ARL-TN-0830 ● June 2017

US Army Research Laboratory

Outcomes from the First Wingman Software-in-the-Loop Integration Event: January 2017

by Kristin E Schaefer, Ralph W Brewer, E Ray Pursel, Anthony Zimmermann, Eduardo Cerame, and Keith Briggs

NOTICES

Disclaimers

ARL-TN-0830 ● JUNE 2017

US Army Research Laboratory

Outcomes from the First Wingman Software-in-the-Loop Integration Event: January 2017

by Kristin E Schaefer
Human Research and Engineering Directorate, ARL

Ralph W Brewer
Vehicle Technology Directorate, ARL

E Ray Pursel
Naval Surface Warfare Center, Dahlgren Division, Dahlgren, VA

Anthony Zimmermann
DCS Corp, Alexandria, VA

Eduardo Cerame and Keith Briggs
US Army TARDEC, Warren, MI

REPORT DOCUMENTATION PAGE						*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* June 2017	**2. REPORT TYPE** Technical Note	**3. DATES COVERED (From - To)** January 2017–September 2017
4. TITLE AND SUBTITLE Outcomes from the First Wingman Software-in-the-Loop Integration Event: January 2017		**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Kristin E Schaefer, Ralph W Brewer, E Ray Pursel, Anthony Zimmermann, Eduardo Cerame, and Keith Briggs		**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-HRF-D Aberdeen Proving Ground MD 21005-5425		**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TN-0830
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

During 27–29 January 2017, the US Army Research Laboratory (ARL) hosted a software integration event to build a working software-in-the-loop (SIL) simulation environment in support of the US Army Tank Automotive Research Development and Engineering Center Wingman program. Wingman focuses on human-machine collaboration, decision making, human-machine combat teaming, and advanced manned-unmanned teaming operations. Its primary operational objective is to effectively project lethality for first contact with enemy forces, enabling friendly freedom of maneuver. Developing a joint SIL will allow for rapid prototyping, software development, and early assessment by joint services and team members. Developments in the SIL can be directly implemented on real-world vehicles. In addition, this SIL will also provide a simulation test bed for ARL to assess human interaction with manned-unmanned teams during periods without real-world field tests. Outcomes of joint research efforts will support the design of a robotic system user interface and enhance communication among manned-unmanned team members, which are critical to achieve Training and Doctrine Command 6+1 required capabilities for robotics and autonomous systems.

15. SUBJECT TERMS

human-robot interaction, robotic wingman, aided target recognition, unmanned ground vehicle, remote weapon system

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Kristin E Schaefer
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	32	**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified			410-278-5972

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Summary

During 27–29 January 2017, representatives from the US Army Research Laboratory (ARL), US Army Tank Automotive Research Development and Engineering Center (TARDEC), Naval Surface Warfare Center Dahlgren Division, and DCS Corp worked together to integrate a software-in-the-loop (SIL) simulation environment in support of the US Army TARDEC Wingman program. A list of attendees is provided in Appendix A. The purpose of this report is to describe the problems, limitations, and outcomes of integrating actual real-world autonomous vehicle software for manned-unmanned teaming into a simulation environment. Developing this joint SIL will allow for rapid prototyping, advancement of software development, and early assessment and testing by joint services and team members. Developments in the SIL can then be directly implemented on real-world vehicles for real-world testing. In addition, this SIL will also provide a simulation test bed for ARL to assess human interaction with manned-unmanned teams during periods without real-world field tests. Outcomes of joint research efforts will support the design of a robotic system user interface and enhance communication among manned-unmanned team members, which are critical to achieve Training and Doctrine Command 6+1 required capabilities for robotics and autonomous systems. This report includes an overview of the Wingman program, highlighting the development of the simulation SIL. It is followed by a general description of the software included in the SIL and general outcomes of the integration event. While this report only documents the first steps of developing the SIL, we were able to demonstrate that all the software components and 2 simulation environments were able to work together, making this a viable working SIL.

2. Introduction to Wingman

The US Army Tank Automotive Research Development and Engineering Center (TARDEC) Wingman program began in fiscal year (FY) 2014 to provide robotic technological advances and experimentation to increase the autonomous capabilities of mounted and unmanned combat support vehicles. At present, Wingman includes a single manned vehicle (with a 5-man crew: Manned Vehicle Driver, Manned Vehicle Gunner, Commander, Robotic Vehicle Operator, and Robotic Vehicle Gunner) working together with a single unmanned robotic vehicle operating in a joint gunnery task (Fig. 1). However, future advancements from this program foresee the single manned vehicle working cooperatively with multiple unmanned vehicles. A major goal of this program is to advance manned-unmanned teaming initiatives by iteratively defining and decreasing the gap between autonomous vehicle control and required level of human interaction.

