

## Rapid Simulation Model Development (RSMD) for Computer Generated Forces

Jackie Zhang, Daniel Olson  
**Infinitas Engineering Inc.**  
Jackie.zhang@infinitasinc.com  
Daniel.olson@infinitasinc.com

Bryan Cole  
**Cole Engineering Services Inc**  
Bryan.cole@cesicorp.com

### ABSTRACT

Computer Generated Forces (CGF) simulations use physical models as the basic components of simulated entities, such as weapons, munitions, sensors, etc. These physical models contain mathematical representations of combat systems and their interactions with the environment and other simulated entities. In order for the simulations to be useful for Live, Virtual and Constructive (LVC) training, the models must use valid algorithms and data from authoritative sources. However, the development of such validated models is resource intensive because of the complex data representation and mathematical implementation. It is cost effective to recreate these models for every simulation program. The need of building validated physical models once and allowing different types of simulation systems to use/reuse these precalculated models when needed is more in demand than ever.

Validated physical models can be from different domains and disciplines with different format and standards. The mathematical formulas are designed to use the ground-truth data to produce meaningful data values that a simulation system can use for entity simulation. We are developing a process and a software tool for automatically generating models for CGF simulations from common authoritative data sources. A proof of concept of this tool, Rapid Simulation Model Development (RSMD) Toolkit, implemented a physical modeling process by using US Army Combat Capability Development Command (CCDC) validated physical models stored in Physical Knowledge Acquisition Documents (PKADs). The RSMD's framework entrusted Modular Open Systems Approach (MOSA) compliant technologies by plugging in the PKADs as an authoritative data source, constructed several physical models for the machine gun M-16, and produced a set of validated physical data which was consumed by the M-16 entity simulated in a well-known CGF system named VR-Forces. The physical models implemented thus far are Direct Fire Weapon Accuracy, Rate of Fire, and Direct Fire Weapon Characteristics. The VR-Forces M-16 scenario that used the validated data showed a statistically significant enhancement in the gun's hit and miss calculation accuracy. The architecture ensures multiple data sources and simulation engines can be plugged into or removed from the architecture as needed. This paper provides the statistical improvements found in our study and the RSMD toolkit software architecture design methodologies.

### ABOUT THE AUTHORS

**Jackie Zhang** is the CEO and owner of Infinitas Engineering Inc. Her 20+ years of software systems engineering experiences has been focused on various development phases of CGF systems especially physical modeling and behavioral modeling, cultural modeling, modeling with Expert Systems and Case-based Reasoning techniques, etc. Ms. Zhang received Bachelor and Master of Science degrees in Computer Science from University of Central Florida.

**Daniel G. Olson** is a senior simulation software engineer with seven years of experience working on various training and simulations systems for the United States Army and Navy. Mr. Olson received Bachelor and Master of Science degrees in Computer Science from University of Kentucky.

**Bryan Cole** is the founder and CEO of Cole Engineering Services Inc. (CESI) It was Mr. Cole's vision for a customer centric, innovative and responsive engineering company that gave birth to CESI in 2004. As a former Division Manager at SAIC, Mr. Bryan managed the development of complex modeling and simulation systems like ModSAF, Automated Exercise and Assessment Systems (AEAS), Dismounted Infantry SAF, VERTS, and OneSAF.

## **Rapid Simulation Model Development (RSMD) for Computer Generated Forces**

**Jackie Zhang, Daniel Olson**

**Infinitas Engineering Inc.**

[Jackie.zhang@infinitasinc.com](mailto:Jackie.zhang@infinitasinc.com)

[Daniel.Olson@infinitasinc.com](mailto:Daniel.Olson@infinitasinc.com)

**Bryan Cole**

**Cole Engineering Services Inc.**

[Bryan.cole@cesicorp.com](mailto:Bryan.cole@cesicorp.com)

### **INTRODUCTION**

Theories of warfare have been tested and refined in military simulation systems to conduct mission planning, mission rehearsal, as well as to predict wargaming results. There have been many different types of Live, Virtual, Constructive and Gaming (LVCG) simulation systems developed to support wargaming capabilities for the military over the last thirty years. One of the challenges these simulation exercises may face is that their results/conclusions maybe flawed due to the approximate nature of the physical models used. The basic building block of any simulation systems is its physical models, such as vehicles, aircraft, weapons, munitions, etc. These models are the lowest level of simulated entities and are used in every aspect of wargaming exercises. Most currently available simulation systems have varying degrees of realism, but not many use authoritative data in their models. Without accurate physical models, it is impossible to generate realistic scenarios in any wargaming exercise.

