Development of closed-loop military simulation software: Challenges, best practices, and lessons learned

Erkin Çilden, İsmail Çağlar Çetintaş, Denizcan Demirok, Büşra Toraman, Levent Hakkı Şenyürek, Fatih Narman, Merve Çağlar Erdemir, Ahmet Sezer

STM Defense Technologies Engineering and Trade Inc.
Ankara, Türkiye

{erkin.cilden, icaglar.cetintas, denizcan.yilmaz, busra.yilmaz, lhakki.senyurek, fatih.narman, cerdemir, asezer}@stm.com.tr

Halit Oğuztüzün

Middle East Technical University,
Department of Computer Engineering
Ankara, Türkiye

oguztuzn@ceng.metu.edu.tr

ABSTRACT

Closed-loop military simulation software, or automated constructive simulations, differ from the more widely known human-in-the-loop counterparts, namely computer-aided war game simulations, by modeling of human decisions integrated into the simulation and executing without human stimulation at the run-time. That is, unlike conventional computer-aided war games, the purpose of closed-loop military simulations is not to train users in the decision-making role, but to conduct an in-depth investigation of the complex nature of the target phenomenon being simulated. Thus, there is a critical difference regarding their intended use. This difference brings about unique challenges and needs to be managed in various aspects during implementation.

In terms of model management and validation, some issues standout such as managing the complexity of the target phenomenon at the model level, determining and applying the appropriate autonomy level of the models, and aligning model fidelities with validation methods in accordance with the intended use. In terms of software engineering and data management, questions should be answered for simulation big data processing, analysis and visualization needs, alignment of different aspects of software architecture, and flexible capabilities to support these needs by the simulation core. From a project management perspective, predictability problems may be encountered in both daily operations and long-term planning. From the user perspective, stakeholder involvement is important in the adaptation of the tool, which can be considered as an interpreted formal language and is usually not easy to practice without particular training.

In this paper, distinguishing aspects that the project team should focus on during the development of closed-loop military simulation software are discussed. Based on the authors' experience of a project carried out over a period of more than five years, the identification of these problems, alternative and proposed solution methods, and lessons learned are presented.

ABOUT THE AUTHORS

Erkin Çilden obtained his B.S., M.S., and Ph.D. degrees in Computer Engineering from Middle East Technical University, and currently is a group leader and senior software engineer at STM. His research interests include multi-agent systems, reinforcement learning, and simulation software design.

İsmail Çağlar Çetintaş obtained his B.S degrees in Computer Engineering from TOBB Economics and Technology University in 2010 and is currently working as lead software engineer in Decision Support Systems department at STM. He has more than 10 years of experience in the defense industry. His expertise includes embedded software, real time operating systems, multi-agent systems, and simulation software design.

Denizcan Yılmaz has a B.S. degree in Computer Engineering from TOBB Economics and Technology University, and currently working as an expert software engineer in Decision Support Systems department at STM. Her research interests include image processing with deep learning methods, big data management, multi-agent based simulation systems, user interface designs of simulation tools.

Büşra Toraman has B.S. and M.S. degrees in Electrical and Electronics Engineering from Hacettepe University and is currently working as senior software engineer in Decision Support Systems department at STM. Her expertise and research interests include simulation modeling, validation and verification of sensor systems, and infrared signature analysis.

Levent Hakkı Şenyürek has B.S. in Electrical and Electronics Engineering from Middle East Technical University and M.S. degree in Systems & Control Engineering from Boğaziçi University, and currently is a group leader and senior modeling engineer at STM. His expertise and research interests include autonomous behavior of military agents, command and control systems modeling, and training & analysis simulations.

Fatih Narman obtained his B.S degree in Electronics Engineering from Istanbul Technical University and is currently working as a senior systems engineer in STM. His expertise includes simulation modeling, planning, design, verification, validation, development, and implementation of complex systems.

Merve Çağlar Erdemir has B.S in Electrical and Electronics Engineering from Kırıkkale University and M.S degree in Engineering Management from Hacettepe University and currently is a project manager at STM. She has Project Management Professional (PMP) certificate. Her expertise includes planning, budgeting, risk management, communication, and stakeholder management.

Ahmet Sezer has a B.S. and M.S. in Mechanical Engineering from Middle East Technical University and currently is the manager of Decision Support Systems department at STM. His expertise includes simulation modeling (weapon systems, flight dynamics) and accreditation.

Halit Oğuztüzün is a professor in the Department of Computer Engineering at Middle East Technical University (METU), Ankara, Turkey. He obtained his BS and MS degrees in Computer Engineering from METU in 1982 and 1984, and his Ph.D. in Computer Science from The University of Iowa, Iowa City, IA, USA in 1991. His recent research activities include distributed simulation and model-driven engineering.

