applied sciences

MDPI

Review

Review of the Research Progress in Combat Simulation Software

Fengshun Lu [1], Xingzhi Hu [1], Bendong Zhao [1,*,†], Xiong Jiang [1,*,†], Duoneng Liu [2], Jianqi Lai [1] and Zhiren Wang [1]

1 China Aerodynamics Research and Development Center, Mianyang 621000, China; lufengshun@cardc.cn (F.L.); huxingzhi99@163.com (X.H.); laijianqi2020@cardc.cn (J.L.); wangwangzhiren@163.com (Z.W.)
2 College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China; liuduoneng@nudt.edu.cn
* Correspondence: zhaobendong_nudt@163.com (B.Z.); 13990153582m@sina.cn (X.J.)
† These authors contributed equally to this work.

Abstract: To address the new functional requirements brought by the introduction of new weapons and new combat modes, a comprehensive survey of the research progress in the area of combat simulation software is performed from the perspective of software engineering. First, the top-level specification, simulation engine, and simulation framework of combat simulation software are reviewed. Then, several typical combat simulation software systems are demonstrated, and the relevant software frameworks are analyzed. Finally, combining the application prospect of artificial intelligence, metaverse, and other new technologies in combat simulation, the development trends of combat simulation software are presented, namely intellectualization, adaptation to an LVC (live, virtual, and constructive) system, and a more game-based experience. Based on a comprehensive comparison between the mentioned simulation frameworks, we believe that the AFSIM (Advanced framework for simulation, integration, and modeling) and the E-CARGO (Environments—classes, agents, roles, groups, and objects) are appropriate candidates for developing distributed combat simulation software.

Keywords: combat simulation; software framework; top-level specification; simulation engine; LVC system

Citation: Lu, F.; Hu, X.; Zhao, B.; Jiang, X.; Liu, D.; Lai, J.; Wang, Z. Review of the Research Progress in Combat Simulation Software. *Appl. Sci.* **2023**, *13*, 5571. https://doi.org/10.3390/app13095571

Academic Editor: Vito Conforti

Received: 4 April 2023
Revised: 27 April 2023
Accepted: 28 April 2023
Published: 30 April 2023

1. Introduction

By simulating military operations, combat simulation can study the operational rules, explore the operational guiding principles, and ultimately provide a scientific basis for military decision making. There are many methods to perform combat simulation, such as a live-fire drill, a sand table, war gaming, computer combat simulation, etc. Since its appearance in the 1950s, computer combat simulation [1] has developed into a very important means of combat simulation for many countries with the continuous progress in computer technology and system engineering theory.

In recent years, sophisticated systems such as hypersonic weapons [2] and unmanned aerial vehicles [3] have been successfully developed, which introduce higher requirements for the application and research of combat simulation technology. In particular, with the rapid development of artificial intelligence, big data, and other intelligent technologies, war is entering the era of intelligence. New weapons and new combat modes will inevitably introduce a lot of new functional requirements, even software reconstruction requirements, to the existing combat simulation software. Therefore, from the perspective of software engineering, we comprehensively review the research progress in the area of combat simulation software frameworks, and provide an outlook on the application prospects for new technologies, such as artificial intelligence and metaverse in combat simulation, in order to effectively support the development of distributed combat simulation software.

1.1. Literature Review Methodology

It is well known that performing research based on existing knowledge is the building block of all academic research activities [4]. However, it has become more and more complex, since knowledge production within the field of combat simulation is accelerating at a tremendous speed. Consequently, we utilized the semi-systematic review methodology [4] to collect and analyze the publications relevant to combat simulation.

As listed in Table 1, we developed appropriate keywords for each theme to search the relevant literature. For instance, we utilized four keywords (i.e., Simulation Framework, AFSIM [5], FLAMES (Flexible analysis and modeling effectiveness system) [6], and E-CARGO [7]) to search for the publications relevant to "Simulation Framework".

Table 1. Literature review keywords.

Theme	Keywords	Modifiers
Top-Level Specification	Specification; SMP [1]; DIS [2]; HLA [3]; LVC [4]	Simulation Software Architecture
Simulation Engine	Simulation Engine; Unreal Engine	Simulation Engine Open Source Combat Simulation
Simulation Framework	Simulation Framework; AFSIM; FLAMES E-CARGO	Component Collaboration
Typical Software	Combat Simulation; EADSIM [5]; JTLS-GO [6]	System

[1] SMP (Simulation Model Portability) [8]; [2] DIS (Distributed Interactive Simulation) [9]; [3] HLA (High Level Architecture) [10]; [4] LVC (Live Virtual Constructive) [11]; [5] EADSIM (Extended Air Defense Simulation) [12]; [6] JTLS-GO (Joint Theater Level Simulation–Global Operations) [13].

The keywords were used in a variety of appropriate databases as listed in Table 2. About 230 papers were obtained and stored in a bibliographic database (EndNote) in order to efficiently analyze the search results and provide flexible options to export the data for final formatting. In addition, we also browsed the homepages of various simulation software and accessed the functionalities of the products. We performed an analysis on the obtained papers and excluded about 180, when they did not strictly meet the relevant themes or were published decades ago.

Table 2. Sources used.

Data Source	Publications of Interest	Main Language
CNKI *	Academic articles	Chinese
Ei Village	Academic articles; e-books; reports	English
Web of Science	Academic articles; e-books; reports	English

* CNKI (China National Knowledge Infrastructure).

