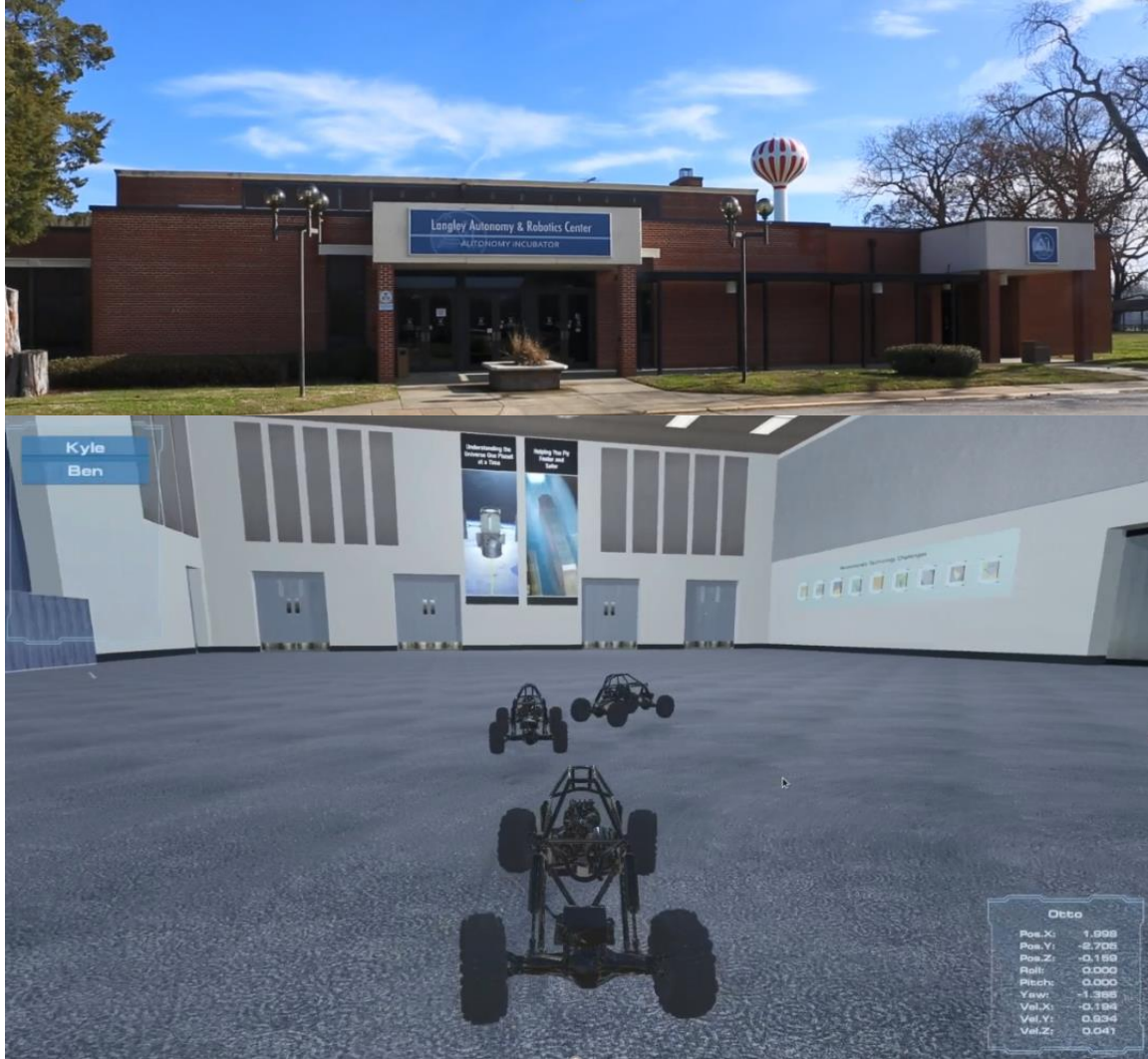


Figure 1: Real (top) and simulated (bottom) Autonomy Incubator Complex at NASA LaRC