Platform weapon proficiency evaluations are conducted on live-fire ranges following the guidelines outlined in Training Circular (TC) 3-20.31 for manned vehicle crews. The TC 3-20.31 "provides a training strategy for crews to attain direct fire weapon proficiency by delivering fire on target(s) using the target itself as a point of aim for either the weapon system or having the leader control fires. It provides training principles and techniques for use by the crew to gain proficiency in engaging and destroying threats efficiently in any operational environment. It describes the crew training program, range operations, the engagement process range requirements and scenario development, and the fundamental crew tables including the qualification standards for all direct fire weapon systems" (US Army Training and Doctrine Command 2015).

The TC currently does not contain guidelines for assessing a joint manned-unmanned team. The mounted machine gun platform using a remote weapon station (RWS) is the closest evaluation, so RWS assessment guidelines were used in the absence of specific doctrine. The strategy of all weapon systems follows a crawl, walk, run method of training and evaluation. Successive gunnery tables enable crews to build upon their skills, culminating with a Table VI qualification. As described in the TC, Tables III–VI outline the task and posture of the firing vehicle, type and number of targets, and the ammunition or weapon type for each table. Table II is a cyber live-fire event conducted within simulation. It uses the same guidelines as Table IV, except it is done in simulation. All other table qualifications are conducted live on the range using the weaponized platforms against stationary or moving targets placed in a tactical array, during day and limited visibility conditions from both offensive and defensive postures.

Fig. 1 Unmanned robotic high-mobility multipurpose wheeled vehicle (HMMWV) with remote weapon station payload executing a Table VI engagement

3. Software-in-the-Loop Simulation: Software Description

Development of a Wingman software-in-the-loop (SIL) simulation environment will allow all members of the Wingman team to access the real-world vehicle software within a simulated environment. The goal is to advance the software, test integration, and assess the crew interactions between the 5-man team, simulated hardware, and the software in a safe and cost-effective environment. More specifically, it will provide a joint capability to advance the individual elements of the real-world system software as well as test the integration of these elements, including the Robotic Technology Kernel (RTK) for autonomous mobility (developed by TARDEC), the Autonomous Remote Engagement System (ARES) supporting the autonomous targeting and weapons systems control (developed by the Naval Surface Warfare Center Dahlgren Division [NSWCDD]), and the Warfighter Machine Interface (WMI; developed by DCS Corp.) providing interactive displays for the Wingman Vehicle Commander, Wingman Robotic Driver, and Wingman Robotic Gunner team members.

The development of a functioning Wingman SIL will allow the team to test the systems capabilities in a simulated Table VI evaluation, allowing integration and assessment of robotic and human-robot teaming capabilities in a repeatable and systematic platform (efforts will be led by the US Army Research Laboratory [ARL]). Integrating the simulation systems into the SIL will substitute the real-world environment and vehicles with a virtual environment and simulated vehicles. Two different simulation systems will be integrated into the SIL. The Autonomous Navigation and Virtual Environment Laboratory (ANVEL) simulation supports the RTK software for autonomous mobility. The Unity simulation supports the integration of the ARES software for weapons systems targeting and firing with dynamic results, such as hitting a target (Fig. 2).