In the modern Army simulation world, simulation system builders know that the golden standard of validated combat related physical models are those that were developed using algorithms from agencies such as US Army Combat Capability Development Command (CCDC) Data and Analysis Center. CCDC provides a large set of validated algorithms and accurate data sets that can be used to simulate physical models that best emulate real world combat equipment. However, not many Army's simulation systems use these authoritative data for their models because in order to make the best use of such data, the simulation modelers not only need a thorough understanding of the complex data format, but also have to implement their own methods to parse the data contents so they can figure out how to use the data in their specific simulation engine. This is a labor-intensive process if the system is built from scratch. If the simulation system was not built with such authoritative data from the beginning, getting authoritative data into existing models takes additional effort because the existing data representations may be in a completely different format, and the modelers will have to go through data analysis, extraction and optimization process to figure out how to insert new data into their existing simulation engine. Every time a different simulation needs authoritative data, the whole process has to be iterated again, with equal amount of time and knowledge spent on that simulation.

Therefore, the challenge is how to get the authoritative data, such as from CCDC, to simulation systems efficiently, regardless if such systems are being developed from scratch or have already been deployed and are used with only non-validated data. In this paper, we present the concept of a toolkit that can facilitate use of authoritative data to develop validated physical models for different simulation engines. This innovative approach utilizes CCDC validated physical models which are stored in Physical Knowledge Acquisition Documents (PKADs) and follows a proven Knowledge Acquisition (KA) and Knowledge Engineering (KE) processes used by simulation engineers and provides intermediate tools and User Interfaces (UI) to allow simulated physical models to be developed with increased accuracy and reusability. A Proof of Concept (POC) system named Rapid Simulation Model Development (RSMD) Toolkit is being developed to test this approach and the intermediate results and analyses are presented in this paper

## **BACKGROUND**

Our feasibility study has been focusing on what the MS&T industry partners have been working in terms of inserting authoritative data in their systems. The result of the research provided a frame of reference on how we should move forward to build validated models for current and future CGF systems. The following systems were selected for the proof of concept development phase based on their design pedigree, adaptability and functionality.

### **One-Semi Automated Forces (OneSAF) and Combined Arms Command and Control Training Upgrade System (CACCTUS)**

OneSAF is a CGF simulation that provides entity-level models and behaviors that are both semi-automated and fully automated. It provides real-world representations of platforms, soldiers, equipment, logistical supplies, emerging threats, and aviation assets designed for Army echelons brigade and below. OneSAF uses the physics-based modeling and data collection from CDCC to simulate its real-world representations of the battlefield. OneSAF is unique because all of its physical models were developed using CCDC data from the beginning. Although OneSAF once had its own representation of physical and behavioral model data, it has since adopted CCDC's Standard File Formats (SFF) (Monday, 2015) which eliminated the need to reformat CCDD data in order to be ingested and used by OneSAF. Once the models are created and tested, OneSAF's domain Subject Matter Experts (SME) collaborate with the modeler and testers to evaluate if the data is represented correctly. This is done at the model verification and validation phase before each physical model is finally integrated into the software baseline. Once the validated physical models are built, behaviors that are associated with these physical models can be created, and military scenarios, mission planning, and wargaming exercises can take place.

Because OneSAF has a large library of validated physical models, it is considered a valuable training and analysis simulation by the Army's analytical community. US Marine Corp's CACCTUS uses the OneSAF line of entity level constructive simulation products as their simulation engine because of its ability to use CCDC model data to develop Marine-specific models, behaviors and task organizations (Armstrong, 2004). Using the validated models in CACCTUS training scenarios, commanders and their staff direct the battle from their combat operation centers, while the virtual forces fight the battle using OneSAF, to ensure effective and realistic training results are generated. Other training and analysis systems, such as Joint Land Component Constructive Training Capability (JLCTCC) and Aviation Combined Arms Tactical Trainers (AVCATT) use OneSAF as a validated simulation engine because of its validated physical models.

Since its inception in 2000, all physical models developed for OneSAF are an integral part of the OneSAF model repository, which was designed to serve OneSAF's operational and technical requirements only. Although internally composable, none of the architectural components in the OneSAF architecture were designed to be used by other simulation systems. The design philosophy for OneSAF has been focused on developing a single self-contained CGF system that provides specific human in the loop training and analytical support across the Army. The OneSAF's KA/KE/Modeling process has been proven to be mature, practical and advantageous when validated data is used. RSMD Toolkit adopts this proven process and uses CCDC validated data from a subset of unclassified CACTUSS database.