B. DDS Communication

By utilizing DDS middleware, we can achieve the P2P connectivity we desire while abstracting much of the low-level communication requirements from the developers. The RTI™ middleware by default uses UDP multicast to facilitate peer discover and then proceeds to communicate peer data via UDP unicast. The middleware is flexible and offers configuration options for using TCP if UDP is unavailable on the current network and allowing to use multicast for data transmission to help reduce overall network bandwidth. The flexibility of RTI's implementation of the DDS standard allows us to take advantage of having a local network configuration where UDP multicast is available to us but allows us to easily switch to using TCP over a larger WAN when that need arises.

As briefly stated in the previous section, the ATTRACTOR simulation is designed using a hybrid architecture. All the network communication is P2P, but there are persistence servers on the network helping to keep the world state if there are no participating agents on the network as shown in Figure 2. Figure 3 shows the proposed communication pipeline for vehicle commands. By providing a uniform command API via DDS, a controlling piece of software such as a Navigation Controller should be able to easily switch between sending commands to a simulated vehicle and a DDS interface to a real vehicle.

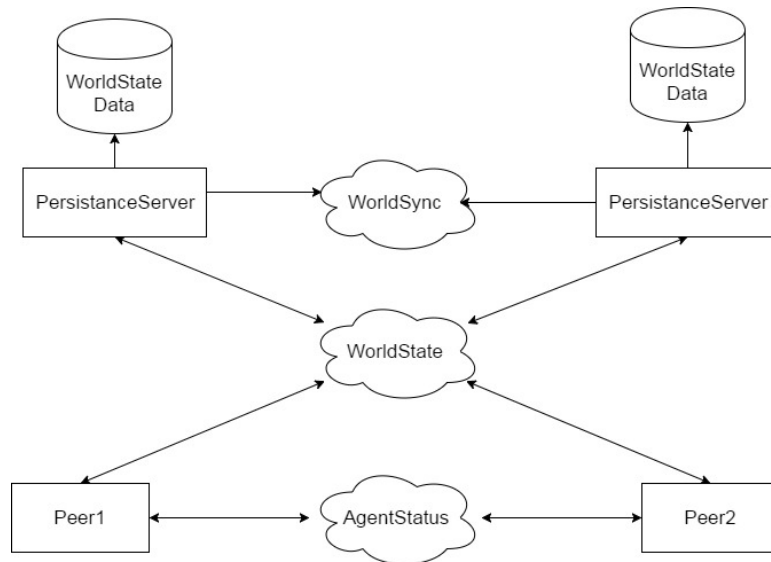


Figure 2: High Level Communication

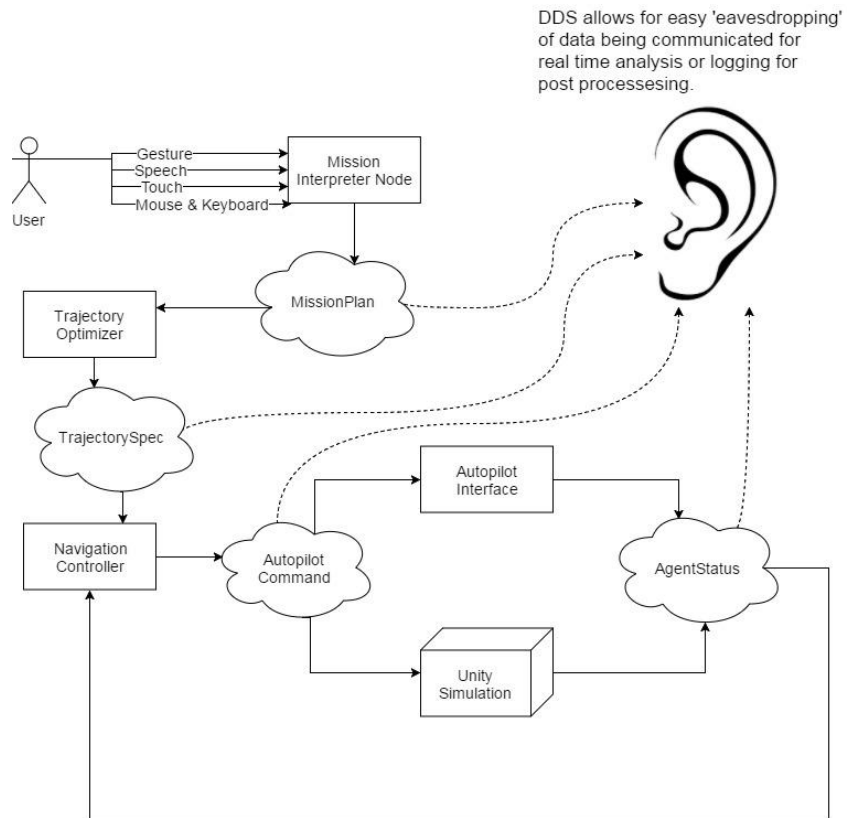


Figure 3: DDS based command API in the Unity Simulation

C. Mixed Reality

Mixed Reality (MR) is a spectrum between reality and virtual reality that includes augmented reality [8]. It mixes virtual objects and real world using systems such as traditional displays, head-mounted display (HMD) or standard aircraft HUD (heads-up display), to place real world information into a simulation such as plotting actual flight data to an agent in a simulation.

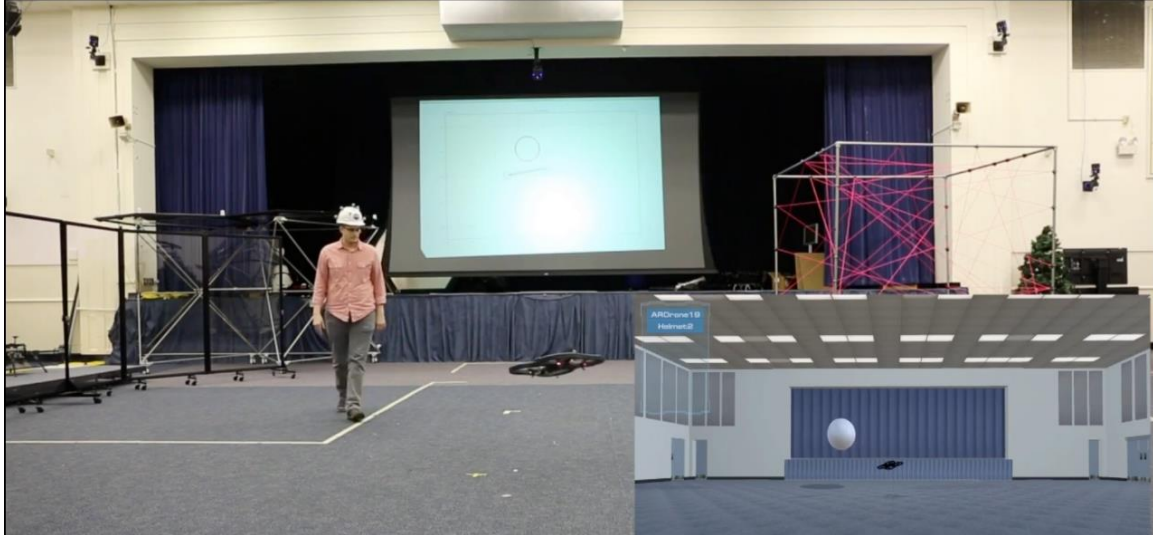


Figure 4: Augmented Virtuality (aka Mixed Reality)

ATTRACTOR's use of mixed reality encompasses two major goals:

1. incorporating real world vehicles and data or high-fidelity simulation data in a simulated world to test new AI agents and
2. understanding their autonomous performance and understanding why autonomous agents are making the decisions they are making (i.e., explainability).

