# ASA: A Simulation Environment for Evaluating Military Operational Scenarios

Joao P. A. Dantas, Andre N. Costa, Vitor C. F. Gomes, Andre R. Kuroswiski, Felipe L. L. Medeiros, and Diego Geraldo

Decision Support Systems Subdivision, Institute for Advanced Studies, Trevo Cel. Av. Jose A. A. do Amarante, 1, Putim, Sao Jose dos Campos 12228-001, Sao Paulo, Brazil. {dantasjpad, negraoanc, vitorvcfg, kuroswiskiark, felipefllm, diegodg}@fab.mil.br

**Abstract.** The Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA* in Portuguese) is a custom-made object-oriented simulation framework developed mainly in C++ that enables the modeling and simulation of military operational scenarios to support the development of tactics and procedures in the aerospace context for the Brazilian Air Force. This work describes the ASA framework, bringing its distributed architecture for managing multiple simulation machines, a data analysis platform for post-processing simulation data, the capability of loading models at simulation runtime, and a batch mode execution platform to perform multiple independent executions simultaneously. In addition, we present a list of recent works using the ASA framework as a simulation tool in the air combat context.

**Keywords:** simulation environment, distributed simulation, data analysis, military, operational scenarios

## 1 Introduction

The Institute for Advanced Studies (IEAv), a research organization of the Brazilian Air Force (*Força Aérea Brasileira – FAB* in Portuguese), has developed, since 2018, the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA* in Portuguese) to provide a computational solution that enables the modeling and simulation of operational scenarios, allowing users to establish strategies, parameters, and command decisions to support the development of tactics, techniques, and procedures in the aerospace context for defense purposes.

The characteristics of the modern battlefield scenarios bring new challenges to building practical combat simulations, requiring more integrated and flexible solutions to address not only technical but also organizational aspects [10]. The Advanced Framework for Simulation, Integration and Modeling (AFSIM) is an example of a framework that is being developed to address some of these challenges [1]; however, it is restricted to a few US Partners. In this context, the ASA environment was conceived to be, at the same time, adequate to support FAB's

strategic planning, meet operational analysis needs, and allow the development and evaluation of new technologies to enhance military research, posing itself as a flexible solution that may be tailored to the user needs. This flexibility was aimed at the clients' diverse characteristics, which led to a wide range of requisites that could not be met only by commercial off-the-shelf (COTS) simulation software. Since developing a completely new solution would not be efficient, the ASA team decided to look at openly available tools aiming to integrate them into a flexible, accessible, and scalable environment.

The proposed solution uses as its simulation engine the Mixed Reality Simulation Platform (MIXR) [11], an open-source software project designed to support the development of robust, scalable, virtual, constructive, stand-alone, and distributed simulation applications. ASA extended MIXR's possibilities, adding extra elements that created an environment to optimize developers' and analysts' tasks. We created a manager application to be the interface between multiple resources, working as a hub to run, store, and analyze diverse simulations on numerous computers. In addition, this application allows for creating a significant number of simulations simultaneously, only changing initial conditions according to the analyst's needs. Also, models and tools can be loaded dynamically at runtime to increase flexibility. All simulation data is stored in a dedicated database, which speeds up the data collection process and promotes more robust statistical analyses. Additionally, considering the complexities of the outcomes and the variable technical knowledge of the ASA users, we integrated a dedicated data analysis platform into the system, not just for planning or visualization purposes, but also for post-processing the data produced from the scenarios.

Thus, the main contribution of this work is to introduce a new environment for modeling and simulation in the aerospace context for military purposes, containing: a distributed architecture for managing multiple simulation machines; an enhanced military operational scenario data analysis platform for post-processing simulation data; a capability of loading models at simulation runtime; a batch mode execution to perform multiple executions using different initial parameters. Besides, we present a list of recent works using the ASA platform as a simulation tool for problem-solving in the air combat domain.

The rest of this paper is organized as follows. Section 2 presents the ASA architecture. In Section 3, we bring some research, using ASA as a simulation tool, related to air combat analysis as application examples of this simulation framework. Finally, Section 4 states the conclusions about ASA's current state and brings some ideas for future works.

## 2    ASA Architecture

The ASA design consists of three different modules. The first part is the simulation framework, defined as AsaSimulation, which provides the applications and necessary services to create and execute simulations. The second part is the user interface applications, denominated AsaUserInterfaces, which provides tools for creating scenarios by listing all available components to be included,

simulation revisualization, and batch executions. Lastly, the third part is the analysis module, called AsaDataScience, which allows for post-processing and analysis of scenario executions.

