24th International Congress on Modelling and Simulation, Sydney, NSW, Australia, 5 to 10 December 2021
mssanz.org.au/modsim2021

Integrating and automating a closed loop simulation capability to enable easier exploration of the parameter space

S. Kiparoglou[a] **and D. Coppola**[b]

[a]Defence Science and Technology Group, [b]Consilium Technology
Email: Stefanos.kiparoglou3@dst.defence.gov.au

Abstract: This paper discusses the implementation of the integration and automation of an analytical simulation capability that consists largely of multiple top level components and user modified tools. These disparate tools create configuration burdens and limit the exploration capability of the wider system to answer a complex question as part of a larger problem. The aim of this project is to develop a capability to explore a wide set of performance characteristics and configurations within a complex simulation environment and perform multiple forms of analysis with little to no human intervention. This is achieved by using a microservice design that can encapsulate the closed loop simulation experiment lifecycle and provide the study lead with a simple way to generate and execute a simulation, extract metrics, perform analysis and provide meaningful insights. The design will take significant steps towards allowing the study lead and associated analysts to focus more on experimental design, verification, validation and sensitivity of data but also introduce new capability options in terms of execution and analysis.

Keywords: *Integration, simulation, automation, parameters, data*

1. INTRODUCTION

The Land Capability Analysis (LCA) branch within the Defence Science & Technology Group (DSTG) maintains an analytical simulation capability in the Combined Arms Simulation (CAS) group. This group maintains and develops an extensive chain of tools and techniques, from individual platform performance modelling, performance database generation, entity behaviours, simulation configuration and results analysis, which are all used throughout the study lifecycle.

Figure 1 depicts the typical study development process that follows the problem scoping phase where performance data entities including weapons, munitions, sensors and platforms are generated in addition to closed loop simulation parameters including formations, aggression levels and schemes of manoeuvre. The execution phase is focused on closed loop simulation of multiple replications of a scenario and analysis can be one of many types of mathematical techniques used to gather outcomes for a client.

Figure 1. Study Development Process. (John 2021)

In a typical study environment, requirements or new information can be provided anytime during the scoping, execution or analysis phase and hence studies are often behind schedule due to the large amount of work involved. This can result in a shortened closed loop simulation run or analysis period.

Close combat has proven difficult to analyse with any single analytical technique (Bowden 2013, Bowley 2003, Labbé, et al 2006). The ability to easily explore a wider parameter space across the study and gain easily accessible results in a timely fashion is of great importance to an analyst. Currently there are multiple aspects that impact the effectiveness across the lifecycle of a study, including but not limited to sharing, traceability, execution, scheduling and a heavy reliance on deep user knowledge. Integrating and automating the disparate tools and elements of our combat simulation environment provides a consistent study framework to reduce errors and allows for wider parameter space exploration.

This paper describes the design and implementation of a service orientated application that enables the integration and automation of a set of simulation components into a single system allowing for a streamlined analytical environment resulting in a reduction in resourcing required for a complex study exploration. The paper will refer to this integration and automation service as the Simulation Integration (SI) environment.

2. BACKGROUND

Currently the CAS analytical simulation capability exists as a set of independent top level components and secondary user modifiable tools as seen in Figure 2 which perform various roles within a combat simulation study. The problem here is that the designers and analysts must work closely together to ensure each component and tool is configured appropriately from data generation via a Simulation Repository (SimR) (Holden et al. 2016) in the design phase through to metric extraction via the analysis merge. The top level components which are larger applications and services that make up the backbone of the simulation capability. The user modified tools are smaller configuration tools that SI leverages and serve a specific purpose; they are often scripts used for filtering and aggregation that transform data in a way that enables further investigation. Many tools utilize the R statistical programing language which is designed for data analysis and mining as well as providing graphical representations of data and results.

