



THE USE OF GAMING ENGINES FOR DESIGN REQUIREMENTS

D. J. Gorsich 

U.S. Army CCDC-GVSC, United States of America

 david.j.gorsich.civ@mail.mil

Abstract

The automotive community has found to design and test autonomous systems, traditional CAE tools are not enough. The number of sensors and controls involved makes it very difficult to predict all the possible scenarios and system reactions to them. An approach to provide input to all the sensors and control systems is to use gaming engines. They are used “headless”, and in other cases with multiple users in the environment. In this paper we will highlight one case on how they are changing how the Army designs, tests, and sets requirements for autonomous ground systems.

Keywords: requirements, computational design methods, gaming engines, virtual reality (VR), customer integration methods

1. Introduction

The way the automotive industry and the U.S. Army iteratively design, develop and test vehicles is changing. Not only are traditional design tools used, but concepts and designs are brought into virtual environments created with gaming engines. This newer approach is changing naturally throughout industry and government because the cost to design and test complex systems with large numbers of sensors and controls is excessive. The race to field autonomous vehicles is accelerating these changes. Virtual environments are used to stimulate the sensors and control inputs are generated from gaming engines such as Unreal (<https://www.unrealengine.com>) or Unity (<https://unity3d.com/unity>), either in real time with users in the loop, or instead “headless” with larger numbers of simulations all running in parallel to sample a broader range of inputs.

For the U.S. Army, one challenge has been getting the initial design requirements correct. Often there are conflicting requirements and poor choices for initial requirements have led to designs that were ultimately too heavy or costly, leading to program failures (Dasch and Gorsich, 2017). Therefore, one key goal is to iterate early on initial design concepts (virtual prototypes) and requirements to find the best virtual prototypes. Gaming engines allow us to obtain relevant soldier feedback on concepts and designs in an operational context. Insights are gained from soldier tactics and behaviours to refine concepts and influence ground vehicle requirements, operational doctrine and designs. We are essentially putting future vehicle concepts into the hands of soldiers today, and modifying the designs iteratively.

2. Background

A number of recent papers have been published discussing this topic. (Wang et al., 2019) states that traditional computer aided engineering (CAE) tools focus on one vehicle, where connected and

automated vehicles must consider the surrounding environment such as other vehicles, roads, infrastructure, and pedestrians. This leads the designers of these systems to commercial gaming engines. As early as 2007, Graighead et al. implemented autonomous vehicles in a gaming environment with various sensors, such as 3D camera, GPS, odometry, inertial measurement units, and planar laser ranging. This work has continued within the design community. (Hudson et al., 2011) did some later work performing an iterative design work flow driving parametric designs with gaming engine output. Research has continued since then and has become more sophisticated. Recently, KTH Royal Institute of Technology in Sweden published papers on their research on the visualization of truck platooning using Unity3D (Estreen and Nord, 2018; Khosravi and Yassiry, 2018). Toyota InfoTechnology Center in USA also contributed a series of work to the vehicle prototyping research by Unity3D. (Yamaura et al., 2016) built a virtual prototype of Advanced Driver Assistance System (ADAS) with a closed-loop simulation framework that consists of four tools: Unity3D, Simulink OpenMETA, and Dymola. (Kim et al., 2017) proposed several associated directions and potential approaches of testing autonomous vehicle software in virtual prototyping environment using Unity3D, from the perspective of test criteria and test case generation. As an extended work of that, (Dai et al., 2017) presented a co-simulation tool-chain for the automated optimization of various parameters in the virtual prototyping environment. (Ekströmer et al., 2019) at the international conference on engineering design discussed also how gaming engines are important for automotive concepts. They used the Unreal 4 gaming engine to visualize automotive lighting design concepts.