Development of closed-loop military simulation software: Challenges, best practices, and lessons learned

Erkin Çilden, İsmail Çağlar Çetintaş, Denizcan Demirok, Büşra Toraman, Levent Hakkı Şenyürek, Fatih Narman, Merve Çağlar Erdemir, Ahmet Sezer

STM Defense Technologies Engineering and Trade Inc.
Ankara, Türkiye

{erkin.cilden, icaglar.cetintas, denizcan.yilmaz, busra.yilmaz, lhakki.senyurek, fatih.narman, cerdemir, asezer}@stm.com.tr

Halit Oğuztüzün

Middle East Technical University, Department of Computer Engineering
Ankara, Türkiye

oguztuzn@ceng.metu.edu.tr

INTRODUCTION

The theory and practice of computer-assisted military exercise technologies have been extensively studied for decades. Modeling and simulation is an interdisciplinary area that has been studied in detail as an enabling technology. Although there is widely accepted literature on definitions, terms, and concepts to provide a common ground for academicians and practitioners, some notions are still being discussed.

In the military context, *wargame* commonly refers to the simulation of a military operation (Ministry of Defence UK, 2017; NATO, 2019, 2023). However, various definitions emphasize the different aspects of these common definitions, such as the existence of players of different sides, the existence of rules-data-methods-procedures, and the effects of decisions to the sequence of events (Hodický et al., 2020). There seems to be an uncertainty on the boundaries of the term, especially concerning the constraints on the role of players and the execution of the sequence of events. For the definition of *wargame simulation*, the situation is similar. As a common understanding, a computer-assisted wargame simulation is a multi-user software incorporated in a multi-player game setting, usually equipped with real-time models with appropriate levels of fidelity and aggregation, tools required to manage the execution, and features to support the evaluation and improvement of the players' real-time decision capabilities.

There is a less common category of a similar simulation technology, in which the players are not in-the-loop during the simulation execution, rather, a predefined scenario script is followed from the beginning to the end without interruption. A term used for this category is *closed-loop simulation* (Hodický et al., 2020; Turnitsa et al., 2022), or less frequently *automated constructive simulation* (ACS) (Hodicky et al., 2019; Nohel et al., 2022). In this paper, we use the term *closed-loop simulation* to mean that the computerized simulation model executes without user intervention, and the encapsulated models serve for military decision making in some sense, including but not limited to training, tactics evaluation, and technical effectiveness analysis.

This paper summarizes challenges, best practices, and lessons learned from our experience on a closed-loop military simulation development projects, examining the problem from different points of view, namely model engineering, software engineering, and project management.

RELATED WORK

Due to their characteristic flow-of-operation, closed-loop military simulation tools enable some interesting features. Since the execution flow of a single scenario run (sometimes also called: a trial) is not interrupted by the user, real-time execution is not a requirement. On the contrary, faster-than-real-time, or best-effort execution is usually an essential property of closed-loop military simulations, since it gives the opportunity to quickly repeat the trials with varying input parameter values, to observe and analyze how the outcome is affected by them. This gives the user to apply the Monte-Carlo method (Biggs & Hirsch, 2022), Design of Experiments (DoE) (Law, 2017), or similar techniques to generate statistically meaningful simulation outputs to make controlled use of stochasticity to understand the impact of input factors on the behavior of simulation models. Given a scenario and variable input parameter values of interest, these techniques provide the user with a simulation output space (response surface) by executing each variation (trial) one by one, and recording the outputs to the target data repository.

Closed-loop military simulations have various use cases. These tools are a perfect medium to support concept development and experimentation (CD&E), to capture the best ideas, gaps, and approaches for an existing or future issue, by using controlled investigation methods (NATO, 2021). They can support training by their inherent capability for *analysis of alternatives*. Procurement decisions can make use of data produced by closed-loop military simulation systems, where the decision maker can compare the performances of two alternate sub-systems by designing and executing experiments using the critical scenarios of interest.

Development of simulation software has some challenges and difficulties, in addition to commonly agreed software engineering practices. Some of the distinctive characteristics of simulation software development are, the emphasis on the design and development of simulation models together with their validation-verification processes, the criticality of alignment of software requirements with modeling requirements, and the importance of designing the correct simulation architecture and maintaining it throughout the development with minimal corrosion. These problems are known and methods to overcome these problems are well-studied in the literature (Gianni et al., 2011; Smith, 1998).

GEMED PROJECT EXPERIENCE

Figure 1 GEMED start screen

Naval Warfare Effectiveness Analysis Model (GEMED) is a closed-loop military simulation with built-in experimentation and effectiveness analysis features, designed and implemented for Turkish Naval Forces as the end-user. There are approximately 90 simulation models targeting naval warfare at the tactical/operational level, including naval and air platforms, RF/IR/acoustic sensors, weapon systems, and missiles. At its core, GEMED is a multi-agent based simulation, in which both discrete event triggers and variable time advance mechanisms are available for the agents, which gives the model engineer flexibility to use hybrid approaches if required. There are six main modules of GEMED (Figure 1): model management, scenario management, DoE-and-execution, reporting, debriefing, and effectiveness analysis (Özkan et al., 2019). All modules are independent at the process level but are interdependent at the data level.

The primary intended use of GEMED is tactical/operational analysis via scientific methods, using the outputs of simulation batch runs. GEMED was a research and development project with multiple phases developed between 2018-2023, and is currently in its maintenance phase. This paper presents the noteworthy aspects that attracted the attention of the GEMED team, based on their experience in the project, during the development of a closed-loop military simulation software.