The integration of real world vehicle data into a virtual world is an example of Augmented Virtuality (AV), a subcategory of Mixed Reality. Figure 4 shows an example of augmented virtuality, where real positional data of objects tracked in the environment are displayed in the virtual environment.

Another major benefit of a virtual environment is the observation platform. In the real world, observations occur live. In simulation, events can be replayed. In reality during an event, everything one wants to observe is only observed from their current point of view. In simulation during an event replay, one can change the orientation or position of observation. Figure 5 shows the virtual observability of real world objects in the simulation.

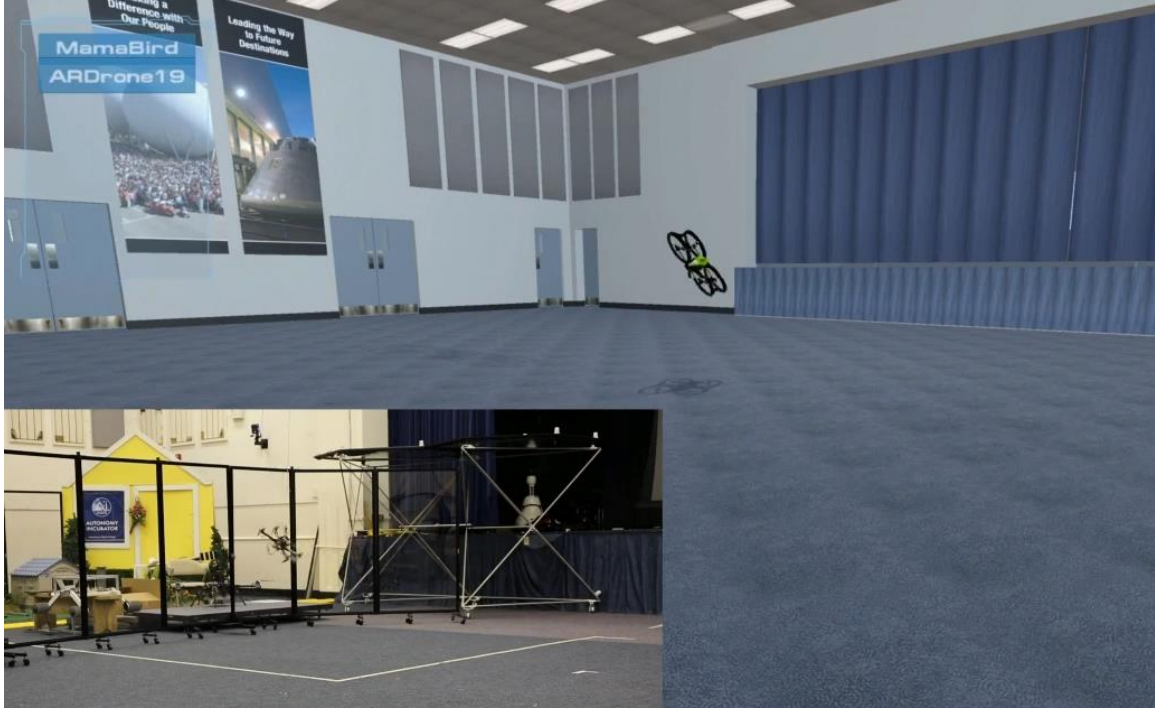


Figure 5: Virtual observation of an anomalous event

D. Distributed Simulation

One of the requirements for the persistent simulation environment is the capability to act as a distributed simulation for multiple research centers. That is, each center is able to see and interact with agents running at other centers. The ATTRACTOR team has demonstrated distributed capability by cross-center participation with participants from both Langley and Armstrong Flight Research Center. This was achieved by first establishing a layer 2 Virtual Private Network between the two centers. The team launched agents from their respective centers and interacted in an environment that models the Langley Autonomy Incubator Complex. Each team member created an instance of an agent in their environment and launched their stand-alone application. They were then able to observe other agents in the simulation participating from the other centers as shown in Figure 1.