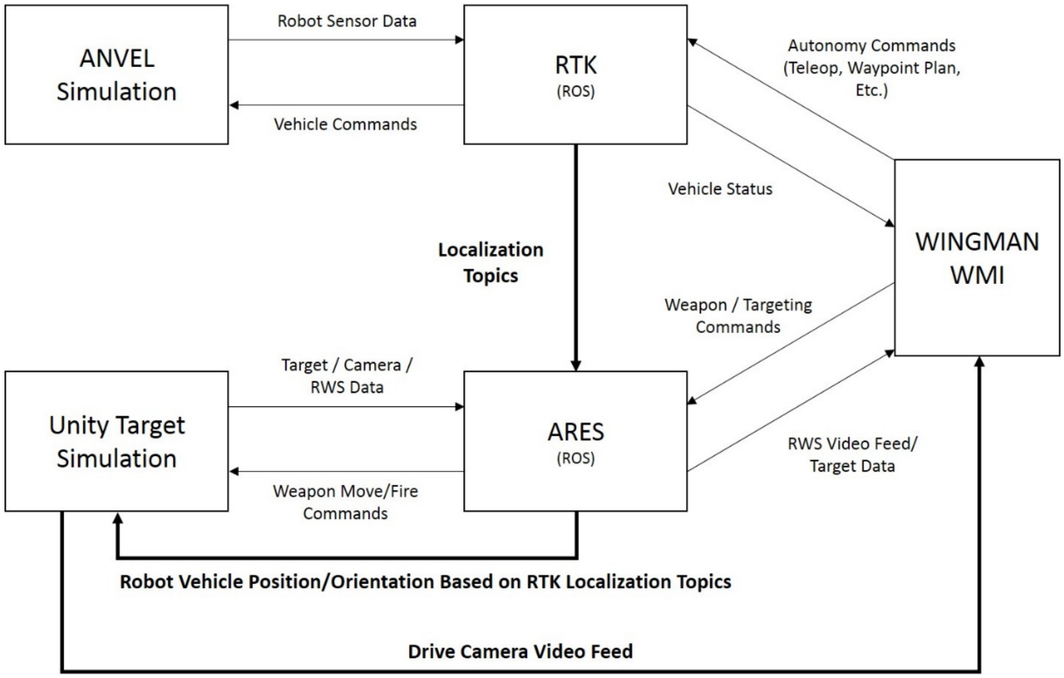

Fig. 2 Model providing the high-level software connections between the 3 Wingman vehicle software components (RTK, ARES, and WMI) and the 2 simulation environments (ANVEL and Unity). In this figure the bold lines represent the new connections that were made during the integration event.

3.1 Robotic Technology Kernel

The RTK is a government-owned, -designed, and -maintained modular autonomy kit for science and technology development that provides a set of common robotic capabilities, including autonomous mobility, across a variety of platforms and efforts. It enables capabilities to build on each other rather than replace existing ones. This results in cost and time savings as new efforts do not need to build from the ground up. Development efforts further the technology by feeding new capabilities back into the RTK, which in turn becomes available for other RTK-leveraged efforts and even legacy platforms. The Dismounted Soldier Autonomy Tools (DSAT) program has continued under the Wingman, Multi-UGV Extended Range (MUER), Trusted Operation of Robotic Vehicles in a Contested Environment (TORVICE), and Autonomous Ground Resupply (AGR) programs (see Mentzer 2014). Integration efforts can take advantage of the RTK by lowering the cost and schedule burden to include robotic technology in capability demonstrators.

3.2 Supervised Autonomous Fire Technology: Autonomous Remote Engagement System

The Supervised Autonomous Fire Technology (SAF-T) program is creating a set of detection, tracking, and fire control algorithms to enable supervised autonomous engagement of targets with a machine gun mounted on a RWS, tied with the development of an automated decision system to manage engagements. The system will autonomously identify, track, and compute firing solutions for targets within the effective range of the weapon system using onboard cameras, communicate with the operator for fire authorization, and then autonomously engage authorized targets.

Supervised autonomy will enable the next generation of RWSs, allowing the operationally effective weaponization of unmanned ground and surface vehicles (UGVs and USVs) with direct-fire engagement systems. It will overcome current limitations of RWS, such as limited situation awareness, and command latency by allocating operator tasks like targeting, tracking, and fire control to the software systems on the platform. Warfighters will approve targets using imagery generated from the autonomous target acquisition and tracking system. This approval is required for both real-time and future-time engagements. If the operator gives approval to engage a target but the criteria for engagement are not met, such as a lost track or firing toward nonhostiles, then the system will not engage. If the criteria are met at a later point in time, a new operator approval will be required to engage. Proper allocation of these tasks will enable Warfighters to fight at the speed of machines, surpassing the limits of human reaction time by focusing operator attention only on the engagement decision and not on the details of aiming and firing the weapon system.

ARES is a product of the SAF-T program and comprises a computer control box and RWS that provide automated engagement capabilities to

- decrease target acquisition time with slew-to-cue, video-based automatic target detection, and user-specified target selection,
- decrease engagement time with video tracking and user-assisted fire-control to keep the weapon system pointed on target, and
- overcome wireless control latency since the system will point the weapon while the user adjusts aim and pulls the trigger, thus removing issues with delay in the sensor-shooter loop.

In FY16, ARES was demonstrated on a stationary manned platform in the Office of the Secretary of Defense's Remote Weapon Station Auto Prioritization, Targeting, and Operator Cueing (RAPTOR) project to automatically detect and track personnel targets using day camera video. In addition to participating in Wingman in FY17, ARES will be used to develop a weapons payload for the US Marine Corps' Robotic Vehicle (Modular) demonstrated on a moving unmanned platform to automatically detect and track personnel as well as vehicle targets using both day and night camera video.

3.3 Wingman Warfighter Machine Interface

To develop a tightly coupled system of mobility and weapon control on the vehicles, an interactive user interface called the WMI was developed. The WMI is a custom user interface for crew interaction with a given vehicle's software. It is built upon the open source Qt framework, with a library of custom widgets available. For example, custom widgets include pan/tilt gauges, compass heading banners, soft handle overlays for sensor azimuth and elevation control, and the like. The interface provides access to capabilities required at a given crew station. Each crew station is customizable in an XML configuration file, launched by a script corresponding with the crew station's intended use. WMI scripts are currently supported for Windows, Linux, or Android architectures. Each crew station configuration file specifies available video streams as well as available controls and status to be displayed. The Qt framework supports style sheets for display customization.