An especially important use of gaming engines for design are their use in iteratively defining design requirements. In the past the U.S. Army defined material requirements in the Training and Doctrine Command or TRADOC. These requirements for vehicle systems many times did not account for physics or technological constraints and were largely determined by the wishes of soldiers in what would enhance their warfighting abilities in various operational scenarios. The engineers, scientists and ultimate designers in both government and industry were separated organizationally and often not involved in the determination of these requirements. Also changing these initial requirements often required senior level leadership above the different Army commands. More recently the Army created the Army Futures Command and put processes and organizational changes in place to ensure both the operational side of requirements development along with the research, development and acquisition side of the army were placed together. These changes helped facilitate the institutional adoption of gaming engines in the requirements process. We will discuss the details of this process in the following paper.

3. Methodology

The approach taken to determine or redefine requirements involves soldiers throughout the process. This iterative process can loop between concepts, requirements and feedback from soldiers in virtual experiments enabled by gaming engines.

The initial step depicted in the top left of the following Figure 1 involves soldier innovation workshops. In these initial workshops soldiers with field experience discuss the ground systems they have or wish they had. With the soldiers are industrial designers from places like the College of Creative Studies who specialize in design and sketching out ideations and concepts. These concepts are then brought into the virtual prototyping phase. [Prototypes for Next Generation Combat Vehicles \(NGCV\)](#) are generated from prior knowledge of ground systems and technologies so an idea of sizes, weight and power are determined and a trade space is developed. These virtual prototypes can then be used to build physical prototypes, or they can be sent into a virtual environment (bottom right of the figure). This virtual environment and what is done there is the subject of the rest of this paper.

The virtual environment is built from a modern gaming engine such as Unity or Unreal. While gaming engines are used, developmental efforts ensure realism for the soldiers a military relevant gaming environment. In this environment up to 100 soldiers who have trained together for various operational environments can work together to accomplish a mission in the environment, and can become injured or fatigued within the game. They can get in and out of the vehicles, fight an opposing force, and call in fires. Soldiers, staff experienced in operations, as well as data analysts, vehicle designers and engineers are involved during the experiment.