MODELING PERSPECTIVE

Model Development and Implementation Aspect

Arguably the most predominant challenge in developing a closed-loop military simulation software is proper modeling of complex target phenomenon and managing their conceptual integrity, while paying close attention to the required autonomy of the models to enable closed-loop execution.

Multi-agent based design:
In a closed-loop military simulation, it is not feasible to assume that the model designer or scenario developer is able to foresee all the alternate futures within a defined simulation scenario. Multi-agent based simulation (MABS) (Abar et al., 2017) allows the scenario to be inherently built up from the bottom to the top. At the bottom resides the agents, which are ontologically separated run-time entities with encapsulated domains of interest (i.e. environments), and a certain level of autonomy. At its core, GEMED incorporates four environments: dynamic environment (Newton physics), acoustic environment (hydroacoustics), electromagnetic environment (electromagnetic spectrum), C2 environment (tactical/operational picture per agent). Each agent is entitled to at least one of these environments and interacts only with them (i.e. no two agents directly interact with each other). There are known design and performance advantages of MABS compared to other design approaches (Çetintaş et al., 2020).

Using MABS practices;

- it is easier to manage the modeling complexity of the target domain, both during the model design and at the closed-loop scenario definition level,
- since the target domain is complex, MABS is an effective approach to understanding the emergent behavior of the complex phenomenon,
- agent encapsulation makes it easier to isolate agents, so that the model designer can focus on certain aspects of the model without any regression on its other aspects

On the other hand, MABS;

- requires a certain model design mindset, where each agent should have a level of autonomy, which requires experience and is a design challenge per se,
- requires a careful design of the "environments" since agents only interact through them. In the model design point-of-view, an environment is the invisible part of an agent, and it is an art to balance the agent skills with the environment capabilities, taking into account the possible number of agents, the volume of common calculation routines, multi-fidelity requirements, and computational constraints (parallel execution, data distribution, etc.).
- requires the engineering of an acceptable level of agent autonomy, since higher levels of autonomy introduce the possibility of emerging behavior outcomes that are difficult to interpret analytically.

Multi-fidelity alignment:
In a simulation where there are no human players throughout execution, it is necessary to merge the mimicry of the physical phenomena with the required repetitive human decision, without loss of generality on the statistical simulation outputs. This can be achieved by lowering the fidelity of human decision centric models, while keeping the others as high as required. However, taking the computation efficiency constraint into account, the model developers have to cope with different levels of fidelities among models, which is not straightforward.

To overcome this difficulty, GEMED team committed to the careful practice of the MABS principles, especially agent-environment abstraction, which is a disciplined way to isolate each agent and each environment from each other. Through agent-to-agent isolation, model implementation and testing become independent of another model's development progress. Such an approach significantly reduces the development time of every model and increases the reusability of the model in other simulations. One downside of this approach is that environments are burdened by volumes of information about model data (positions, RCS, etc.), and model-environment dependency increases. Additionally, the team identified that multi-fidelity alignment should be addressed in close association with the model validation activities, where the end-users, developers, and domain experts should agree on a common understanding about the model fidelities. It is obvious that the process is not straightforward, and not always completed with an optimal alignment.

The multi-fidelity alignment;

- comes across as a requirement or a constraint for a closed-loop military simulation,
- can be achieved by careful application of MABS principles,
- usually requires the contribution of multiple stakeholders,
- benefits from early integration using mock models.

On the other hand,

- there is no agreed-upon measurement unit on fidelity level, so the alignment process requires careful consideration of multiple aspects with multiple stakeholders,
- multiple-fidelity modeling has potential adverse effects on consistency, such as repeatability, reliable emergent behavior, and smooth user experience.

Model Validation and Verification Aspect

The MABS approach provides a natural decomposition (Pěchouček et al., 2012) according to real-life-inspired model ontology, which supports the validation and verification (V&V) process by mimicking the natural structure of platforms. However, the sum of the parts does not necessarily add up to the whole. In other words, V&V on individual agents is not enough to conclude that the whole simulation model is valid and verified. For a closed-loop military simulation, the validation process is even more difficult, since the process should also consider the impact of complex emergent behavior, including unpredicted outcomes of the scenario, the meaning of scenario variants (alternate futures), and the impact of stochastic parameters on the outcomes.

Feature-driven defined process:
For a closed-loop military simulation system, any subtle change in a single model may have a major impact on the emergent behavior. Therefore, it was obvious that the validation activities would have a repetitive nature. To understand the impact and to conduct the required follow-up actions, a *defined V&V process* is required. In order to define a process and ensure it is sustainable, the building blocks of validation activities should be defined. Inspired by feature-driven development practices (Palmer & Felsing, 2001) GEMED team focused on descriptive run-time features of each model as building blocks of the V&V process. As an example, “pop-up maneuver” may or may not be a required feature of a missile model, depending on its intended use. In GEMED, a simulation scenario-driven acceptance test procedure was defined for software verification activities, and a separate validation process was defined and executed for validation of the model space. Both processes were essentially based on the features of the software and models (Çilden et al., 2023).