The Wingman WMI is further customizable by user settings to allow control over things such as preferred units (metric/imperial), default overlay color, and others. The Wingman Gunner, for instance, is primarily focused upon the lethality functionality of the robotic vehicle. Therefore, the weapon video feed(s), weapon controls, and weapon status updates are located in the primary view for this station. Likewise, the Wingman Robotic Driver WMI is primarily drive camera- and map-centric (Fig. 3). The Wingman Vehicle Commander has access to all robotic vehicle video feeds, though typically in an observational capacity.

Fig. 3 Robotic Vehicle Driver WMI in use during Table VI exercise

3.4 Autonomous Navigation and Virtual Environment Laboratory

As part of RTK development, integration has been done with the ANVEL, a virtual simulation tool that specializes in UGV technologies and applications. ANVEL provides a modular "virtual proving ground" for developing and testing UGV technologies in an interactive, visual manner. The application simulates a wide collection of variables that are involved in vehicle operation, including physical mobility and sensor modeling. The RTK team has been able to use ANVEL as a high-level development tool for testing basic functionality of autonomy behaviors, functional testing, integration with the WMI, and exploration of new sensors and sensor placement. For testing, ANVEL is used as a vehicle representation that provides the sensor data to feed the autonomy and the mobility response to the autonomy commands. This is achieved by creating a connection between ANVEL and the Robot Operating System (ROS) through custom ANVEL plug-ins and ROS packages.

3.5 Unity

Unity is a cross-platform game engine developed by Unity Technologies (https://unity3d.com/unity). It is designed to work with a number of different application program interfaces (APIs) including Direct3D on Windows and Xbox

360, OpenGL on Mac, Linux, and Windows, OpenGL ES on Android and iOS, as well as proprietary APIs on video game consoles. Unity provides a customizable and dynamic virtual environment that is being used to support and test the weapons targeting and firing protocols for the Wingman project (Fig. 4). These simulation frameworks include the Virtual Prototyping Environment for Robotics (ViPER) upon which the Wingman targeting environment is built.

Fig. 4 A ViPER simulation built using Unity3D for an ARES project

3.5.1 Virtual Prototyping Environment for Robotics (ViPER)

ViPER is an approach taken by the Unmanned and Robotic Systems Branch (H42) of NSWCDD to aid in the research and development of autonomous systems. When developing or integrating systems, especially joint systems, we seldom have all of the systems' subsystems in one place at one time. Similarly with components under development, those components may not be mature enough to be integrated immediately. Instead of waiting until all of the pieces are in one place, which may only happen occasionally at integration events and demonstrations, or waiting until a component is mature, we use simulation to instantiate those subsystems and components that are not readily at hand.

The ViPER framework can stimulate real components or emulate components with simulation. The basic intent for ViPER is that every component connected in ViPER should be unable to determine whether any other component is real or simulated. The notion of developing a version of a system specifically to be used in simulation is redundant and a drain on limited resources, not to mention a challenge to configuration management.

This approach has been successfully used to mock-up proposed systems, demonstrate a system's potential capability while most of its hardware components were still under development, and to give a system's developers a sandbox in which to test software components before testing them on the single hardware resource. The ViPER approach has also been used to build 2 different control paradigms, teleoperated and semi-autonomous operation, to conduct user experiments.

3.5.2 Wingman Targeting Environment

The Wingman Targeting Environment is being developed in the Unity game engine. It is the most recent ViPER simulation in support of unmanned systems development and, specifically, in support of Wingman integration. The targeting environment stimulates the ARES system by providing electro-optical gun camera (targeting) video and camera interfaces, an interface with an emulated Picatinny Lightweight Remote Weapon Station (PLWRWS), and a virtual instantiation of the Grayling Table VI gunnery range terrain shared with the ANVEL simulation. In the Wingman scenarios, 2 vehicles are simulated on the range: a command vehicle and a weapon vehicle. The weapon vehicle uses RTK for navigation and ARES for target management and engagement with the PLWRWS; its position and orientation are provided by RTK via ARES, and stimulated by a virtual drive camera feed from the Wingman Targeting Environment and by sensors in the ANVEL simulation. The command vehicle is positioned in the environment through event-driven scripting. A video feed of the weapon vehicle drive camera to the WMI and a simple implementation of the Long-Range Advanced Scout Surveillance System (LRAS3) targeting device on the command vehicle will also be provided by the environment.