Figure 1 displays a summary of the ASA architecture, and the following subsections provide details on all these three primary ASA modules. All ASA applications use network communications, allowing processing to be distributed across multiple servers on a network.
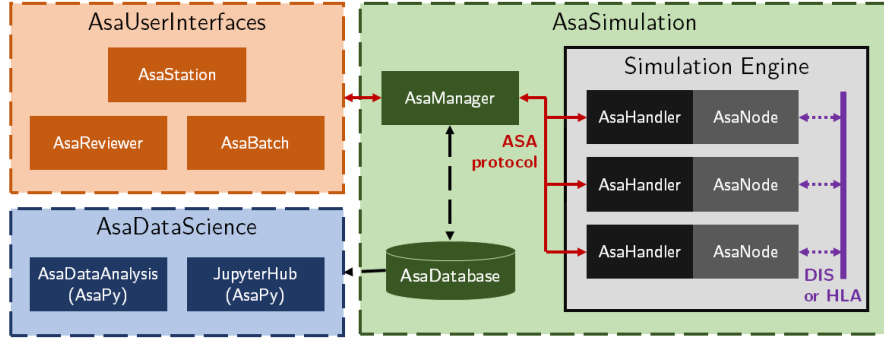


**Fig. 1.** ASA: modules and software applications.

### 2.1    AsaSimulation

The AsaSimulation module provides the necessary components for developing and executing a scenario simulation. It consists of applications, services, and libraries that help create agent models by developers and in the elaboration and simulation of scenarios by analysts.

The main features of this module are management and execution of simulations, dynamic loading of extension models, management of simulation batch processing, and distribution of simulation executions.

A permanent storage service called AsaDatabase keeps agent metadata, simulation scenarios, execution data, and analysis results. This service meets the storage demands of the AsaSimulation and AsaDataScience modules.

New agent models can be added to ASA by extending the interfaces and classes available in a library called AsaExtension. A new agent model to be loaded in the AsaSimulation framework must provide its compiled code (.dll or .so) and a JSON file that describes the parameters and components accepted by the model. AsaExtension also provides functionalities to allow extensions to store agent data in the AsaDatabase. This is done through the C++ macro `RECORD_ASA_CUSTOM_DATA("tag", agent)`, which receives as its first parameter a tag that identifies the type of the agent and the agent object with the attributes

to be stored in the AsaDatabase. Storing the state of the agents at each simulation step is optional, but it is essential if there is an interest in performing post-processing of the simulation data.

The AsaSimulation framework allows analysts to select agent parameters (placeholders) to be defined during the simulation execution request to facilitate the execution of batch simulations. Analysts can request a batch execution from these simulation scenario templates by informing a list of initial conditions for the previously selected agent parameters. Each set of initial conditions combined with the scenario template produces an execution request allocated to run on a distributed processing node.

The distribution of simulation processing is done by dividing the responsibilities of simulating to three applications on a network: AsaManager, AsaHandler, and AsaNode. The following subsections detail each of these applications.

### 2.1.1 AsaManager

AsaManager is responsible for coordinating and eventually synchronizing the processes to use ASA in a distributed manner. In addition to controlling the simulation processing nodes, AsaManager also receives control commands (play, pause, resume, and stop) via an iterative terminal or the user interface and transmits them to the simulation engine, processing the simulation itself. One of the essential functions of AsaManager is to identify which computers on the registered network are able to process simulations and, automatically and transparently to the user, allocate the requested simulation to be executed on some accredited machine. Furthermore, the application incorporates the CRUD (Create, Read, Update, and Delete) methods responsible for accessing the database and updating it. As a result, AsaManager has become a critical application of the ASA platform with the ability to interact with applications assigned for providing the user interface, data repository, simulation processing, and machine registration services. All these communications are made using a custom protocol called ASA protocol. Besides, it makes many complex tasks imperceptible for the user, such as loading extensions metadata, converting the JSON scenario specification to a compatible format (MIXR initialization scheme), interpreting commands to control the execution progress on the simulation machine, and changing orders in agents' behaviors during real-time simulation.

### 2.1.2 AsaNode

AsaNode is the simulation engine of the ASA platform, and its primary function is to process the simulation itself. It is an executable file obtained from the compilation of codes developed in MIXR and features developed by the ASA team, such as dynamically loading extensions and controlling the execution of the simulation (pause, resume, stop, execution speed, etc.). It estimates how the scenario will evolve, considering the models incorporated in each agent present in the simulation. AsaNode can run on the same machine as the AsaManager application or in a clustered computing environment. This capability is essential

when the user wants to simulate a set of scenarios, called batch, and the main difference between them is the initial configuration of each agent.