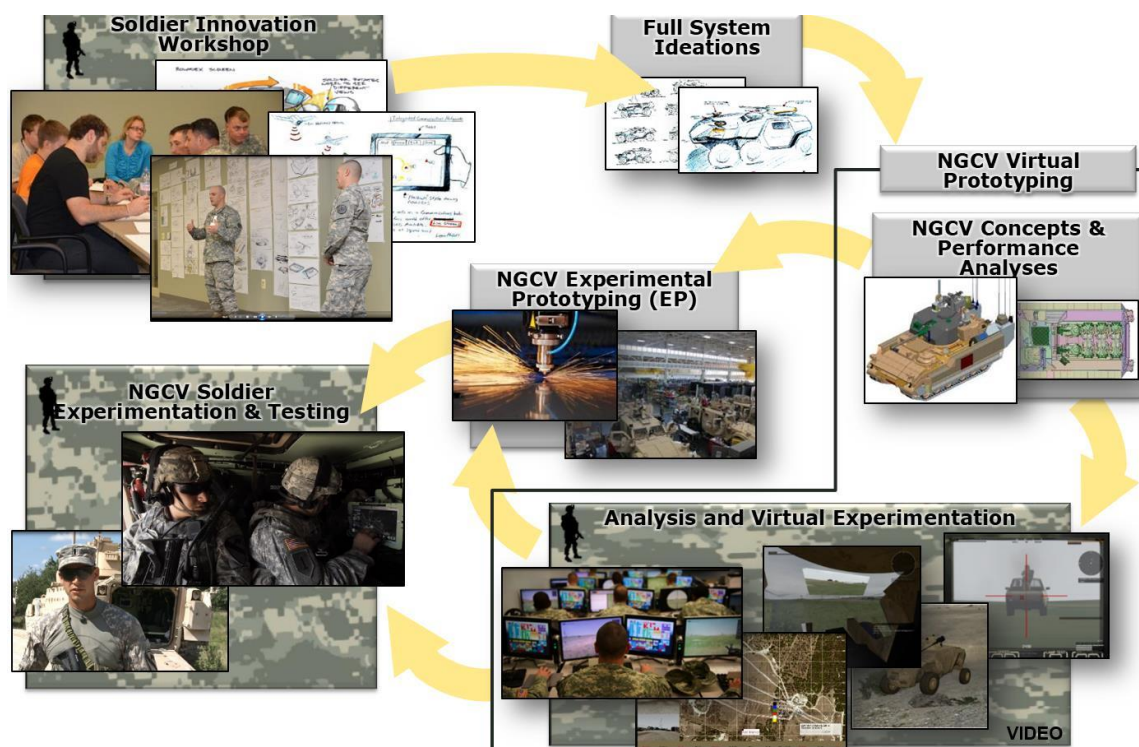


Figure 1. The virtual prototyping and experimentation process

3.1. Experimental methodology

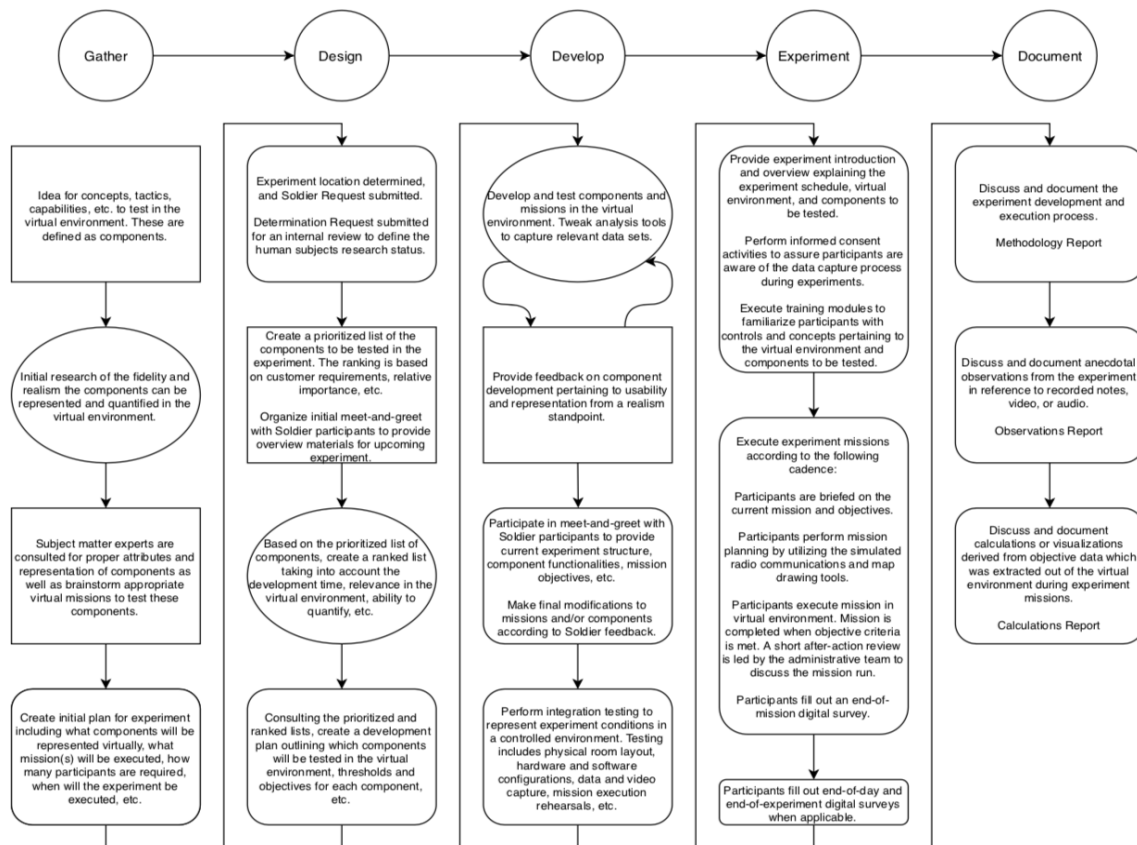
The experiment process consisted of five main phases: gather, design, develop, experiment, and document. The steps to provide results to the design of the system are depicted in Figure 2. The gather phase aligned the experiment objectives with the learning objectives from the NGCV program. The design phase characterized how the experiment would be structured in the form of experiment missions (route reconnaissance, area reconnaissance, breach) and experimental stimulus (technologies, opposing forces). The develop phase assures proper infrastructure support for experiment execution and data collection as well as proper representation of real-World assets and capabilities within the virtual environment. The experiment phase covered execution (training, missions, and administration) and data collection (after action reports, video, surveys) of the virtual experiment. The document phase organized the output products (methodology, calculations, observations) for the virtual experiment.