4. Integration Event Goals and Outcomes

Prior to the integration event, there were 2 different parts of the SIL that were operational. First, the robotic mobility component connecting the ANVEL simulation to the RTK software and the Wingman WMI was developed to support the real-world vehicles. Second, the weapons targeting system connected Unity with the ARES software and the Robotic Gunner's WMI. A number of integration issues had to be addressed to allow the vehicle software to be integrated with 2 different simulation environments. The major goals of this event were to get all the software installed, develop a matching virtual environment between the two different simulation systems, pass data between the software systems and simulation environments, and update the WMI to work with both simulation environments.

4.1 Software Installation

The first potential hurdle for software integration was the successful installation of all the software. A number of problems developed along the way.

4.1.1 Problem 1: Computers to Run the SIL

The first problem to be addressed was the minimum number of computers that would be needed to be able to run a working version of the complete SIL. Prior to the event, a single Windows machine was able to run ANVEL and the Robotic Vehicle Operator WMI, with a Virtual Machine to run the RTK to test robot mobility. Three machines were needed to operate the weapons targeting and firing capabilities, including a Linux machine for ARES, and 2 Windows machines for Unity and the Robotic Gunner WMI, respectively. In order to include 3 out of the 5 manned roles for future teaming research efforts, the working SIL will require a minimum of 5 machines, including those described above as well as the fifth machine for the Vehicle Commander WMI.

Another ideal change for a future setup would be to break apart the ANVEL, RTK, and Robotic Vehicle Operator WMI elements to separate computers. Running all 3 on a single computer leads to high power requirements (main instance at TARDEC uses a Xeon W5580 @ 3.20GHz processor with 16 GB of RAM and can still be overloaded, hampering real time simulation efforts). Separating RTK to a stand-alone Ubuntu machine also allows for software to be built/run in a more comparable way to how RTK will run on an actual vehicle.

4.1.2 Problem 2: Computer Networking

The SIL computers were connected through a D-Link gigabit switch. In order for the computers to communicate effectively, it was important to set each machine's IP address (below). The WMI for the robotic gunner, robotic operator, and vehicle commander should have unique IP addresses that do not conflict with another system.

1. RTK Virtual Machine 192.168.1.8 (virtual adapter bridged to ANVEL computer network adapter)
2. ARES 192.168.3.45
3. Unity 192.168.3.33
4. ANVEL 192.168.1.6

To set the IP, select the network adapter, highlight Internet Protocol Version 4 (TCP/IPv4), then click the Properties button. Change the IP (as shown above) and set the Subnet mask to a 16-bit network (255.255.0.0). The 16-bit mask was necessary to ensure visibility for all video streams on all WMI stations.

4.1.3 Problem 3: Installation of ARES

ARES, like RTK, relies on the ROS framework as the backbone of the system. It incorporates deep learning algorithms and relies heavily on image processing to execute target detection and tracking, which makes it very sensitive to the computer hardware on which it is running. Because it is not hardware-agnostic, building a disk image or a virtual machine image of ARES is not an option for installing the software.

Installing ARES is not a trivial process. It starts with installing ROS on an Ubuntu machine, then finding and loading the latest appropriate video card drivers to ensure it can use the latest GPU software, CUDA. This informs which version of the deep learning software, Caffe, will be loaded next to enable target detection. Then, the ARES software can be loaded, with its dependencies, and compiled. Again, as ARES is currently being developed actively for several configurations, of which Wingman is one, compiling the software for any specific application requires specific configuration. For these reasons, the installation of the ARES software for the Wingman configuration cannot currently be accomplished through an installer or virtual machine image.

4.2 Developing Matching Virtual Environments between ANVEL and Unity

Using 2 different virtual simulation engines in a federation may or may not require tightly matching terrain databases. In our case, The Wingman Driver will see video in his WMI from the weapon vehicle drive camera in the Wingman Targeting Environment. The Wingman Gunner will see targeting video also from that environment, and RTK will be making navigation decision based on ANVEL inputs. Most of those things are exclusive of the others; the exception is that the obstacles to navigation that RTK is sensing through ANVEL should also appear in the video from the weapon vehicle drive camera that the Driver will see in his WMI.

The current goal is to match the 2 terrain databases as closely as possible. However, the tangible requirement is in the video on the Driver's WMI. At no time should it appear as though the weapon vehicle is colliding with an obstacle that the RTK is successfully avoiding. Those are the only circumstances in which the terrain correlation will be obviously insufficient.

During the integration event, we successfully demonstrated that it was possible to import the ANVEL .raw16 terrain height map files in Unity to generate a terrain for the Wingman Targeting Environment. It appears that at present, vegetation and obstacle placement will have to be "hand-placed" in the targeting environment for both ANVEL and Unity. Additional testing will be required to ensure that geospatial information is also correlated between the 2 environments, as well as identify methods for creating textures on the terrain height maps.