In a broad sense, a feature-driven defined V&V process;

- facilitates sustainability via the repeatability of the process steps,
- makes it relatively easy to measure the progress of the V&V activities,
- defines a common ground for the whole activities for discussion and negotiation,
- enables automated V&V at the feature level, leaving more available resources for higher-level (emergent behavior) analysis.

On the other hand, such a process;

- requires the consensus of all stakeholders at the beginning of the project, which may not be always easy to achieve,
- makes it essential to use automation tools to keep track of progress, and be able to repeat some steps, which may require additional license fees, learning effort, and dedicated personnel.

Continuous regression management:
Possibly the most exhausting validation activity of a closed-loop simulation model is to cope with the notion so called the *butterfly effect*. Out of its mathematical context in chaos theory, the butterfly effect is popularly used to describe a phenomenon where a tiny change in start conditions can lead to vastly different outcomes. Developers of closed-loop simulations frequently observe this effect, even within a simple simulation scenario (Tang et al., 2021). In fact, this is directly related to the emergent behavior of MABS approach, since one of the aims of MABS is to observe the

emergent behavior, which is expected to give insight into the target complex phenomenon by observing those – possibly unexpected- outcomes. Every subtle change in models is a potential butterfly effect, and may cause regression on the overall simulation model. A continuous regression management strategy is required to keep an eye on this problem.

Continuous regression management;

- ensures that the earlier achievements of the validation activities are still intact,
- significantly elevates stakeholder confidence.

Additionally, continuous regression management;

- should be supported by automation tools (such as GUI testing tools, project specific scripts)
- would better be practiced by dedicated personnel,
- is a work package for which the amount of resources is difficult to predict and plan in advance.

USER AND ANALYSIS PERSPECTIVE

Data Management and Simulation Output Analysis Aspect

Closed-loop military simulation systems can be seen as generators of synthetic data. In a very broad sense, a trial is a function of input model and scenario parameters (variables) and returns a set of output data. The analysis contexts of closed-loop simulations are usually complex phenomena, otherwise, it would be possible to incorporate analytical methods to solve the target problem. Therefore, both input and output sets are expectedly extremely large, and difficult to construct, maintain, and process.

Simulation big data:
Multiple executions of scenario variants; which is a powerful potential of closed-loop military simulations, produce a huge amount of output data, i.e. simulation logs. When the number of trials reaches to hundreds/thousands, massive log data is required to be processed. During run-time, there are mainly commit-intensive data repository operations. Since a closed-loop simulation usually executes in best-effort mode, meaning there is no real-time constraint on the models, it is highly probable that write operations of simulation execution become a computational bottleneck for the overall performance of the batch executions, which requires proper handling. After the completion of the executions, on the other hand, the data repository operations are rather fetch-intensive, usually requiring fast reading of bulk data, so that required analysis can be rapidly processed.

GEMED team tried to produce an answer where a balanced system response could be obtained for both cases. A NoSQL database solution with a file system level event-log mechanism is devised, using MongoDB with Logstash. MongoDB collection mechanism is well suited to store different simulation log typed data, but big collections require extra indexing workload, which should be calibrated for various database use cases within the project (Çekinel et al., 2019).

Simulation big data problem;

- can be relaxed using recent advances in database management, such as NoSQL databases,
- may benefit from ontology-driven design best practices (Hofmann et al., 2011),
- may require the user to select a subset of the simulation outputs to reduce the data repository workload.

However,

- it is not straightforward to calibrate the data management solution for conflicting database operation requirements,
- database indexing for optimal performance may require significant effort, and possibly can not be finalized until the late stages of development.

Inspection of simulation outputs:
After the execution of the closed-loop scenario variants, a simulation big data resides in the data repository ready for processing. Although the contents of these inspections may vary, practices of simulation software architecture usually incorporate 3rd party software tools for this purpose. All the aforementioned data inspection forms have their own software solutions and related communities, with a crowded set of popular tools and libraries; such as R, PyCharm,

OpenRefine, PowerBI, SIMDIS, Unity to name a few. To use the power of 3rd party tools, it should be possible to convert inputs and outputs of the closed-loop simulation software into the appropriate data format, so that they can be processed by the tools of choice. For example, GEMED software can export simulation outputs into SIMDIS animation file format (*.asi) so that the SIMDIS player can animate the resulting scenario instance. GEMED also makes use of R scripts for descriptive statistics over the row output data.

For closed-loop simulations;

- it might be a good idea to make extensive use of 3rd party data inspection software,
- putting some planned effort into the selection of target tools and development of export libraries will probably pay back,
- no single 3rd party tool is a silver bullet for data inspection, so it might be a good idea to support at least a few popular/required software.

On the other hand,

- care should be taken regarding the license terms of the 3rd party software to be used,
- especially for advanced analysis, a meta-data derivation layer might be required before export.
- output data for graphical analysis is a near-must when it comes much-needed user understanding of complex phenomena. Data for graphs such as run-by-run comparable attribute value change rate, hypothesis analysis on select model attributes, time series, etc. is generated during a simulation run even though that data itself is not necessary from an analysis perspective.