### **VR-Forces**

VR-Forces is a MAK Technologies CGF solution. It provides land, maritime and airspace simulation models within synthetic environments. A much used CGF in the international simulation industry, VR-Forces provides intuitive UIs to build and execute scenarios. It includes both entity level and aggregate level simulations with various models such as movement, sensor, combat, non-combat, and command and control. However, VR-Forces models were not developed using any US military validated data sources for its models. Data-driven methods were developed to ingest parameterized sets of model data into the associated behavior models during runtime, while leaving some ground truth calculations at runtime. This approach served the purpose of building an exportable CGF system that used can customize for their particular forces and fidelity requirement. However, it may not satisfy modern military training and analytic rigor requirements. RSMD uses VR-Forces as the target simulation engine for our case study in adding authoritative models to a CGF simulation.

## Parametric Data Services (PDS)

PDS is a Defense Advanced Research Projects Agency (DARPA) Government-off-the-Shelf (GOTS) tool. PDS was developed as a software tool for the Prototype Resilient Operations Testbeds for Expeditionary Urban Scenarios (PROTEUS) project. It includes a repository of an initial baseline of CCDC validated CACCTUS physical model data to include combat equipment, munitions and platform models that is accessible via web services. Currently, PDS hosts its data repository in MongoDB. Its Graphical User Interface (GUI) provides mechanisms for the RSMD toolkit users to query model data sets and values needed by the computational algorithms specified in PKADs.

## RSMD DESIGN PHILOSOPHY

The CGF training and analytical community has been trying to create innovative technologies that can provide simulation modelers with tools and mechanisms to efficiently develop models for their simulation engines (Oijen, 2019). These types of technologies should increase productivity, lower costs and be able to grow with the moving target of additional requirements and increasing training rigor. Our design philosophy is based on years of simulation modeling experience coupled with a vision to bring the modeling community together to tackle the hard problems, when modeling military assets in simulations. RSMD is a tool designed for simulation modelers, with the concept of an expandable, reusable and sustainable toolkit to promote the use of authoritative data; simplify the physical modeling process; and to encourage reuse of well-developed and validated models by the community.

*Expandability:* RSMD is designed to accommodate future additions to its capacity of developing new physical models. Types of additions include new authoritative sources, new physical models, new simulation engines, and new algorithms that present ground truth of existing and/or new models.

*Reusability:* Any validated physical models developed, verified and stored in the RSMD repository can be reused by any simulation engine with minimum development effort. Because the ground truth of physical models does not change, RSMD will ensure that the calculations can be easily adapted for different simulation engines by reusing existing computational algorithms.

*Sustainability:* With more new simulations being built and more legacy systems being modernized, RSMD will remain data-driven and functionality focused, while following the proven modeling process, in order to maintain its usability and credibility.

## How RSMD Works

The design philosophy provides the fundamental guidelines for the RSMD software design and implementation that follow. The result is a software development toolkit designed for modelers who build models for military platforms in their simulation engines. This toolkit can receive multiple authoritative data sources and build physical models for multiple simulation engines. Each model is built according to selected source input and stored in a model repository. With new requirements to build additional models, the modeler can verify if it's already in the repository and decide if this model needs to be created from scratch or adapted by using existing models for the selected simulation engine (Figure 1.)

- *Authoritative Data Sources:* A set of validated model data sources that can be used to develop physical models. For example, CCDC is one source from the US Army.
- *Input Components:* When a data source is specified, an Input Component is created to process all required raw model data for use in the RSMD toolkit.
- *Controller:* Takes in processed data and produces simulation engine model data.
- *UI Component:* Used by the modeler to define output format, the input source to use, and to get information on the type of model to be created for which simulated entity.
- *Output Components:* takes the authoritative data and produces model data in the format of the designated simulation engines.

- *LVCG Simulation Engines*: This is the set of target simulation engines that can be used to consume the physical models created by RSMD for wargaming exercises.