### 2.1.3 AsaHandler

AsaHandler application is responsible for informing AsaManager which computers on a network are available to execute simulations. AsaHandler reports the machine where the simulation will be processed, with active status, and waits for the AsaManager request for the AsaNode initialization. In general, either with the distributed approach or with standalone processing, AsaHandler receives the properly formatted specifications from AsaManager; runs an instance of AsaNode dedicated to processing them; and transmits execution, pause, and end of the simulation commands based on the orders received from AsaManager, which obtains them from the user's commands in the AsaManager iterative terminal or the user interface responsible for creating scenarios.

## 2.2 AsaUserInterfaces

AsaUserInterface module has three tools – AsaStation, AsaBatch, and AsaReviewer – which, respectively, provide a visual interface to facilitate the creation of scenarios, the management and execution of batch simulations, and the revisualization of an executed simulation.

### 2.2.1 AsaStation

AsaStation is a Graphical User Interface (GUI) in which the analyst can create a scenario using military symbols, geometric drawings, aeronautical charts, and digital terrain models and visualize how this scenario will evolve during the simulation. For this purpose, we used the AEROGRAF Platform [13], a geographic information system developed at IEAv intended for planning tactical scenarios of interest to FAB, through one application (plugin) responsible for allowing the visualization of the ASA components available to run the simulations. Some of the features provided by AsaStation are: connecting to AsaManager; creating, deleting, or modifying a simulation; selecting a specific simulation to run; adding, removing, or modifying simulation components/agents; and changing the simulation execution view. Figure 2 shows an example of the visualization produced for air combat scenarios in AsaStation.

### 2.2.2 AsaBatch

AsaBatch is a solution that allows the preparation and monitoring of the execution of multiple simulations in parallel in a simplified way. This resource focused on three demands: analyzing scenarios with stochastic factors, model optimization, and machine learning techniques. The solution leverages MIXR's proposal of scenario initialization through scripts, allowing analysts to create as many simulations as necessary from a reference scenario. This feature allows

the reproduction of simulations with different random numbers, which creates numerous analysis possibilities, allowing the variation of all initialization parameters of the models, for example, starting positions of each component, weapon performance factors, and characteristics of the behaviors implemented for the agents.

The capabilities of AsaBatch already supported several academic studies [4, 2, 12, 5, 6, 3, 8] that used the execution of thousands of simulations to achieve their results. It is worth noting that the high individual performance of each simulation is mainly due to the capabilities of MIXR itself. However, the developed feature allowed the user to distribute the processing across clusters of computers transparently, making it more straightforward for analysts to scale experiments. The GUI enables the analysts to follow the primary outcomes of the simulation executions in real-time, which can also be evaluated later, in more detail, according to information of interest previously selected for recording in the database.

### 2.2.3    AsaReviewer

AsaReviewer is a software component that allows the user to revisualize simulations that have already been executed and whose data has been stored in the AsaDatabase. Its primary purpose is to support the decision-maker when it is desirable to observe how the simulated scenario has evolved, which may lead to changes regarding air force assets from initial scenario decisions. Based on a previously executed simulation, the decision-maker can use AsaReviewer to assess whether it will be necessary to make minor adjustments in military resources to better achieve the mission's objective in the simulated operational environment.

Like AsaStation, AsaReviewer is also a plugin developed in the AEROGRAF Platform, in which, through its user interface, it is possible to select a simulation, then choose an execution of this specific simulation (since the same simulation can have several executions with the same parameters or different ones), and finally visualize how the agents in the scenario interacted. Its final purpose is to help the decision-maker refine the simulated scenario when we desire to have control of the dynamics of agent interaction, which is often the most critical aspect in the analysis and subsequent decision-making processes.