User Experience and User Interface (UX/UI) Design Aspect

Every software tool has its own way of interacting with the user. Both user experience (UX) and user interface (UI) design are specialties that have been gaining increasing attention for a couple of decades, including simulation technologies (McArdle & Hilmer, 2022). In coherence with the increasing computation capacity, realism and scientific rigor within military simulation systems have been scaling up, demanding more effort from the developers to optimize UX/UI for the end-user to cope with the increasing complexity.

Closed-loop scenario definition challenge:

For closed-loop military simulations, designing the UI for scenario definition is a challenge. For multi-player simulations, a scenario mostly consists of the initialization set of simulation entities (i.e. the initial positions and orientations, starting values of running parameters) and a relatively small set of possible planned injections. After the start of the execution, the players shape what happens next through time. For closed-loop simulations, on the other hand, a form of scenario script is required to guide the simulation execution flow (Seymour, 2014), informing the simulation engine what to do under alternate circumstances, which may also be some unforeseen situations unless the scenario is executed at least once. This requirement gives rise to a script interpreter implementation task, together with a UX/UI design problem to make things easier for the user.

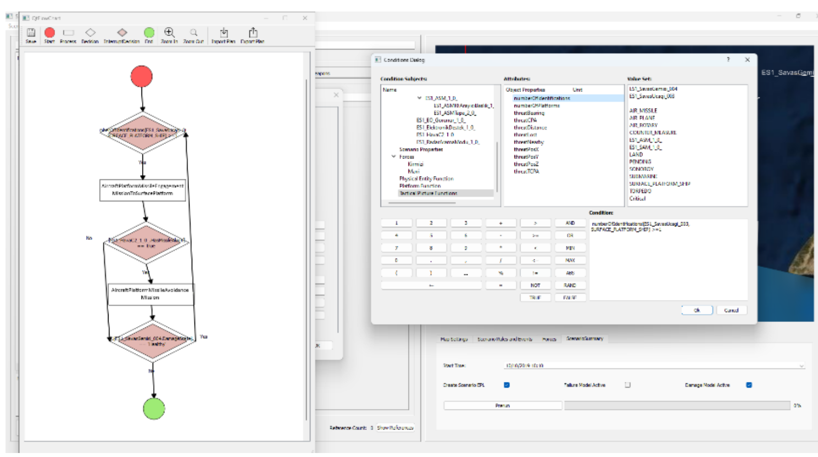

Figure 2 BPM based visual programming environment

GEMED implements a visual scenario scripting tool, inspired by business process modeling (BPM) practices (García-Holgado et al., 2015), together with a 3D mapping area for the visual creation of scenarios. The tool enables the user to define sequential and concurrent tasks to be executed by agents with conditional transitions or synchronizations in between, and also to compose temporal plans based on structures similar to swimlane activity diagrams. While this approach allows scenarios that are essentially reducible to executable scenario scripts, it is also human readable, and easier to alter (Figure 2). A pre-run capability is also available to quickly run the scenario and see that it executes in the desired fashion.

Closed-loop military scenario definition solution;

- requires supporting tools for the user to easily define a set of rules that would cover the whole automated scenario flow,
- would better allow pre-runs, meaning the user can speed-run the target scenario to properly build up a “happy path” scenario, suitable to use as a base for further altered executions (trials),
- can make use of the widely accepted model and process definition standards and languages, such as BPML, UML, SysML,

On the other hand;

- scenario definition tools for closed-loop simulation may require close negotiations with the end-user to understand their needs, where frequent interim releases of the tool might help,
- the proposed solution may be tightly coupled with the underlying model design; therefore, it can be important to set an appropriate model abstraction level to build it on.

Scenario variations management:
If there is the need for closed-loop simulation, then there is in fact the need for repeated execution of a target scenario upon some intended variations, possibly to generate synthetic data. Once a base scenario is available in a specific format (i.e. XML, JSON, custom script, etc.), there are alternative ways to define alterations of the scenario, depending on the intended analysis, such as random walks, Monte Carlo simulations, and DoE. It is of course possible to generate variations of the base scenario manually and execute them one-by-one using the operating system tools. However, from the user’s point of view, a systematic batch run via software UIs helps a lot. Additionally, the data management tools are expected to support this simulation variety, for a smooth post-execution analysis.

GEMED is equipped with a DoE software tool that guides the user to build an experiment on top of a base scenario. Step-by-step, the user defines the levels, factors, and stochastic parameters of a statistical experiment design, then the software takes care of the rest. It generates all variants of the base scenario according to the defined experiment, automatically dispatches to computation units (CPU threads), and displays the progress during execution. An experiment in GEMED is defined as an aggregation encapsulating a base scenario, definitions of the variation around the base scenario, and execution outputs of the experiment.

Scenario variations management;

- should include tools to easily alter the scenario (such as DoE, random variables, replaced entities, etc.), and to manage the scenario variations,
- is closely dependent on the ontology, meaning that careful ontology management can enable smooth batch management at the data layer,
- can be coupled with scenario execution management, which may allow a fluent control flow of simulation software, and can be practical for the user.

However,

- the user should be informed and/or trained about the meaning of the batch run methods presented by the tool, since they would have a statistical impact on the expected output, which may require additional interpretation,
- the batch executions can block much of the computing resources for a long time, therefore, it is a good practice to design the simulation software non-blocking for other processes of the host computer.