4.3 Vehicle Localization Shared between the RTK and ARES

On the robotic vehicle, localization data (i.e., robot location, orientation, and GPS data) are shared between RTK and ARES so the weapon can accurately aim at targets. The data are shared using the multimaster package, which allows one ROS system to share specific topics with another ROS system. While exploring options for connecting the ANVEL and Unity simulations, the group found that sharing data between RTK and ARES with the multimaster package could also be used to match the vehicle position and orientation between ANVEL and Unity. In this system, ANVEL provides the data that RTK turns into localization topics which multimaster then shares with ARES. The Unity simulation is configured to use the vehicle position data from ARES to place the vehicle model, which should match the ANVEL position in the environment. This has the benefit of matching the simulation configuration to the real world configuration and does not require new ANVEL plug-ins or software to be developed for the co-simulation of ANVEL and Unity.

There were some initial challenges getting multimaster to function correctly. These were identified as the same challenges for connection between RTK and ARES on the real-world vehicle. Initially, multimaster was not installed on the RTK virtual machine because it was not part of the standard software build. Once multimaster was built, the host files on the RTK machine were modified to include the name and IP of the ARES machine, and the configuration files for multimaster were modified to indicate what topics should be shared from the RTK machine. For both the vehicle and simulation, the shared topics are /localization/UTM_odom/ localization/UTM_band; and /localization/UTM_zone. At this point, running the launch file for multimaster with the target IP set for the ARES machine allowed localization data to be pulled up in ARES.

4.4 Updates to the Wingman WMI

The primary changes occurred in the Robotic Gunner version of the Wingman WMI. The major problem was that the Robotic Gunner WMI was only designed to work with Unity, not a whole SIL. As such, Unity was using a vehicle model setup to work with a HMMWV, while ANVEL was designed to work with the MRZR vehicle. This led to an error launching the WMI. Since the ANVEL environment is highly customized for the MRZR configuration, changing HMMWV to MRZR was the fastest and cleanest solution. The following changes were made to the Robotic Gunner WMI:

1. To connect to the MRZR, the asset name had to match the platform controller by being set to "MRZR". In addition, C:\RVCA\config\services \asset-config-ares-mrzr.xml needed to be updated as well for video visibility, "M240SensorAres.[HMMWV|MRZR]”
2. The default location also had to be set to match simstart in C:\RVCA \scripts\start-ares-mrzr.bat and updating "set position"=32.2806 -106.476 1221
3. To have the cursor on target IP, an update was needed in C:\RVCA\scripts\start-ares-mrzr.bat, set cotIp=192.168.3.45
4. Having 2 like-named assets in the same location can cause problems. The ANVEL MRZR gets its position most directly, while ARES MRZR may lag slightly in transit via ROS Multimaster, which is converted to and sent over TCP. Therefore, in C:\RVCA\scripts\start-ares-mrzr.bat, set useCOTPosition=true.
5. To start ARES MRZR, now select C:\RVCA\scripts\start-ares-mrzr.bat
6. It is possible to use Aresstub as a work-around for video feed (competing with ANVEL for vehicle positioning).

5. Conclusion and Future Work

Following the integration event, the team was able to demonstrate that all components of the SIL could work together. Figure 5 demonstrates the existing software connections (represented by the solid arrows), the new capabilities from the January 2017 meeting (represented by the dotted arrows), and the identified connections that still need to be developed (represented by the dashed lines).