### 2.3    AsaDataScience

The AsaDataScience module provides tools for understanding the factors that led to the simulation results. It contains a web dashboard, which allows the scenario analyst to visualize indices and metrics that help to measure performance, cost, and restrictions of the military means used in the simulated scenario, in addition to a library called AsaPy – since it is written in the Python programming language – with methods that provide the following features: manipulation of data generated by the simulations, design of experiments, exploratory data analysis, supervised machine learning, and metrics. Table 1 presents the descriptions of its four packages. This module also has a cloud-computational environment
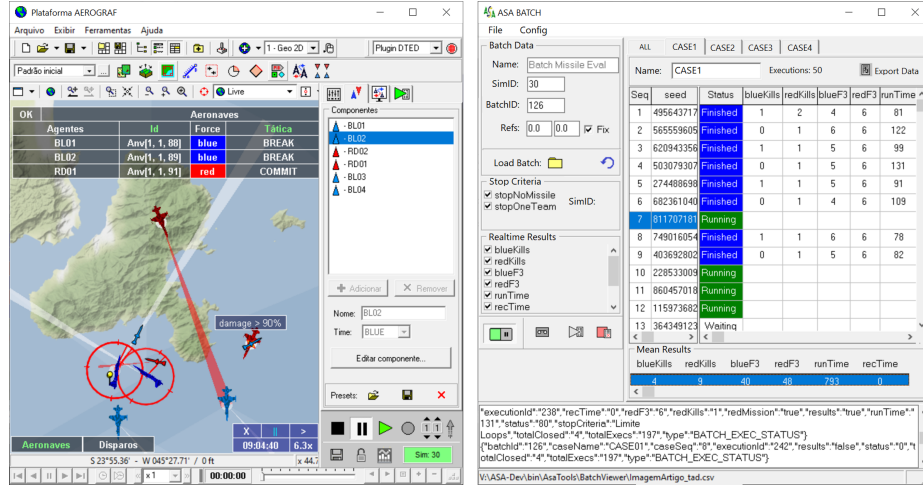
**Fig. 2.** AsaStation: interface to create and visualize simulations (left). AsaBatch: interface to manage multiple simulations (right).

that includes high-performance hardware and a JupyterHub [1] interactive programming environment to optimize the batch execution of simulations and their subsequent data analysis.

**Table 1.** AsaPy packages: description and methods

| Package | Description |
|---|---|
| DoE | Design of experiments methods for planning the input parameters. |
| API | Access to the functionalities provided by the AsaSimulation elements, especially those related to the execution and management of simulations. |
| Data | Access data present in the AsaDatabase and contains the methods necessary for handling the data and metadata simulation. |
| Analysis | Machine learning and statistical methods for analyzing output data. |

Running a simulation invariably results in massive data being collected at every step. Once this data is adequately stored and structured in the AsaDatabase, one may employ techniques capable of inferring high-level information from data visualization, an area of data science that relates to the graphical representation of abstract information to bring a general idea of the data analyzed.

The AsaDataAnalysis platform aims to present the analysis of data and metrics developed within the scope of military operational scenarios and consists of a web-service (back-end) to provide the analysis results and a user interface (front-end) for interaction and graphical display of analyses. The web-service has

---
[1] https://jupyter.org/hub

functions for processing the data displayed on the front-end. After processing the data, the results are delivered to the front-end via a REST API and stored in the database to avoid reprocessing large volumes of data and consequently speeding up future requests. The AsaDataAnalysis can address even data provided from different simulation frameworks, as is done in [7].

## 3   Applications

ASA has helped develop extensive academic work in the aerospace and defense context relating to different fields of study, such as artificial intelligence, machine learning, data science, and optimization. These applications can be seen in [4, 2, 12, 5, 6, 3, 8, 9], being detailed as follows.

In [4, 8], the authors designed a multilayer perceptron neural network using data from constructive simulations to be employed in an embedded device to enhance the pilot's situational awareness in the in-flight decision-making process.

In [2, 3], the authors proposed the use of both artificial potential fields and simulation optimization to achieve more robust results for simulated military aircraft to fly in formation, using a large set of scenarios for the optimization process, which evaluates its objective function through the simulations.

In [12], the author analyzed the feasibility of using Agent-based Modeling and Simulation (ABMS) to assess air defense capabilities in the context of strategic planning, presenting a case study based on simulations of BVR air combat scenarios at the engagement level.

In [5], the authors designed an engagement decision support tool for BVR air combat, using a supervised learning model based on decision trees, to measure the quality of a new engagement, which is the moment the pilot engages a target by executing offensive maneuevers.

In [6], a Deep Neural Network estimates the Weapon Engagement Zone (WEZ) maximum launch range, allowing pilots to identify the airspace in which the available missile has a more significant probability of successfully engaging a particular target.

In [9], the authors integrated the ASA infrastructure into an external optimization framework to apply six different metaheuristics to optimize Unmanned Aircraft Vehicles' tactical formations, considering enemy variables such as firing distance and initial position in a BVR combat scenario.