ENGINEERING AND MANAGEMENT PERSPECTIVE

Software Engineering Aspect

A closed-loop military simulation is a complex and large software and needs rigorous application of appropriate software engineering practices to achieve a successful completion. Besides the body of knowledge on developing simulation models and tools (Andreas Tolk, 2012; Nutaro, 2011), there are few issues that can arise when engineering for closed-loop military simulation software.

The simulation engine:
Orchestration of all models in the simulation loop is at the core of all simulation software. The initial drive of the orchestration method is the modeling approach, for which there are well-founded mainstream methods, such as discrete event, incremental time progression, system dynamics, and multi-agent based simulation (Law, 2013). This taxonomy does not impose an orthogonal solution space, but rather offers a set of tools to fulfill the simulation requirements. Development of a closed-loop simulation may require a hybrid solution depending on the existence of restrictive requirements, such as computational performance envelopes, the existence of multiple-fidelities, the distribution of models, and so on.

GEMED, being a closed-loop military simulation software, benefits from variable time, discrete event, and multi-agent based simulation approaches. This hybrid modeling approach stemmed from model integration and performance requirements of the project and constrained the software architecture to fuse these methods in harmony. Although there were COTS tools and libraries that were eligible for this purpose, the team chose to develop its own in-house simulation engine that harmonizes the approaches to satisfy the project's target criteria. At this point, if performance requirements are the most important bounding frames on the design of the simulation engine, modeling fidelity requirements are probably the second critical ones to consider.

For a closed-loop simulation, selection or design of the simulation engine;
- usually tightly depends on performance and model fidelity requirements,
- is on the critical technical path of the software development plan,
- it is possible to fuse multiple modeling approaches to achieve targeted software quality attributes, depending on the intended use,

However, it should be noted that;
- the design of the simulation engine has the impact to shaping the design and development practices for models and, therefore may require careful planning and incremental prototyping,
- there is a trade-off between decentralization of the simulation engine (or distribution of simulation models, including interoperability) with repeatability and overall execution performance,
- the selection or design of the engine should take into account the autonomous execution principles of the intended simulation models, since they determine the techniques (events, time advance functions, state transitions, etc.) required to serve the models throughout their execution.

Utilizing computational resources:
A batch run is greedy on computational resources, and the obvious solution is to implement the utilization of multiple processing resources for faster execution. A closed-loop simulation software makes use of batch runs, so some viable parallel execution solutions should be addressed. Depending on the available hardware resources, parallel processing, distributed computing and grid computing techniques are all possible to achieve this utilization (Fujimoto, 2015).

GEMED is a desktop application that utilizes the available CPUs of the host computer. A trial is chosen as a sequentially indivisible (atomic) computation chunk, to assure repeatability and controlled stochasticity. By default, it uses all available CPU threads provided by the operating system.

For a closed-loop military simulation, computing resource utilization;
- would be necessary most of the time, which differentiates it from most other simulation systems,
- can be achieved simply by distribution or parallelization of each trial,
- is essentially a trade-off problem where several factors should be considered, including computation complexities of trials, inter-process communication or network connection latencies, network topology, and hardware specifications.

On the other hand,
- distributing simulation models/entities over computing resources would not be straightforward to implement, while trying to maintain common target attributes of a closed-loop simulation such as repeatability, faster than real-time execution, etc.; instead, distribution of trials may be a more practical design choice,
- addressing pseudo-randomness may require special handling in case of parallel/distributed executions,
- sophisticated utilization problems may require specific expertise.

Project Management Aspect

A closed-loop military simulation is a comprehensive software project, with all the hassles of every step of the software development lifecycle, with some additional concerns due to all the other aspects previously mentioned. Therefore, there are a couple of points to mention, that we believe are specific issues to be handled throughout a closed-loop military simulation development project, summarized as follows.

Significant difficulty in effort estimations:
One of the most obstructive challenges that can be faced while managing a closed-loop simulation development project is perhaps the significant difficulty in effort estimations The authors believe that this issue primarily stems from various difficulties in the development of the simulation models, including additional layers of model autonomy, frequent necessity for emergent behavior calibration, unpredictable effort required by model fidelity alignment tasks, and relatively high coupling of models at the conceptual level.

GEMED team experienced and realized these problems within the first quarter of the project time span. Tailored applications of existing interpretations of agile practices, namely Kanban and Scrum, helped the team to devise project-specific mechanisms (Alqudah & Razali, 2017). At the early stages of the project, the team proceeded with the Kanban practices because each newly developed model exhibited unpredictable emergent behaviors interacting with other models, requiring short but frequent interventions in multiple models to alter or correct them. In this phase, the team focused on limiting the number of work items in progress, instead of limiting the effort reserved for each task. The activities mainly aimed at overcoming technical uncertainties, advancing research-based efforts as much as possible, and developing the first working prototype of the software, to mitigate technical ambiguities. After a certain confidence in the coherent execution of critical models is achieved, the team preferred to switch to a tailored Scrum practice, a more disciplined and systematic way of iterative software development, with more accurate plans based on historical data-oriented estimates.