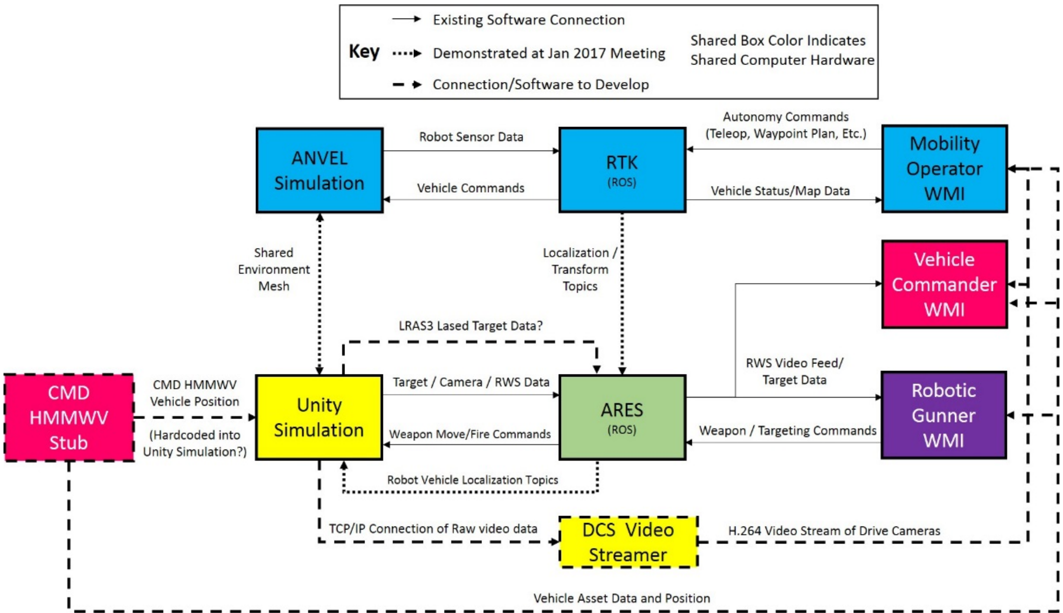

Fig. 5 Model of the detailed software connections that were previously developed, connections demonstrated at the integration event, and connections that still need to be developed

A full list of taskers is available in Appendix B. Near-term needs for the Wingman SIL include the following:

- Update WMI so that Unity video and virtual map are displaying in WMI. For the Commanders WMI, the map (on right window) should have the virtual map from Unity in order to set waypoints to match roads and paths.
- Determine if there is a way in the Commander WMI to load a predetermined waypoint path rather than manually adding them every time. This is important for consistency in trial runs. There is some potential for waypoint plans to be saved locally on the computer to address this issue. A back-up for not being able to place waypoints accurately on an overhead map would be to record a route driven via teleoperation and use that as a waypoint plan.
- Test targeting and firing, including assessment of dynamic capabilities of targets.
- Test manual control of manned vehicle including addition of both a manned and unmanned vehicle operating in Unity. Determine the requirements for this to occur.
- Create a virtual environment for Camp Grayling test course that integrates to both ANVEL and Unity.

6. References

Mentzer C. Autonomous ground systems: DSAT (Dismounted Soldier Autonomy Tools) – from conception to deployment. Presented at the 6th Annual Ground Vehicle Systems Engineering and Technology Symposium (GVSETS); 2014 Aug 12–14; Novi, MI [accessed 2015]. http://ww2.esd.org/GVSETS/PDF/AGS/1030Mentzer.pdf.

US Army Training and Doctrine Command. Training and qualification, crew. Washington (DC): Department of the Army; 2015 Mar 17. Training Circular No.: TC 3-20.31 [accessed 2017 June 6]. https://armypubs.us.army.mil/doctrine/index.html

INTENTIONALLY LEFT BLANK.

Appendix A. List of Attendees

Table A-1 List of attendees at the integration event

Name	Title	Affiliation	Role
Ralph W Brewer	Computer Scientist	ARL	Human-Robot Interaction, Simulation Assessment
Keith Briggs	Engineer	TARDEC	Wingman Test & Evaluation Lead
Eduardo Cerame	Mechanical Engineer	TARDEC	RTK, ANVEL
Xung Nham	Computer Engineer	DCS Corp	Plug-ins connecting RTK, ANVEL, and WMI
E Ray Pursel	Computer Scientist	NSWCDD	ARES, Unity, ViPER
Kristin E Schaefer-Lay	Engineer / Modeling & Simulation	ARL	Human-Robot Interaction, Simulation Assessment
Jae Song	Operations Branch Manager	DCS Corp	Plug-ins connecting RTK, ANVEL, and WMI
Thomas (Tom) Udvare	Robotic Project Manager	TARDEC	Wingman Project Manager
Benjamin (Ben) Wheeler	Computer Scientist	NSWCDD	ARES, Unity, ViPER
Anthony (Tony) Zimmermann (phone)	Computer Engineer	DCS Corp	WMI

Appendix B. Upcoming Taskers

Table B-1 Upcoming taskers

Tasker	Responsible	Deadline (2017)
Identify Configuration Management for Remote Collaboration	TARDEC	28 Feb
WMI	TARDEC	28 Feb
ANVEL Plug-ins	TARDEC	28 Feb
Virtual Machine / RTK	TARDEC	28 Feb
ARES (NSWC AiTR)	NSWC	28 Feb
ViPER (NSWC Simulation)	NSWC	28 Feb
Document Installation Instructions	ALL	28 Feb
Create Operator Instructions	ALL	Ongoing
Draft Technical Note	ARL	28 Feb
Update Architecture Document	TARDEC	6 Feb
Create HMMWV Software Stub	DCS	6 Feb
Video Server / Steamer		
Update Legacy Class	DCS	3 Feb
Develop Video Server	NSWC	28 Feb
Document Hardware Specs for SIL Reproducibility	TARDEC	6 Feb
LRAS Operator Interim Solution	NSWC	
Identify Approach (2 courses of action)	NSWC	13 Feb
Implement	NSWC	1 Apr
Metrics & Instrumentation		
SIL Experiment Plan	ARL	TBD
Human Subjects Approval	ARL	TBD
Eye tracking & Audio instrumentation	ARL	TBD
Modeling		
ERDC Follow-up	TARDEC	6 Feb
Update HMMWV Model	TARDEC	15 Mar
Test Course	TARDEC	28 Feb
Target Modeling	TARDEC	15 Mar
Future Work		
Driver Station	TBD	
LRAS Operator Station	TBD	