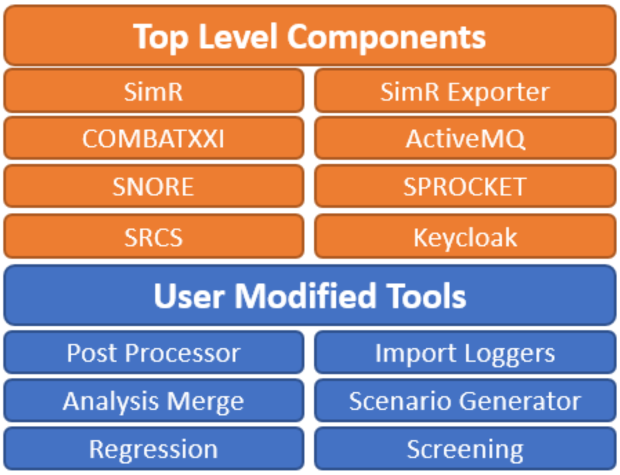

Figure 2. Components and Tools

The top level components are briefly described below:

- SimR is a central repository that stores simple, verifiable fundamental attributes for systems including weapons, munitions, platforms and sensors. Holden et al (2016)
- SimR Exporter is a data generation tool containing a set of algorithms based on physics used to calculate complex data for a given simulation. Holden et al (2016)
- COMBined Arms Analysis Tool for the 21st century (COMBATXXI) is a high-resolution, closed-form, stochastic, analytical simulation tool created by United States Army division of The Research and Analysis Center. TRADOC Analysis Center (2021)
- ActiveMQ is an open source message broker service used for distributed processing of large tasks. Apache (2021)
- Study into the Number or Replications for Experimentation (SNORE) is an analytical tool used to establish a robust, reliable and accountable method for the minimum number of replications required to produce statistically valid results. Schultz et al (2019)
- Simulation Pattern Recognition and Outlier Classification for Key Events Tool (SPROCKET) is an analytical tool used for pattern recognition and anomaly detection in simulation event log data. Schultz et al (2020)
- Statistical Ranking Colour Scheme (SRCS) is a parametrized pairwise comparison tool. Villacorta (2015)
- Keycloak is an open source identity and access management solution utilised by SimR

3. SYSTEM DESIGN

3.1. SI Requirements

Given the problem outlined above the first steps involved undertaking a complete review of current components, tools, processes and methods used in a study before identifying improvements that would aid the analyst rather than just stitching together a set of components. A series of workshops were used to generate a requirements list and expand into a more ideal future focused environment that not only integrates disparate tools and systems but also automates as much of the simulation tool chain as possible.

SI should allow an analyst to focus on the assumptions and design of experiment, analysis of results and the verification and validation of data rather than the mechanics of execution and configuration of components. It should also give the possibility for exploration of areas otherwise not possible including sensitivity analysis and expand the simulation space by providing the capability to use multiple closed loop simulation engines.

Finally, it was important to provide an easy to use interface where all aspects of SI can be controlled and have this accessible by multiple analysts.

In addition to the integration and automation of features, the SI tools seeks to address limitations we currently face with designing experiments within our simulation environment. For example, if one wishes to explore the impact of changing the range of a particular weapon within different environmental conditions and different tactical approaches, it currently requires manual synchronisation between teams to produce the appropriate databases, edit the entities within the combat simulation and build correct scenarios, which is time consuming and prone to error. Instead, we see the SI tool facilitating building such a design, exposing relevant parameters in a single interface regardless of their source and implementation.

A lengthy list was produced following these sessions with the key requirements of SI outlined below:

- Centralized HMI
- Integration and automation of all top level components and user modified tools
- Modifiable performance data
- Modifiable closed loop simulation execution data
- Study sharing
- Process Scheduling
- Ability to easily implement future closed loop simulations

The requirements in essence point towards and underlying reduction in human involvement via integration and automation which is critical to complete analysis and provide results for a client within reasonable timeframes. It was determined that where parallel execution is possible it should be performed to fully utilise resources and reduce overall study time.

3.2. SI Design and Implementation

The SI requirements are largely addressed using a microservice architecture which follows the software development of a single application utilising several small, loosely coupled services. Using a microservice means each service can be deployed as a separate unit and accessed via an Application Programming Interface (API) using a variety of protocols. Microservice architecture has advantages when compared to the more traditional monolithic architectures where decoupling of components is managed within the application but in the same process. These include scalability, optimisation, maintainability and reliability.

Figure 3 shows the individual service components for each state in the simulation study lifecycle. The SimR export of the performance database and the subsequent scenario executions within the combat simulation are the most time intensive tasks and are typically parallelised. Scenarios are considered to be a description of the physical, enemy and friendly environments for which a simulation can occur (Bowley 2004 and Pincombe et al 2013). These tasks are services outside of the scope of SI, but are implemented in a method to reduce build and execution time through the realisation of data caches and compute farms.