To properly manage effort estimations of closed-loop simulation projects:

- it is useful to practice iterative and incremental development (such as Agile methods), which makes it easier to collect the effort history of the team and update future estimations accordingly,
- early integration and frequent delivery practices can facilitate a quicker reduction in errors within effort estimations, since early integration helps in the earlier identification of unpredictable emergent behavior issues, while frequent delivery aids in the rapid clarification of user expectations,
- it would be useful to initiate model validation activities starting from the requirements analysis phase, at the conceptual level.

Additionally;

- there is no generic solution for accurate effort estimation for the development of complex simulation software, so additional effort may be required to find tailored solutions specific to the project,

Adverse effects of multiple intended uses:
For simulation software, all requirements are shaped based on a commonly agreed intended use (Balci, 2011), especially within the modeling context. If, however, there are multiple intended uses, it is a technical challenge to design a software and/or conceptual model space, in such a way that it covers every intended use of the resulting software. In the case of conflicting intended uses, the solution may even result in implementation of different modes of simulation software, especially for closed-loop simulations. Missile-target engagements (with a reference simulation-time resolution of hundreds of seconds), compared to a force-on-force analysis of combatant platforms (with a reference simulation-time resolution of days or weeks) might be an example of conflicting intended uses.

To cope with multiple indented uses;

- requirements elicitation phase should be carefully handled, frequently reviewing the intended use(s) until the early prototypes,
- a higher-level authority may be responsible for aligning the stakeholders around a single intended use.

On the other hand;

- frequent delivery or early prototyping may fail to demonstrate the conflicting intended uses since they focus more on lower-level software and model features compared to a conceptual intended use, which may delay the identification of the problem.

CONCLUSIONS AND DISCUSSION

This paper summarizes potential pitfalls that can be faced throughout the development of a closed-loop military simulation software and suggests ways to avoid or overcome these problems. We try to avoid cross-context overlaps, such as problems that can be encountered in any simulation development process, and focus on the ones that we believe are specific to closed-loop simulation. The paper defines the issues, summarizes our experience –including the causes and solutions-, and recommends further items that we foresee that can either help or harm on that issue.

Closed-loop military simulation software presents unique challenges due to factors such as limited user intervention, model emergent behavior, the butterfly effect, and the inevitable autonomy of models. Our experience with the GEMED project highlights three critical areas that require attention. First, simulation models pose challenges in terms of design, development, and credibility. From the user's perspective, there may be a need for unique solutions in areas such as big data management, output analysis, and user interface design. Lastly, from a software engineering and project management perspective, there is the challenge of carefully balancing a number of trade-off decisions.

This paper is based on experiences, thus, a possible future work would be to systematically analyze the issues identified here and explore the impacts of the proposed solutions through collected data. For instance, a unified analysis of project management metrics, software development process metrics, and measurements collected during the model V&V process may yield informative results. As another example, the contribution of GUI test automation tools within the scope of continuous model regression management could be analyzed through comparative measurements.

Another future work is that we currently explore methods to fully automate the closed-loop simulation experience, i.e. to achieve fully-automated wargame simulation capability. This work primarily aims to develop ontology-driven end-to-end closed-loop simulation software and is in the early research phase.

A possible research direction is the development of common design frameworks that all stakeholders can work on together, particularly addressing the two most challenging concepts of closed-loop military simulations: intended use and model fidelity. These frameworks will likely involve disciplines such as requirements management and simulation ontology.

ACKNOWLEDGEMENTS

This study was supported by the Scientific and Technological Research Council of Türkiye (TUBITAK) under Grant No. 115A044. The authors thank TUBITAK for their support.

REFERENCES

Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. P. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, *24*, 13–33. https://doi.org/10.1016/j.cosrev.2017.03.001

Alqudah, M., & Razali, R. (2017). A comparison of scrum and Kanban for identifying their selection factors. *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, 1–6. https://doi.org/10.1109/ICEEI.2017.8312434

Andreas Tolk (Ed.). (2012). *Engineering principles of combat modeling and distributed simulation*. Wiley Online Library.

Balci, O. (2011). How to successfully conduct large-scale modeling and simulation projects. *Simulation Conference (WSC), Proceedings of the 2011 Winter*, 176–182.

Biggs, A. T., & Hirsch, D. A. (2022). Using Monte Carlo simulations to translate military and law enforcement training results to operational metrics. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *19*(3), 403–415. https://doi.org/10.1177/15485129211021159

Çekinel, R. F., Poyraz, D., Karagöz, P., Çilden, E., Oğuztüzün, H., & Gültekin, E. (2019). Simülasyon kayıtlarının verimli saklanması ve işlenmesi (Efficient storage and processing of simulation logs). *9. Ulusal Savunma Uygulamaları Modelleme ve Simülasyon Konferansı Bildiri Kitabı*, 140–151. http://usmos.metu.edu.tr/system/files/usmos2019_kitap.pdf#page=150

Çetintaş, İ. Ç., Çilden, E., & Yılmaz, B. (2020). *Multiple environments in multi-agent based simulation: A case study*. The 1st International Workshop on Frontiers of Multi-Agent Systems, In conjunction with PRIMA 2020.