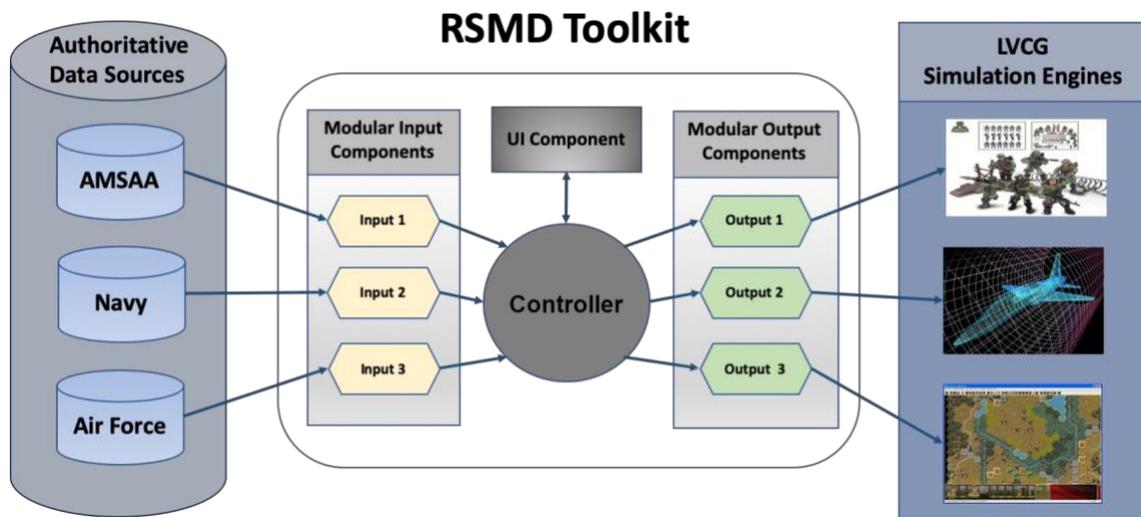


Figure 1 RSMD System Overview

## RSMD ARCHITECTURE

The purpose of the RSMD toolkit is to assist physical modelers to build validated military platform representations in simulation engines. The development of CGF physical models is resource intensive because the knowledge needed to build models in any specific simulation engine comes from understanding how the simulation engine uses the models and how they connect with other components in the system. Each simulation engine is built differently with different goals, databases, design philosophies and implementation choices. Therefore, the plethora of simulation-specific modeling tools are useless when new models are needed for different simulation engines because transferring models to another simulation is simply not possible. Typically, modelers have to start from scratch.

RSMD architecture follows Modular Open System Architecture (MOSA) methodology, to ensure components are modular and that the technology is scalable to support various data sources and simulation engines. Each component can be plugged into or removed from the architecture as needed. Common interfaces are shared amongst components, yet all components are loosely coupled. For example, any time each new authoritative data source is specified, one input component is created to process its data. Similarly, one output component is created for each designated Simulation Engine to handle its unique data format. This design ensures a set of highly cohesive, yet severable modules working together to take away the burden of developing new models and/or inserting authoritative data into simulation engines of choice. Figure 2. depicts how our component architecture follows MOSA design disciplines:

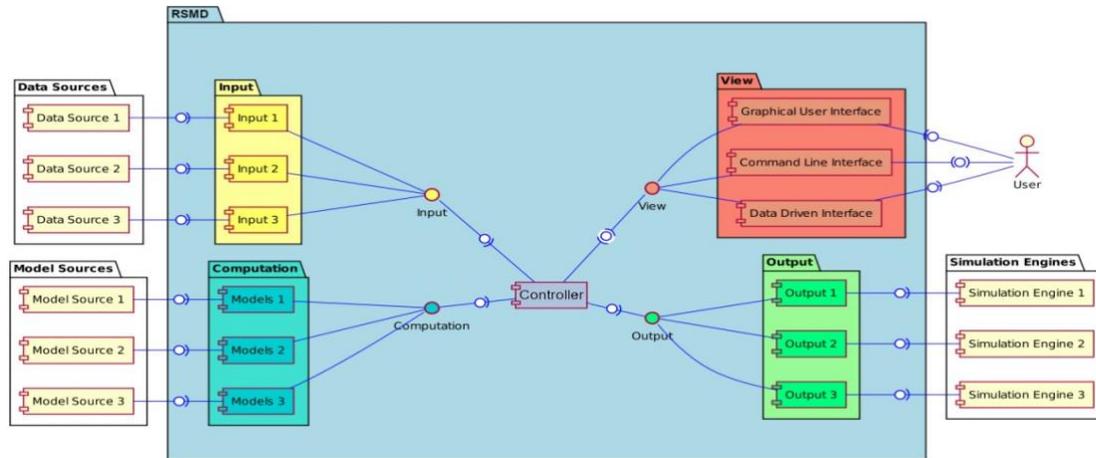


Figure 3 RSMD MOSA Architecture

## ARCHITECTURE MODULES

### UI Module

This module provides a graphical tool designed to help the modeler to start the modeling process. Typically, before the modeling process starts, the simulation modeler should have the necessary knowledge of the target simulation engine and should know which data source is needed to use to get model data. In addition, modelers also know which entity they are building physical models for. For example, a OneSAF physical model engineer knows that the data source comes from CDCC, the simulation engine is OneSAF, the entity they are building models for is a M-16 rifle, the physical model they are building is a Direct Fire Weapon Accuracy model. Instead of using the traditional way to build this weapon accuracy model, RSMD's UI Module is designed to start the process with a series of queries to verify if this model already exists. If not, then it will list similar models and associated computations to suggest reuse. If nothing is suggested, then the modeler will be able to build a new model from scratch using available data and existing methods developed for similar models.

### Input and Output Module

RSMD Toolkit works with data. It takes different formats of data as input to produce data in different targeted formats. Therefore, it is essential to ensure the design and the tools used in the design have the flexibility to handle various data formats. Figure 3 shows a simplified view of data flow through the Input and Output modules. Requests for data from the controller are handled by the input module, which gets and processes the data from the data source. The processed data is passed to the output module and exported for use in the specified simulation engine (e.g. simulation specific file).

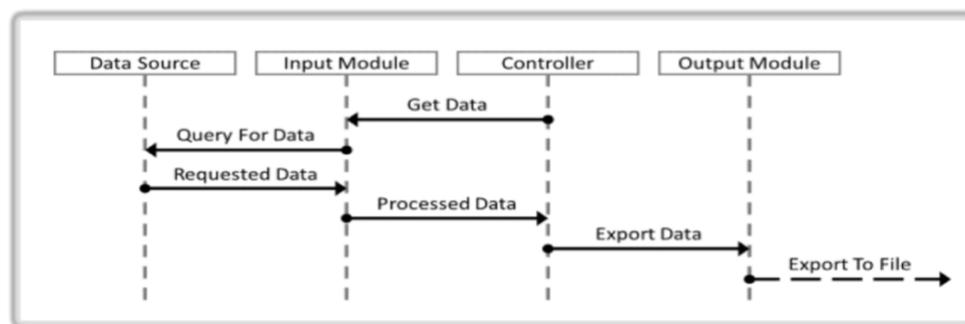


Figure 4 Input and Output Modules

### Computational Module

Computational modules are implemented based on authoritative data sources (e.g. PKADs). To ensure modularity and increase reuse potential, each module implements the logic of a single PKAD and makes that logic available to the Controller to processing data. The Controller gets data from the data source via the input modules, input data gets used by the computational logic, and the specified simulation engine gets its desired results.

### Controller Module

To facilitate a modular design, direct connections between modules are minimized in RSMD, with the Controller handling module communication by passing data between the modules. When exporting data, the controller module works with the output module to iteratively retrieve and process required data for export to the desired simulation format.

### CCDC MODEL DATA REPRESENTATION AND RETRIEVAL

CCDC’s primary mission was to plan and conduct major systems analyses of proposed and existing US Army weapons/material systems, programs and support throughout their life cycle. It is one of the most recognized authoritative data sources for Army MS&T. CCDC data has its own well-organized format. It is typically in a colon-separated format defining column names and values for all the data types. Digging into native format of CCDC data every time while building a physical model is not efficient. To be efficient, tools and processes must be developed to import and ingest the data. Adopting the OneSAF’s KA/KE process, RSMD follows the procedural steps to dissect and digest the data structures of each required physical model specified in PKADs. PKADs also specify where the supplemental materials are located if additional definitions and assumptions are needed. PKADs are considered validated physical model products because all contents have to go through successful peer review in the knowledge engineering environment by KA/KE engineers and SME. A typical PKAD provides a wealth of model data, containing almost everything needed to build ground truth of military equipment. It contains the Objective of the model to be built, input, output, data structures used for data representation, verification and validation collection data, algorithms for specific calculations, as well as references to other source materials. For example, in the PKAD named Delivery\_Accuracy\_for\_Direct\_Fire\_Weapons, a specific algorithm is designed to calculate the Range Detonation Point(Figure 4):

$$\text{Range Detonation Point [meters]} = \left( \frac{\text{Height of Round}}{\tan(\text{Angle of Fall})} \right)$$

Figure 5 Sample PKAD Validated Algorithm

Each of the inputs, such as Height of Round, is defined earlier in the same document. Some data is available via look-up tables, such as Angle of Fall which requires system, weapon, mount position, specific munition type, and range to get its value with an additional equation being provided for interpolating values between ranges in the look-up table.