This demonstration proved that the ATTRACTOR framework’s DDS middleware is capable of displaying multi-agents with no noticeable jitter and latency. It also demonstrated that the hybrid peer-to-peer architecture can run in standalone mode or multiplayer mode requiring no intervention from the user. The distributed capability is a key requirement for providing a persistent simulation environment for researchers.

VI. Future Work

A. Small verses Large Scale Simulation Environment

Several considerations must be made when creating “worlds” for the persistent simulation environment. Given that an agent could travel across a continent or around the globe in a sufficiently long simulation run, one consideration is how the earth is modeled. The initial simulation space modeled in the Unity environment was a representation of the Autonomy Incubator Complex at Langley. The site is an enclosed facility where experiments are performed on small AUAS. A flat earth projection is sufficient to properly simulate coordinates of agents in this type of environment. The terrain is considered geometrically as a horizontal plane. However, when working in a large-scale environment, developers need to be aware the earth is not flat nor a sphere, but rather an ellipsoid.

The map datum is an important role to define where entities actually reside in space. A datum is a reference system that approximates a point on the earth (Figure 6). Current simulation will continue to use a flat Earth projection for larger environments. However, in the future work will be accomplished to employ a WGS-84 Geodetic coordinate system for mapping projections on the earth surface. Even though WGS-84 is a fairly common map datum, not all map projections use this geodetic coordinate system. External agents may use geocentric depending on their use case, that is positional data is referencing the earth as a sphere. Projection of maps may employ other coordinate systems. Latitude errors can arise integrating systems that employ map projections due to failing to convert from a geocentric coordinate system to a geodetic due to the fact the initial point for angle of measure. For air to air engagements, geodetic values for latitude and altitude. External systems that integrate into the Persistent Simulation environment must consider if their coordinate system requires conversion prior to publishing their coordinates once a standard has been established.

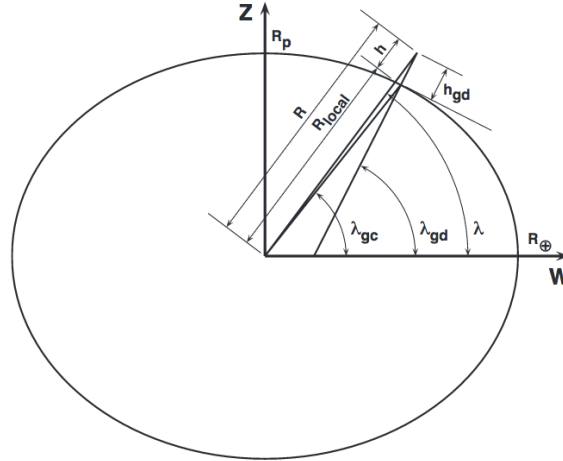


Figure 6: Geocentric and geodetic coordinate systems on Oblate Surface [9]

B. Development of Data Analytics between agents and environment

One benefit of a gaming engine such as Unity™ for simulation is the rich Application Program Interface (API) to support interactions between agents and agents with the environment. As a tool for researchers, work is underway to provide data analytics to capture these interactions for post experiment analysis and playback. One option already implemented is the use of the RTI™ DDS record capability to capture and agent's state. Future capability will include an observer agent that will capture agent to agent and agent-to-environment interactions utilizing the Unity APIs and publish the information via the DDS middleware.

C. Feasibility of VR/AR systems for use in analysis of behaviors of Autonomous Systems

Multiple groups across a large geographic area are supporting ATTRACTOR. As previously written; ATTRACTOR is able to include virtual agents, as well as real flight or high-fidelity simulation data. An area of interest for future work includes the increased use of augmented reality devices such as the Oculus™ or HoloLens™. An initial effort to show effectiveness of this approach for mission planning and execution is shown in Figure 7.