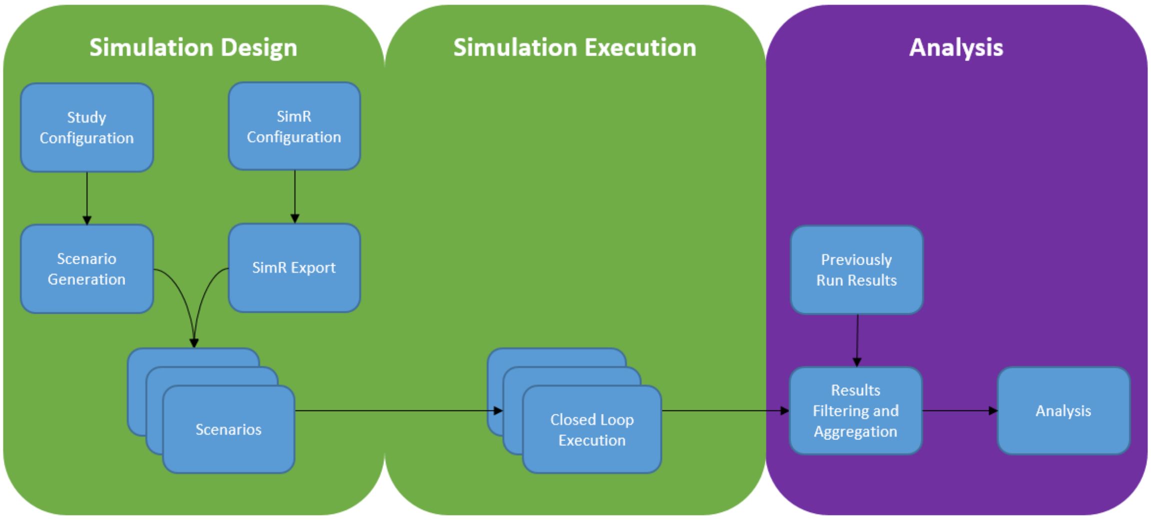

Figure 3. Individual Service Components comprising the simulation execution process

Given the service architecture, once the scenarios are sent to be executed, configuration of another iteration of SimR Export and Scenario Generation can commence. This permits the longest simulation processing tasks to be carried out in parallel of the execution phase.

To permit entity modifications to the key performance data characteristics an API provides access to entities from SI within SimR. It allows a set of values to be applied to a characteristic depending on its nature and then subsequently the raw performance data is sent to the SimR Exporter which passes the data through a set of algorithms and generates entity performance data for a given simulation.

Closed loop simulation execution attributes include tactics, behaviours, environmental conditions and the number of replications to execute a scenario. Using SI an analyst can provide these parameters via the HMI and have the simulation execute as many times as required.

Figure 4 outlines the functional diagram of the various services that comprise SI. This architecture is easily updatable and dynamic and the service oriented architecture makes it easy to diagnose faults and decouple services for development, testing and deployment.

The centralized HMI is a web service providing front-end graphical functionality. It allows access to each of the provided components, services and persistent data through a GraphQL (Facebook Inc 2015) server, providing a consistent query and mutation interface for all actions.