Table 16. Example Angle of Fall Data.

System	Weapon	Mount	Munition	Range	Angle of Fall
ICFullyLoaded	grenadeLauncher40mmM203	proneSupported	grenade40mmHeM406	400	32
ICFullyLoaded	grenadeLauncher40mmM203	proneSupported	grenade40mmHeM406	500	41

Figure 6 Example of Angle of Fall Data

Following the OneSAF physical modeling process, we realized that a tool was needed to help the RSMD users understand how these data is structured and how to use it efficiently. Our survey research resulted in a decision to use DARPA's PDS. PDS is a GOTS tool that containing CCDC verified data stored in MongoDB. It provides a set of UIs to view and retrieve CCDC data using a spring boot front end application and GraphQL back end, respectively.

### Data Retrieval from PDS

DARPA's PDS provides two APIs to retrieve model data from MongoDB, where CCDC data is hosted. General entity data and complex data are kept in tables with tabular format, as specified in PKADs. Parsing and filtering methods are being developed with anticipated requirements from the Computational Module.

Our retrieval methods work with both APIs to get model data parsed and stored. Figure 6. shows a sample query for retrieving weapon data (left) and the returned results (right).

```

1 query getWeaponDetails($id: ID!) {
2   getWeapon(id: $id) {
3     name
4     displayName
5     description
6     minRange (unit: METER)
7     maxRange (unit: METER)
8     munitions { displayName }
9     country
10    length
11    height
12    weight
13    capabilities { name displayName description }
14  }
15 }

```

```

{
  "data": {
    "getWeapon": {
      "name": "5.56mm M16 Rifle",
      "displayName": "5.56mm M16A2",
      "description": "5.56mm M16A2 Assault Rifle",
      "minRange": 0.5,
      "maxRange": 550,
      "munitions": [
        {
          "displayName": "5.56mm, Ball, M193"
        },
        {
          "displayName": "5.56mm, Ball, M855"
        }
      ],
      "country": "US",
      "length": 1.0067,
      "height": 0,
      "weight": 2.89,
      "capabilities": [
        {
          "name": "DAMAGE.SHOOT.AIMED_FIRE",
          "displayName": "Direct Fire",
          "description": "The ability to fire a weapon by aiming the weapon to a target that can be seen by the shooter."
        }
      ]
    }
  }
}

```

Figure 7 Internal Data Representation

### CASE STUDY

In this case study, our goal was to test the POC framework of the RSMD Toolkit to study how all components work together to produce simple validated weapon data, and how this data can contribute to a more accurate result in a simulated scenario.

We chose VR-Force as our targeted simulation engine. VR-Forces has a relatively intuitive user interface, and we can build a simple Dismounted Infantry (DI) shootout scenario quickly. VR-Forces also uses a simple textual format to specify its weapon accuracy data used when calculating the Probability of Hit in their scenarios. This provides a perfect opportunity for us to insert RSMD produced weapon accuracy data into VR-Forces' model data files, so during scenario execution our weapon data can be used to calculate the Probability of Hit. Running the exact same scenarios twice, with different data sets, we will collect measurements on Probability of Hit. The measurements allow us to understand how the RSMD data, which uses CDCC calculations, should be applied, refined and adopted to better serve CGF physical modelers.

### Simulation Exercise Setup

There are three workstations used in this experiment. One is used to host PDS and MongoDB for input source data, one is used for RSMD, and another one to host VR-Forces. Figure 7. shows the exercise setup.