List of Symbols, Abbreviations, and Acronyms

AGR	Autonomous Ground Resupply
ANVEL	Autonomous Navigation and Virtual Environment Laboratory
API	application program interface
ARES	Autonomous Remote Engagement System
ARL	US Army Research Laboratory
DSAT	Dismounted Soldier Autonomy Tools
ERDC	US Army Engineer Research and Development Center
FY	fiscal year
HMMWV	high-mobility multipurpose wheeled vehicle
LRAS3	Long-Range Advanced Scout Surveillance System
MUER	Multi-UGV Extended Range
NSWCDD	Naval Surface Warfare Center Dahlgren Division
OTMSim	On-the-Move Simulation
PLWRWS	Picatinny Lightweight Remote Weapon Station
RAM	random access memory
RAPTOR	Remote Weapon Station Auto Prioritization, Targeting, and Operator Cueing
ROS	Robot Operating System
RTK	Robotic Technology Kernel
RWS	remote weapon system
SAF-T	Supervised Autonomous Fires Technology
SIL	software-in-the-loop
TARDEC	Tank Automotive Research Development and Engineering Center
TC	Training Circular
TORVICE	Trusted Operation of Robotic Vehicles in a Contested Environment

UGV	unmanned ground vehicle
USV	unmanned surface vehicle
ViPER	Virtual Prototyping Environment for Robotics
WMI	Warrior Machine Interface

1 (PDF) DEFENSE TECHNICAL
INFORMATION CTR
DTIC OCA

2 (PDF) DIRECTOR
US ARMY RESEARCH LAB
RDRL CIO L
IMAL HRA MAIL & RECORDS
MGMT

1 (PDF) GOVT PRINTG OFC
A MALHOTRA

1 (PDF) ARMY RSCH LAB – HRED
RDRL HRB B
T DAVIS
BLDG 5400 RM C242
REDSTONE ARSENAL AL
35898-7290

8 (PDF) ARMY RSCH LAB – HRED
SFC PAUL RAY SMITH CENTER
RDRL HRO COL H BUHL
RDRL HRF J CHEN
RDRL HRA I MARTINEZ
RDRL HRR R SOTTILARE
RDRL HRA C A RODRIGUEZ
RDRL HRA B G GOODWIN
RDRL HRA A C METEVIER
RDRL HRA D B PETTIT
12423 RESEARCH PARKWAY
ORLANDO FL 32826

1 (PDF) USA ARMY G1
DAPE HSI B KNAPP
300 ARMY PENTAGON
RM 2C489
WASHINGTON DC 20310-0300

1 (PDF) USAF 711 HPW
711 HPW/RH K GEISS
2698 G ST BLDG 190
WRIGHT PATTERSON AFB OH
45433-7604

1 (PDF) USN ONR
ONR CODE 341 J TANGNEY
875 N RANDOLPH STREET
BLDG 87
ARLINGTON VA 22203-1986

1 (PDF) USA NSRDEC
RDNS D D TAMILIO
10 GENERAL GREENE AVE
NATICK MA 01760-2642

1 (PDF) OSD OUSD ATL
HPT&B B PETRO
4800 MARK CTR DR
SUITE 17E08
ALEXANDRIA VA 22350

3 (PDF) US ARMY TARDEC
DIR TARDEC
T UDVARE
K BRIGGS
E CERAME

5 (PDF) DIR NSWCDD
F LAGANO
R POFF
R GRIPSHOVER
B WHEELER
E PURSEL

4 (PDF) DCS CORP
A ZIMMERMANN
J SONG
X NHAM
B WOOD

ABERDEEN PROVING GROUND

16 (PDF) DIR USARL
RDRL HR
J LOCKETT
P FRANASZCZUK
K OIE
K MCDOWELL
RDRL HRB
D HEADLEY
RDRL HRB A
M LAFIANDRA
RDRL HRB C
J GRYNOVICKI
RDRL HRB D
C PAULILLO
RDRL HRF A
A DECOSTANZA
RDRL HRF B
A EVANS
RDRL HRF C
J GASTON
RDRL HRF D
A MARATHE
K SCHAEFER-LAY
RDRL VTA
H EDGE
M FIELDS
R BREWER

INTENTIONALLY LEFT BLANK.