Through AR we are able view the simulation environment and how it overlays into a real environment with locally piloted or autonomous vehicles. The locally operated vehicle can feed back its position information to the ATTRACTOR simulation. The remote simulation data can be then taken into account by a local UAV operator in which case the remote algorithm can be interacted with a local real UAV. Alternatively, the remote or simulated UAV information can be fed into the local UAV's autopilot. In this case, a real UAVs autopilot can be evaluated and local operators are able to observe the interaction of the real local UAV with the virtual remote algorithm or UAV.

In either case the use of VR/AR systems allows for a much more immersive experience of observation. Local video or simulation screen capture only provides a single frame of observation (per camera). Through these mixed reality situations, the observer can change their observation orientation and positions. In the case of replays, speed can be adjusted down to a frame-by-frame playback. The use of AR/VR allows for observation from an optimal position of where a decision or change was made. A more advantageous position and immersive feel will allow for a more intuitive understanding of decisions made by the autonomous system and increase explainability.

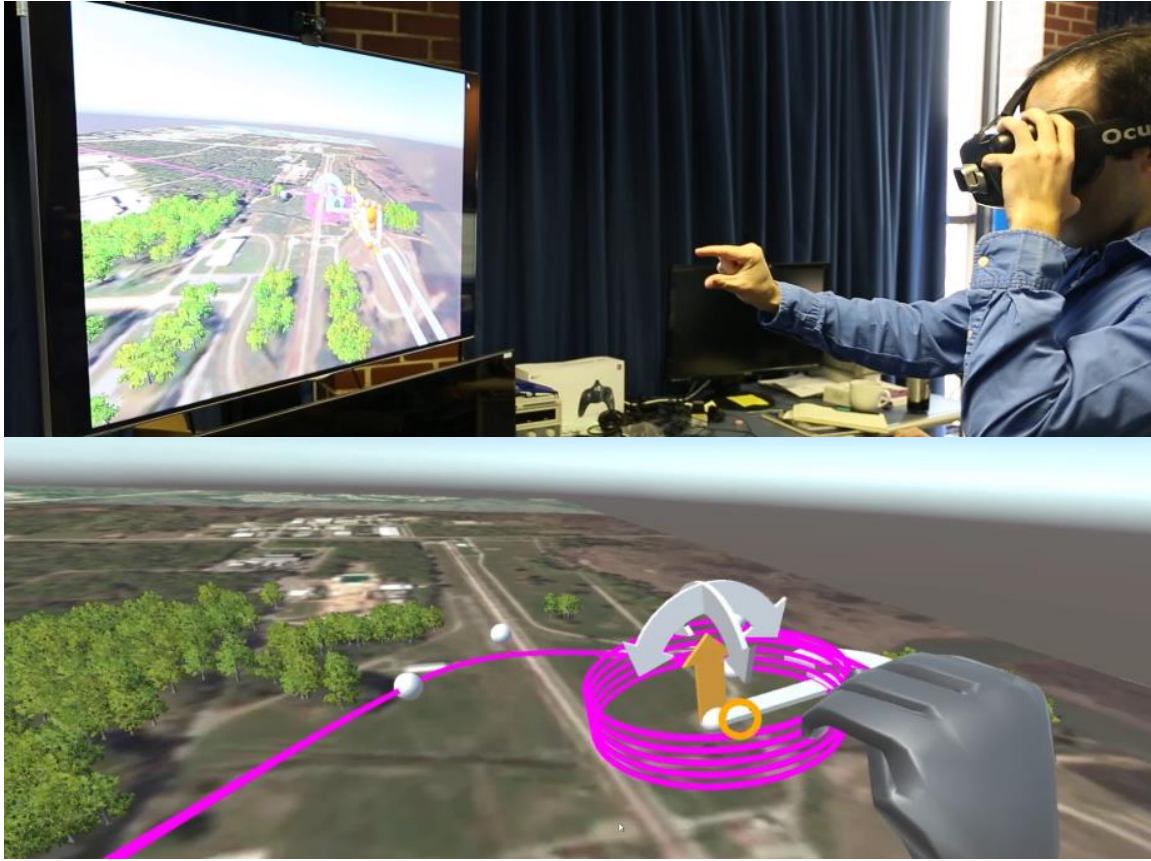


Figure 7: Immersive VR for Mission Planning and Execution

VII. Conclusions

The Unity™ Environment may seem an unusual choice for a flight simulation; however, it is an ideal choice to quickly setup an environment to use when analyzing interaction of multiple agents given its industry use as massively multiplayer online gaming environment and its rich agent interaction APIs. The project's approach to use commercial off the shelf (COTS) technologies has reduced cost and time to implement a persistent simulation environment. A modeling and simulation tool such as this is important to researchers because it allows them to characterize how an autonomous system will perform in a dynamic environment prior to costly flight test experiments.

The use of DDS in the AEON Framework and ATTRACTOR simulation environment provides a means to extend the system with reliance on the Unity Engine demonstrated with the example of augmented virtuality to integrate real world hardware systems in the virtual environment. There is still progress to be made to provide a usable tool for researchers to simulate their models in a persistent simulation environment. However, the team has shown it is