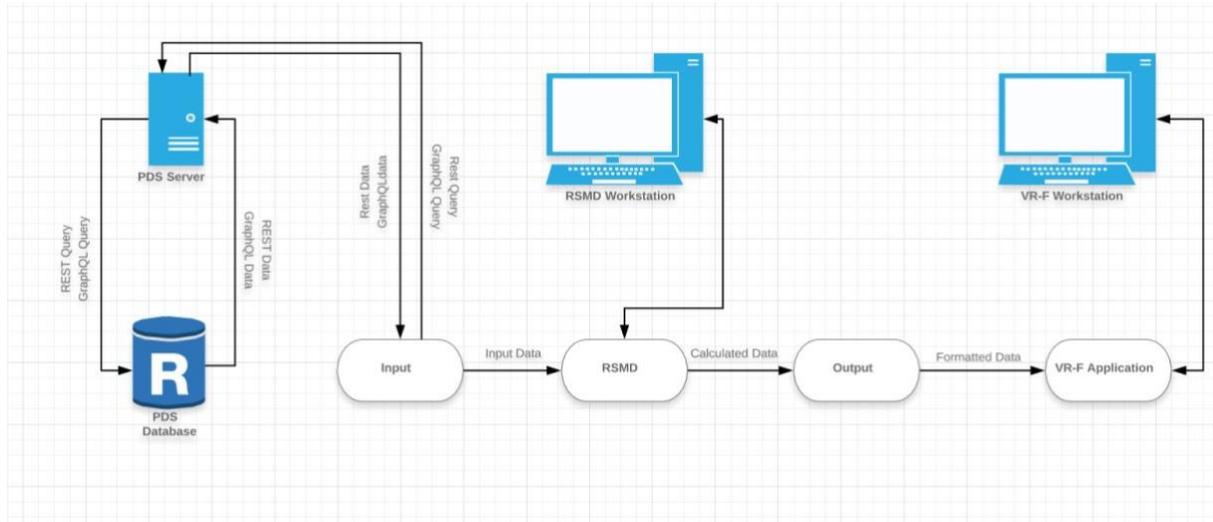


Figure 8 Simulation Exercise Setup

### Workflow Example

The task for the model developer is to build a model that returns Probability of Hit for the M16A2. After the modeler chooses the target simulation system, specific weapon type, and the type of entity, the required output data is displayed and populated with lists of existing methods and data that are associated with the weapon type. Once the accuracy model and target type is selected, the modeler can test the model and export the data. Figure 8. shows an intermediate UI to facilitate the workflow.

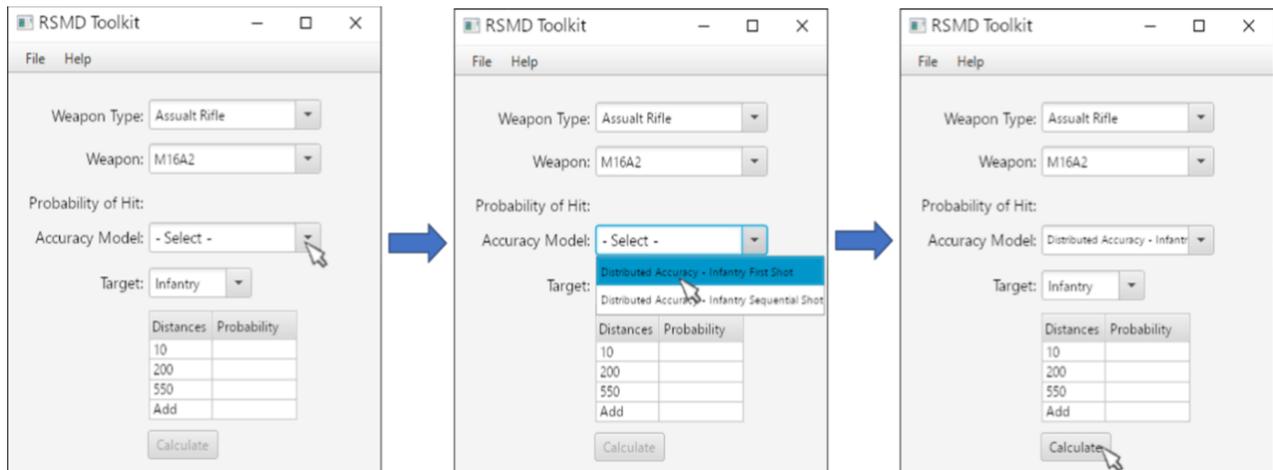


Figure 9 Intermedia UI Example

In this experiment, the model we built is based on the *Delivery\_Accuracy\_for\_Direct\_Fire\_Weapons* PKAD which contains all data and algorithms needed to calculate a Linear Miss Distance (LMD) distribution for the M16A2 based on distance to the target. Of the twenty algorithms in that PKAD, only half are used to calculate the LMD distribution. Using the LMD distribution, the Probability of Hit is calculated for the target size and distances to target provided by VR-Forces. The resulting values were formatted to comply with VR-Forces' data format so it can be consumed by VR-Forces during scenario execution. Figure 10 shows Probability of Hit data for both.