3.2. Experiment calculations

Virtual experimentation provides value directly through the qualitative/anecdotal feedback obtained from participants. However, as participants are manipulating their avatars in the virtual environment, we are able to instrument their behavior, potentially with far more meaningful accuracy than what can be relayed through qualitative exploration. Given the caveat of limited sampling, several notional metrics are examined (Map Traversal, Time Spent Static vs. Dynamic, system Deployment Count & Location, system Deployment Duration, technology utilization over time, and mission success metrics). This is an area of great interest to fund additional research in.

3.3. Experimental observations

Throughout the experiment the research team garnered feedback from Soldiers in the form of surveys, formal After-Action Reviews (AARs), and informal side discussions. Surveys were taken by Soldiers after each mission run to allow them to give unfiltered feedback prior to larger group discussion. AARs were lead primarily by the unit's Platoon Leader and supplemented by the research team for clarification of concepts and technologies. In addition to these primary forms of data collection, the research team and Soldiers had several smaller group discussions to further explore these concepts and technologies.



(Circles represent phases — Ellipses represent developer actions — Rectangles represent customer/stakeholder actions — Rectangles with rounded edges represent collaborative actions)

Figure 2. The experimental design process

4. Completed experiments

A series of experiments with the above methodology were conducted with the goal of obtaining relevant soldier feedback in an operational context on concept ground vehicle systems and technologies. Insights were gained from soldier tactics and behaviours to refine concepts and influence ground vehicle requirements, operational doctrine and design. Future vehicle concepts were used by soldiers in operational scenarios before any vehicle has been built.

In Virtual Experiment 1, during December 2018, 16 soldiers experimented with manned-unmanned teaming (MUM-T) concepts using a robotic combat vehicle (RCV) prototype. A modern gaming engine was used which accounted for the realistic aspects of an operational environment such as soldier injuries, fatigue and system performance. Feedback from the experiment informed RCV requirements under the NGCV program, and informed the Army regarding the improved survivability and the use of unmanned aerial vehicles with RCVs and how the requirements should be set for them.

In Virtual Experiment 2, during March 2019, 27 soldiers provided further feedback regarding RCVs along with a demonstrator vehicle that was built called the Mission Enabling Technology Demonstrator or MET-D. A key investigation was to determine the best approach to command and control the RCVs from the MET-D. This experiment has informed the MET-D live physical experiments that will be carried out in March 2020. These live experiments will further validate a unit's ability to conduct manned-unmanned teaming.

In Virtual Experiment 3 during September 2019, 30 soldiers provided feedback on RCV crew configurations, formations, enabling technologies and networked capabilities. This feedback included the exploration of new tactics, techniques, and procedures (TTPs) of employing the new capabilities. This included the determination of optimal employment in open and urban terrains,

how control of RCVs is shared between mounted and dismounted soldiers, and trade-offs between survivability and maneuverability requirement of the RCVs.

In Virtual Experiment 4 in October 2019, six soldiers supported Crew Optimization and Augmentation Technologies experiment and provided feedback on the RCV crew station design and user interfaces. In these experiments, actual crew stations and software were integrated into the gaming environment. This experiment greatly informed the machine interface design. It also informed our process to collect and analyse MUM-T research data, specifically team performance measures. Furthermore, the virtual environment was improved using the Unreal 4 gaming engine.