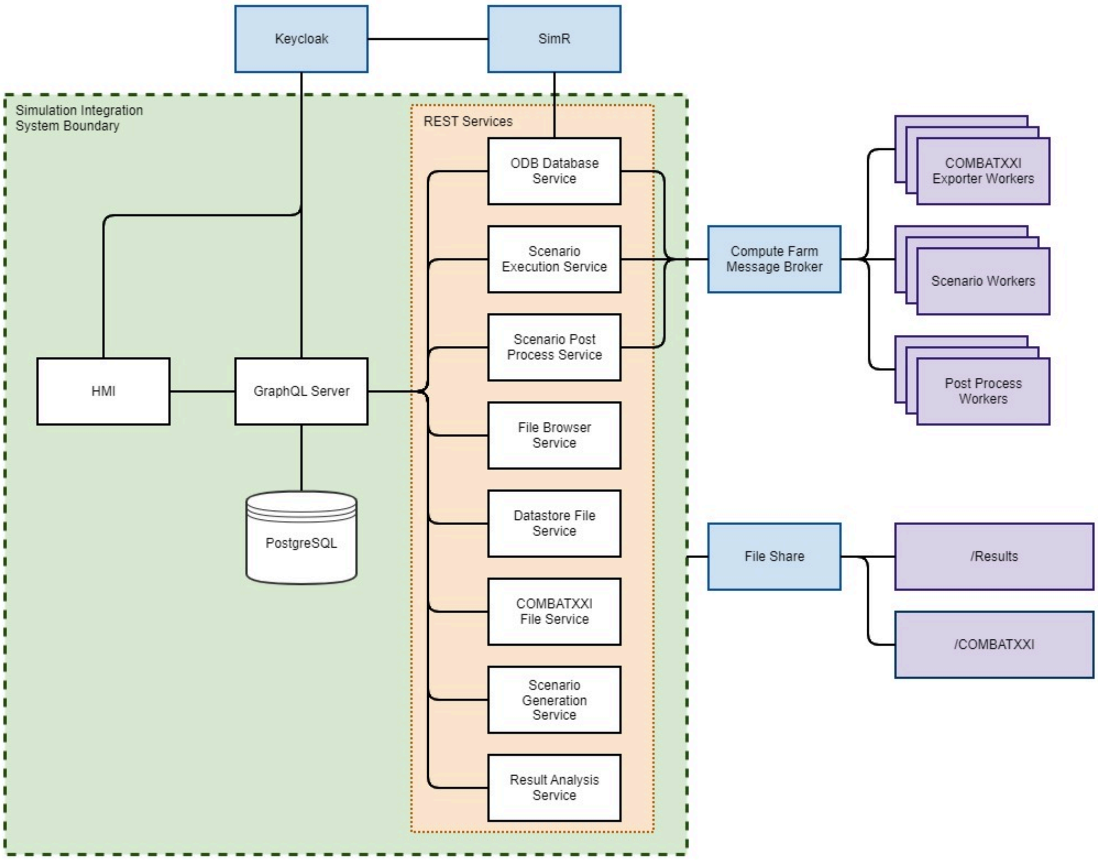

Figure 4. Functional Diagram

The GraphQL server validates an analyst via a JSON Web Token (JWT) with Keycloak, providing single sign capability across SI and SimR. Using this method an analyst can share a study with other team members and provide full access to the HMI. This means a single permission model can be used to restrict user access within the integration system to permissible data within SimR. This permission model is extended to applicable services through the use of the JWT.

As all services are stateless by design; as each service executes its function within the experiment lifecycle, the state is maintained within the PostgreSQL database (PostgreSQL Global Development Group 2021). This internal state forms the basis of the business logic to synchronise tasks to ensure consistency and repeatability of the results as well as maximise efficiency of the executing tasks.

Top level components and user modified tools are integrated in an efficient manner using microservices that allows SI to be easily updatable for future versions of each component. This includes updated software versions of components or the applications of additional future closed loop simulations. The microservice architecture can also make reuse of existing scripts in their native environment. For example, extant functional components written in Python or R could are reused by incorporating a REpresentational State Transfer (REST) interface and deploying them as a service. This increases reliability and confidence as the components have been validated through their previous use within the study lifecycle.

SI uses a file share to store scenario input, output and analysis data. This data is referenced by a PostgreSQL database to provide the connection between data, metadata and logical associations between performance databases, scenarios, executions, results and analyses.

The HMI, GraphQL Server, PostgresSQL database and each service within the SI boundary are packaged as individual containers. This permits a range of cloud based deployments and redundancy options. In this case a simple docker-compose orchestration is sufficient based on the initial number of users of the system.

4. DISCUSSION

Once of the largest challenges posed to a study designer or analyst when confronted with a problem is the exploration of key experimental parameters, in this case simulation performance data or closed loop simulation execution parameters. Pre SI an analyst would be required to manually modify and configure these characteristics and manage the administration burden that comes with performing such a study.

SI has enabled the exploration of a set of data points with ease at any stage of the study lifecycle. For example, if it has been determined that the muzzle velocity characteristic of a munition needs to be investigated to accurately determine a munitions lethality against a platform, SI can give the analyst the opportunity via a shareable and easy to use HMI the ability to do so. SI can perform a simple exploration of that characteristic and have that executed in a new simulation cycle without the need for a data analyst to revisit that characteristic. Moreover, our vision is that SI will be able to include parameters from across the simulation ecosystem, combining performance, behaviour and environmental parameters into a study design without manually managing the data across disparate tools.

We believe that the design of SI will enable more complex design of experiments techniques, in particular iterative techniques as discussed by Chau et al (2017). The more automated simulation to analysis loop also creates an environment more amenable to machine learning techniques such as reinforcement learning, supervised learning or human-on-the-loop techniques. SI fully completes the study lifecycle loop to allow rapid iterations with large data sets that can be executed with as much computer power as possible, allowing more efficient use of high performance computing.