All methods created for the *Delivery\_Accuracy\_for\_Direct\_Fire\_Weapons* are stored in an internal repository in the controller module. This ensures that in the future when a similar model needs to be built, RSMD toolkit will direct the modeler to reuse most of the methods that already exist.



similar methods to use. In a long run, the model development cost will significantly decrease. For example, once M16 models are built, the modeler can build AK-47 models by simply reusing M16 models, with minor changes to its associated methods. If a different target simulation engine needs the same model data but in a different output format, only a new output module needs to be developed.

Although this experiment only uses one source data and one target simulation engine, the RSMD toolkit shows the potential of multiple source data and simulation engines can be supported because of its modular architecture.

## **FUTURE WORK**

### **Model Testing on Different Simulation Engines**

To be able to test the modularity and reusability of the RSMD framework, additional simulation engines should be used. A virtual simulation trainer that can simulate weapons is desirable because the current available models and methods in the repository can be tested for reusability.

### **SME Involvement**

In CGF model development process, SMEs were always a part of the testing, verification and validation (V&V) team. They should be working with model developers during intermediate testing as well as Government Acceptance Test (GAT). No models shall be considered validated before GAT is performed. Therefore, it is imperative that future model development will bring SMEs on board for V&V.

### **Context-Based Reasoning (CBR), Machine Learning (ML) and Artificial Intelligent (AI)**

As the model repository grows when more and more models are created and stored, it will become a database of validated CGF physical models. Data scientists and software engineers should be able to develop advanced algorithms to make the development process more efficient by inserting technologies such as CBR, AI and ML.

## **CONCLUSION**

Using CGF simulation systems for military training and analysis depends on the credibility of CGF models and how the simulation engines use these models to provide training rigor. In this paper we presented a concept of Rapid Simulation Modeling Development toolkit to facilitate simulation modelers to effectively create CGF physical models with validated data and algorithms. Our goal is to make this toolkit not only enforce the use of authoritative data, but also to set up the foundation for future reuse of developed models. We believe that reusability can be achieved by providing a repository, and a set of filtering methods, so the modelers can start development by searching the repository for reusable models.

We designed the RSMD framework to be modular to allow multiple data sources and simulation engines to be plugged in and removed as needed. In our experiment, Army's CCDC data source was used to create a direct fire weapon accuracy model using data and algorithms defined in PKADs. VR-Forces is used as the target simulation engine to test the RSMD generated weapon data. The case study result is promising because the accuracy of the weapon used in the test scenario increased to be more realistic.

Another inspiration of creating such a toolkit is to encourage reuse in CGF modeling. The CGF Physical modeling process has always been resource intensive because of the complex data representation and

mathematical implementation. Currently, most of the military simulation systems were built with specific requirements, different goals, unique design philosophies and implementation choices. None of the simulation-specific models can be reused without getting into the complex entanglement of legacy code. The RSMD's Computational Module was designed with reuse in mind. To ensure modularity and increase reuse potential, each module implements the logic of a single PKAD. All methods associated with each module are developed with an in-depth understanding that these methods might be used by different simulation engines in the future. Searching and filtering methods are being developed to help the simulation modeler find reusable models efficiently.

As the RSMD Toolkit matures, and modeling with authoritative data process is adopted by more simulation modelers, we expect the model repository will grow, and the reusability of the tool will be well tested and greatly enhanced. In addition, with more simulation engines using the toolkit to develop models, the CGF modeling community will finally be able to break the barrier of not being able to develop a common set of physical models to be used by multiple simulation systems.

## **ACKNOWLEDGEMENT**

Thanks to Dr. Buck Surdu for introducing DARPA's PDS to the RSMD development team. Thanks to Ms. Carol Byers-Bendle for helping Infinitas get the CACCTUS database DA in place. Thanks to Mr. Peter Swan and Ms. Deb Fulford for helping set up VR-Forces in our lab.

## **REFERENCES**

- Monday, Paul, "The Expected Results Method for Data Verification", *proceedings of IITSEC, 2015*.
- Lt. Col. R. Armstrong, Dr. M. Bailey, "CACCTUS: Linking the Live, Virtual, and Constructive Environments", *proceedings of IITSEC, 2004*.
- Oijen, van Joost, "Effective Behavior Modeling for Computer Generated Forces", *proceedings of IITSEC 2015*