## 4   Conclusions

In this paper, we have provided a high-level overview of the ASA simulation framework, which IEAv has developed since 2018 with the primary goal of evaluating military operational scenarios of interest to FAB. The platform has the distinct features of managing multiple simulation machines on individual or various computers, dynamically loading `.dll` and `.so` files, running simulation batches, and providing the AsaDataScience module, a data analysis platform for military operational scenarios. Additionally, we presented some work using ASA as

a simulation tool to support air combat applications. For future work, we plan to release part of the source code with the general architecture to encourage the development of different applications in the same simulation platform. Furthermore, ASA will operate as a simulation-as-a-service (SimaaS) tool to attend to diverse simulation demands in the defense and aerospace context, which may lead to more interoperability between government, academia, and industry.

## Acknowledgments

## References

[1] Clive, P.D., Johnson, J.A., Moss, M.J., Zeh, J.M., Birkmire, B.M., Hodson, D.D.: Advanced framework for simulation, integration and modeling (afsim)(case number: 88abw-2015-2258). In: Proceedings of the International Conference on Scientific Computing (CSC), p. 73. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2015)

[2] Costa, A.N.: Sequential Optimization of Formation Flight Control Method Based on Artificial Potential Fields. Master's Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil (2019)

[3] Costa, A.N., Medeiros, F.L., Dantas, J.P., Geraldo, D., Soma, N.Y.: Formation control method based on artificial potential fields for aircraft flight simulation. SIMULATION **98**(7), 575–595 (2022). DOI 10.1177/00375497211063380. URL https://doi.org/10.1177/00375497211063380

[4] Dantas, J.P.A.: Apoio à decisão para o combate aéreo além do alcance visual: uma abordagem por redes neurais artificiais. Master's Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil (2018)

[5] Dantas, J.P.A., Costa, A.N., Geraldo, D., Maximo, M., Yoneyama, T.: Engagement Decision Support for Beyond Visual Range Air Combat. In: 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE), pp. 96–101. IEEE (2021). DOI 10.1109/LARS/SBR/WRE54079.2021.9605380

[6] Dantas, J.P.A., Costa, A.N., Geraldo, D., Maximo, M.R.O.A., Yoneyama, T.: Weapon Engagement Zone Maximum Launch Range Estimation Using a Deep Neural Network. In: A. Britto, K. Valdivia Delgado (eds.) Intelligent Systems, pp. 193–207. Springer International Publishing, Cham (2021)

[7] Dantas, J.P.A., Costa, A.N., Medeiros, F.L.L., Geraldo, D., Maximo, M.R.O.A., Yoneyama, T.: Supervised Machine Learning for Effective Missile Launch Based on Beyond Visual Range Air Combat Simulations. In: Proceedings of the Winter Simulation Conference, WSC '22 (2022)

[8] Dantas, J.P.A., Maximo, M.R.O.A., Costa, A.N., Geraldo, D., Yoneyama, T.: Machine Learning to Improve Situational Awareness in Beyond Visual Range Air Combat. IEEE Latin America Transactions **100** (2022). URL https://latamt.ieeer9.org/index.php/transactions/article/view/6530

[9] De Lima Filho, G.M., Kuroswiski, A.R., Medeiros, F.L.L., Voskuijl, M., Monsuur, H., Passaro, A.: Optimization of Unmanned Air Vehicle Tactical Formation in War Games. IEEE Access **10**, 21,727–21,741 (2022). DOI 10.1109/ACCESS.2022.3152768

[10] Hill, R.R., Tolk, A., Hodson, D.D., Millar, J.R.: Open challenges in building combat simulation systems to support test, analysis and training. In: 2018 Winter Simulation Conference (WSC), pp. 3730–3741 (2018). DOI 10.1109/ WSC.2018.8632233

[11] Hodson, D.D., Gehl, D.P.: The Mixed Reality Simulation Platform (MIXR). In: Worldcomp 2018, International Conference on Scientific Computing (CSC'18) (2018)

[12] Kuroswiski, A.R.: Modelagem e Simulação Baseada em Agentes como Ferramenta de Apoio à Avaliação de Capacidades de Defesa Aérea. Master's Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil (2020)

[13] Petersen, J.F., Aquino, M.R.C., Salles, R.N.: Plataforma AEROGRAF: um SIG voltado para a Força Aérea. Revista Spectrum, n. 11, Comando-Geral de Operações Aéreas (2008)