There were key realizations and decisions made about requirements for NGCV. First, having an assured wireless “tether” or communication link for the RCVs was crucial. Secondly, five soldiers were needed for the control of two RCVs, and the specific ways in which they would be controlled were determined. The time required to train soldiers how to use the RCVs was much less than anticipated. Challenges with signature management of the RCV were identified as well as the requirements of 360-degree situational awareness sensors. Finally, the requirements of the tether on the unmanned aerial vehicles and how they would be used in combination with the RCVs were found. These determinations were made mainly from soldier feedback after the experiments, but also observations of how the soldier teams performed during the exercises and some basic analytics on the data.

5. Conclusions

Traditional computer aided design tools used to develop virtual prototypes including CAD models, dynamics, mobility and structural modelling, along with prior vehicle designs, are used to develop a trade space of designs. This trade space of designs is constantly informed by soldier feedback. More recently virtual prototypes from this trade space have been inserted into virtual experiments with soldiers to determine operational capabilities of the prototypes and whether the initial requirements should be adjusted to improve the operational results. This gives us early feedback on the actual capabilities of the designs and technologies. These design cycles can be done quickly before actual physical prototypes are built. Because of the complexity of autonomous ground systems, and because their interactions and use by soldiers often determine their actual usefulness in battle, the use of gaming engines to create virtual environments to observe soldier-system interaction as well as stimulating all the systems sensors and controls is here to stay in determining design requirements.

Acknowledgement

The author would like to thank the tremendous work being done by the virtual experiments group at the [U.S. Army Ground Vehicle Systems Center \(GVSC\)](#) and the authors of the GVSC technical reports, where much of this information has come from. Their vision and skills have made this all possible.

References

- Craighead, J., Burke, J. and Murphy, R. (2007), “Using the Unity3D Game Engine to Develop SARGE: A Case Study”, *Computer*.
- Dai, S. et al. (2017), “Control Parameter Optimization for Autonomous Vehicle Software Using Virtual Prototyping”, *Proceedings of IEEE 28th International Symposium on Software Reliability Engineering*.
- Dasch J.M. and Gorsich D.J. (2017), “The Role of R&D in an Acquisition Program”, *Journal of Defense Management*, Vol. 7 No. 2.
- Ekströmer, P. et al. (2019), “Shedding Light on Game Engines and Virtual Reality for Design Ideation”, *International Conference on Engineering Design, ICED19*, 5-8 August, Delft, Netherlands.
- Estreen, T. and Nord, S. (2018), “Visualization of Platooning in Unity3D”, *Degree Projects in Technology*.
- Ground Vehicle Systems Center Immersive Simulations Group, (n.d.), “Manned-Unmanned Teaming (MUMT) Virtual Experiment #1,” Defense Technical Information Center.
- Ground Vehicle Systems Center Immersive Simulations Group, (n.d.), “NGCV MUM-T VE COAT experimental report Oct 2019”, Defense Technical Information Center.
- Ground Vehicle Systems Center Soldier Experimental Gaming and Analysis Group, (n.d.), “Next Generation Combat Vehicle Manned-Unmanned Teaming Virtual Experiment,” Defense Technical Information Center.

- Ground Vehicle Systems Center Soldier Experimental Gaming and Analysis Group, (n.d.), “NGCV MUM-T Virtual Experiment #3; Virtual Experimental Rollup,” Sept 2019, Defense Technical Information Center.
- Hudson, R., MacDonald, D. and Humphreys, M. (2011), “Race track modeler. Developing an iterative design workflow combining a game engine and parametric design”, *ACADIA Regional 2011: Parametricism*.
- Khosravi, M. and Yassiry, A. (2018), “Virtual Truck Platooning Implementation in Unity3D”, *Degree Projects in Technology*.
- Kim, B. et al. (2017), “Testing Autonomous Vehicle Software in the Virtual Prototyping Environment”, *IEEE Embedded Systems Letters*, Vol. 9 No. 1, pp. 5-8.
- Wang, Z. et al. (2019), “Agent-Based Modelling and Simulation of Connected and Automated Vehicles Using Game Engine: A Cooperative On-Ramp Merging Study”, arXiv: 1810.09952.
- Yamaura, M. et al. (2016), “ADAS Virtual Prototyping Using Modelica and Unity3D Co-Simulation via OpenMETA”, *Proceedings of Japanese Modelica Conference*.