Çilden, E., Sezer, A., Canberi, H., & Oğuztüzün, H. (2023). Iterative and incremental validation of simulation conceptual models. *The Interservice Industry, Training, Simulation, and Education Conference (I/ITSEC) Proceedings*.

Fujimoto, R. (2015). Parallel and Distributed Simulation. *Proceedings of the 2015 Winter Simulation Conference*, 45–59. http://dl.acm.org/citation.cfm?id=2888619.2888624

García-Holgado, A., García-Peñalvo, F. J., Hernández-García, Á., & Llorens-Largo, F. (2015). Analysis and Improvement of Knowledge Management Processes in Organizations Using the Business Process Model Notation. In *New Information and Communication Technologies for Knowledge Management in Organizations* (Vol. 222, pp. 93–101). Springer International Publishing. https://doi.org/10.1007/978-3-319-22204-2_9

Gianni, D., D'Ambrogio, A., & Iazeolla, G. (2011). A software architecture to ease the development of distributed simulation systems. *SIMULATION*, *87*(9), 819–836. https://doi.org/10.1177/0037549711400777

Hodický, J., Procházka, D., Baxa, F., Melichar, J., Krejčík, M., Křížek, P., Stodola, P., & Drozd, J. (2020). Computer Assisted Wargame for Military Capability-Based Planning. *Entropy*, *22*(8), 861. https://doi.org/10.3390/e22080861

Hodicky, J., Prochazka, D., & Prochazka, J. (2019). Automation in Experimentation with Constructive Simulation. In J. Mazal (Ed.), *Modelling and Simulation for Autonomous Systems* (Vol. 11472, pp. 566–576). Springer International Publishing. https://doi.org/10.1007/978-3-030-14984-0_42

Hofmann, M., Palii, J., & Mihelcic, G. (2011). Epistemic and normative aspects of ontologies in modelling and simulation. *Journal of Simulation*, *5*(3), Article 3. https://doi.org/10.1057/jos.2011.13

Law, A. M. (2013). *Simulation Modeling and Analysis* (5th ed.). McGraw-Hill.

Law, A. M. (2017). A tutorial on design of experiments for simulation modeling. *2017 Winter Simulation Conference (WSC)*, 550–564. https://doi.org/10.1109/WSC.2017.8247814

McArdle, J., & Hilmer, E. (2022). Effectively Integrating Technology into Wargames. *The Interservice Industry, Training, Simulation, and Education Conference (I/ITSEC) Proceedings*.

Ministry of Defence UK. (2017). *Wargaming Handbook*. Development, Concepts and Doctrine Centre. https://assets.publishing.service.gov.uk/media/5a82e90d40f0b6230269d575/doctrine_uk_wargaming_handbook.pdf

NATO. (2019). *NATO Glossary of Terms and Definitions* (AAP-06). https://www.coemed.org/files/stanags/05_AAP/AAP-06_2019_EF.pdf

NATO. (2021). *NATO Concept Development and Experimentation Handbook: A Concept Developer's Toolbox* (2.10). NATO Allied Command Transformation. https://www.act.nato.int/wp-content/uploads/2023/05/NATO-ACT-CDE-Handbook_A_Concept_Developers_Toolbox-1.pdf

NATO. (2023). *NATO Wargaming Handbook*. NATO Allied Command Transformation. https://paxsims.wordpress.com/wp-content/uploads/2023/09/nato-wargaming-handbook-202309.pdf#page=63.05

Nohel, J., Hradský, L., Flasar, Z., Zahradníček, P., & Kristalova, D. (2022). Utilizing the Maneuver Control System CZ in the Course of Wargaming Modelling and Simulation. In *Modelling and Simulation for Autonomous Systems* (pp. 357–373). Springer International Publishing.

Nutaro, J. J. (2011). *Building Software for Simulation: Theory and Algorithms, with Applications in C++*. John Wiley & Sons, Inc.

Özkan, G., Çilden, E., Gültekin, E., Sezer, A., & Canberi, H. (2019). A reference architecture for military simulations with experimentation and analysis features. *Proceedings of the 2019 Summer Simulation Conference*.

Palmer, S. R., & Felsing, M. (2001). *A Practical Guide to Feature-Driven Development* (1st ed.). Pearson Education.

Pěchouček, M., Jakob, M., & Novák, P. (2012). Towards Simulation-Aided Design of Multi-Agent Systems. In *Programming Multi-Agent Systems* (Vol. 6599, pp. 3–21). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28939-2_1

Seymour, C. N. (2014). *Capturing the full potential of the Synthetic Theater Operations Research Model (STORM)* [M.S. Thesis]. Naval Postgraduate School.

Smith, R. D. (1998). Essential techniques for military modeling and simulation. *Proceedings of the 30th Conference on Winter Simulation*, 805–812. http://dl.acm.org/citation.cfm?id=293309

Tang, Y., Li, L., & Liu, X. (2021). State-of-the-Art Development of Complex Systems and Their Simulation Methods. *Complex System Modeling and Simulation*, *1*(4), 271–290. https://doi.org/10.23919/CSMS.2021.0025

Turnitsa, C., Blais, C., & Tolk, A. (2022). *Simulation and wargaming*. Wiley Online Library.