5. CONCLUSION

Simulation Integration has enabled study designers and analyst of the Land Capability Analysis branch to achieve several positive outcomes in regards to an analytical study problem. With the introduction of a microservice architecture and REST interface components and tools have been successfully integrated and automated. A study lead is now able to:

- Focus more on the assumptions and design of experiment
- Focus more on the analysis of results
- Focus more on the verification and validation of data
- Explore areas otherwise not possible of feasible
- Share study visibility with multiple analysts
- Add additional simulation execution capability
- View and control the study via a HMI

The achievements above are derived from a set of requirements that were drawn from workshops and brainstorming sessions that not only achieve a short term outcome but enable SI to be adaptable to future study conditions.

ACKNOWLEDGEMENTS

Denis Shine and Darren Lohmeyer for their roles in the leadership of the design phase of the project.

Members of the Land Capability Analysis branch of Joint and Operations Analysis Division.

REFERENCES

Apache, Apache ActiveMQ, 2021, Apache ActiveMQ Reference Guide, viewed 30 August 2021, http://www.ttmsolutions.com/Apache_Software/ActiveMQ_Reference_Guide.php

Bowden, F, Williams, P, 2013, A Framework for Determining the Validation of Analytical Campaigns in Defence Experimentation, MODSIM 2013, Adelaide, Australia. DOI:10.36334/modsim.2013.e1.bowden

Bowley, D, Castles, T, Ryan, A, 2003, Constructing a SUITE of Analytical Tools: A Case Study of Military Experimentation, Australia Society for Operations Research Bulletin, 22(4), 2-10.

Bowley, D, Lovaszy, S, 1999, Use of Combat Simulations and Wargames in Analytical Studies Battlelab, Proc SimTecT 99, Paper 11

Chau W, Gill A.W & Grieger D, 2017, Using Combat Simulation and Sensitivity Analysis to Support Evaluation of Land Combat Vehicle Configurations, MODSIM 2017, Hobart, Tasmania. DOI:10.36334/modsim.2017.d1.chau

Facebook Inc, 2018, GraphQL, viewed 30 August 2021, http://spec.graphql.org/June2018/

Holden, L, Dexter, R.M, Shine, D.R, 2016, SimR: Automating Combat Simulation Database Generation, Defence Operations Research Symposium 2016, Canberra, Australia DOI: 10.1007/978-3-319-55914-8_21

John, M, 2021, Provided via personal communications with creator, Defence Science and Technology Group, Australia

Labbé, P, Bowley, D, Comeau, P, Edwards, R, Hiniker, P, Howes, G, Kass, R, Morris, C, Nunes Vaz, R, Vaughan, J, 2006, Guide for Understanding and Implementing Defense Experimentation GUIDEx, The Technical Cooperation Program. 10.13140/2.1.4937.6648.

Pincombe, B, Blunden, S, Pincombe, A, Dexter, P, 2013, Ascertaining a hierarchy of dimensions from time-poor experts: Linking tactical vignettes to strategic scenarios, Technological Forecasting and Social Change, 80(4) pp. 584-598. DOI: 10.1016/j.techfore.2012.05.001

PostgreSQL Global Development Group, 2021, What is PostgreSQL, viewed 30 August 2021, https://www.postgresql.org/about/

Schultz, L, Chiera, B, John, M, 2019, Co Operative Research Agreement into Land Simulation Event Analysis (CORALSEA): The Extended Study into the Number of Replications for Experimentation (SNORE) and the SimRep tool for Replication Selection, University of South Australia, Final Report to the Defence Science Technology Group.

Schultz, L, Chiera, B, 2020, Simulation Pattern Recognition and Outlier Classification for Key Events Tool (SPROCKET) – Phase 2, University of South Australia, Final Report to the Defence Science Technology Group

TRADOC Analysis Center, 2021, COMBATXXI, TRADOC Analysis Center, viewed 30 August 2021, http://www.trac.army.mil/COMBATXXI.pdf

Villacorta, P.J, Saez, J.A, 2015, SRCS: Statistical Ranking Color Scheme for Visualizing Parameterized Multiple Pairwise Comparisons with R, R Journal, **7**(2) DOI: 10.32614/RJ-2015-023