feasible that MMO paradigms can a useful approach in research as it pertains to persistent modeling and simulation in an UAM environment. They have also shown that is feasible to create a hardware-in-the-loop test and evaluation environment that can interact with other agents and contingencies by virtual means, thus lowering risks of damage to autonomous system assets since the events are not a real-world physical interaction, rather a virtual experiment of a real world system.

VIII. Acknowledgements

The ATTRACTOR Software Team would like to thank our co-designers and co-developers. Kyle McQuarry, Matt Vaughan, and Andrew Miloslavsky have contributed creatively and constructively and we could not have gotten this far without them. We also thank Dr. Natalia Alexandrov, co-Principal Investigator of the ATTRACTOR project at NASA Langley Research Center (LaRC), for her thoughts and insight on researcher requirements. Projects like ATTRACTOR provide us (software engineers) with the opportunity to innovate and create systems that support research. The work accomplished in the ATTRACTOR project will aid in building a basis of certification for autonomy as we introduce AUAS systems in the National Airspace.

References

- [1] <http://www.nationalmuseum.af.mil/Visit/Museum-Exhibits/Fact-Sheets/Display/Article/198095/kettering-aerial-torpedo-bug/> [retrieved 24 April 2018]
- [2] Office of the US Air Force Chief Scientist “Technology horizons: a vision for Air Force science and technology 2010-2030” pp. 123
- [3] “Autonomy Research for Civil Aviation: Toward a New Era of Flight”, Committee on Autonomy Research for Civil Aviation, Aeronautics and Space Engineering Board, Division on Engineering and Physical Sciences, NATIONAL RESEARCH COUNCIL, <https://www.nap.edu/read/18815/>, 2014.
- [4] “NASA Aeronautics Strategic Implementation Plan” 2017 pp. 5
- [5] <https://www.nasa.gov/aero/nasa-embraces-urban-air-mobility-2017> [retrieved 24 April 2018]
- [6] Neha Rungta, Eric G. Mercer, Franco Raimondi, Bjorn C. Krantz, Richard Stocker, and Andrew Wallace “Modeling Complex Air Traffic Management Systems” 2017 pp. 4
- [7] Charles D. Cross et al. “Towards an Open, Distributed Software Architecture for UxS Operations” 2016 pp. 6
- [8] P. Milgram and A. F. Kishino (1994). “Taxonomy of Mixed Reality Visual Displays”. IEICE Transactions on Information and Systems. pp. 1321–1329.
- [9] Robert Clarke, Louis Lintreuer, and Cathrine Bahm, “Documenting the NASA Armstrong Flight Research Center Oblate Earth Simulation Equations of Motion” 2016 pp. 14