1.2. Manuscript Structure

The rest of the manuscript is structured as follows. Section 2 introduces several top-level specifications of simulation software. Section 3 demonstrates the simulation engines and simulation frameworks that could accelerate the development efficiency of simulation systems. Section 4 presents three typical simulation software widely utilized in current combat simulation, followed by a prospectus of the development trends in Section 5. Section 6 provides the concluding remarks.

2. Top-Level Specification of Simulation Software

Herein, the top specification of simulation software denotes the industry standard that leads and supports the development of simulation software. The research, establishment, promotion, and application of the top specifications for combat simulation software have attracted worldwide attention, and many specifications have been issued, such as DIS and HLA, as listed in Table 3.

Table 3. Simulation software specifications.

Specification	Promulgator	Year	Description
SMP2	Europe	2004	Simulation Model Portability
DIS	USA	1989	Distributed Interactive Simulations
HLA	USA	1995	High Level Architecture
LVC	USA	2007	Live, Virtual, and Constructive

2.1. SMP

SMP2 [8,14] is a simulation model portability specification designed by the European Space Agency. It ensures the platform independence and reusability of the model by defining high-level common concepts and low-level common types. As shown in Figure 1, we took China's aircraft carrier as an example to illustrate the architecture of the SMP2, which includes three different levels of abstraction.

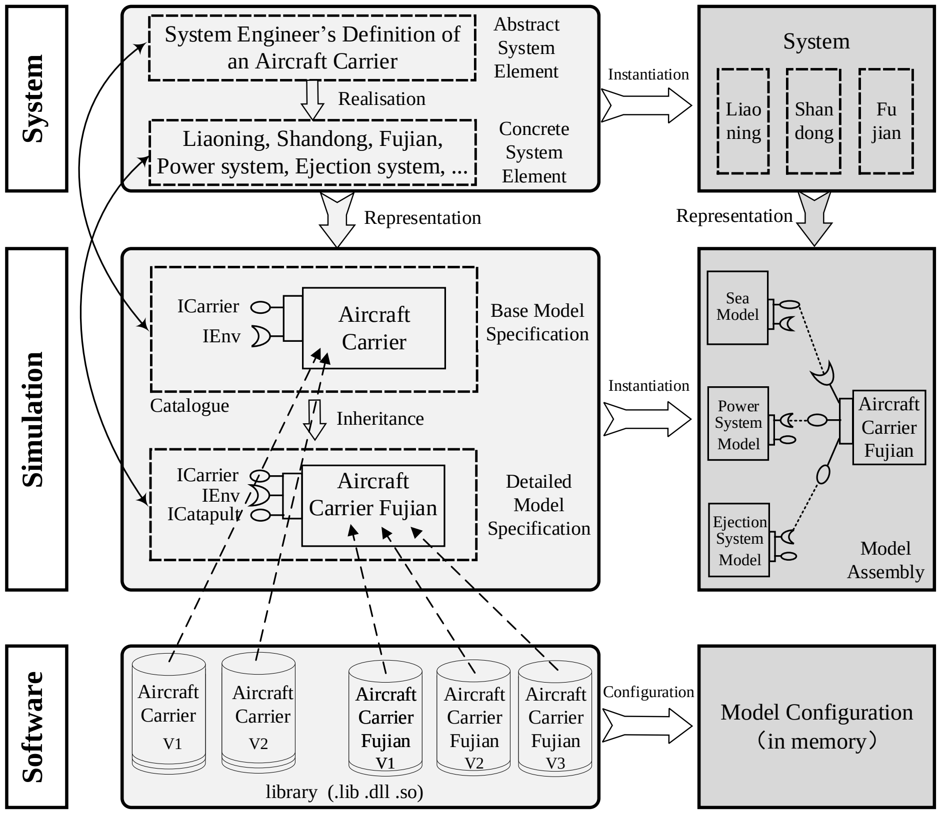

Figure 1. SMP2 high-level view.

- The *system* layer does not contain any abstract content and only describes the system objects in the real world to be simulated. In terms of abstract system elements, the system engineer can define that an aircraft carrier is composed of power, command and control, communication, and other subsystems according to the existing knowledge system, while in terms of specific system elements, China's existing aircraft carriers include three warships named Liaoning, Shandong, and Fujian.
- The *simulation* layer mainly describes the platform-independent model of the simulation system, focusing on the model specification (catalog file) and the integration, configuration, and scheduling of the model instances. Concretely speaking, Figure 1 defines the basic model specification of an aircraft carrier, and the Fujian aircraft carrier represents a derived specific model specification, which adds the interface of the ejection system model.

- The *software* layer describes the platform-related parts of the simulation system. SMP2 provides a convenient tool to generate a large number of C++ codes from the catalog file and finally presents various simulation models in the form of libraries.

2.2. DIS

Under the joint advocacy of the Defense Modeling and Simulation Office (DMSO) and the Defense Advanced Research Projects Agency (DARPA), the Distributed Interactive Simulation (DIS) standard [9,15] was formally put forward in 1989, and its related technology and architecture were reflected in the IEEE 1278 standard. The basic principle of DIS is to use a computer network to connect geographically dispersed simulators, realize information exchange between heterogeneous simulation subsystems through standardized communication interfaces, and finally form a comprehensive simulation environment.

Figure 2 shows the architecture of a typical DIS simulation system, which includes communication networks, system configuration services, and multiple simulation nodes, which consist of system management services, application simulations, core services, and distributed services.

- The *system configuration service* configures the the operation environment for the whole system in an offline manner, such as the initial state of each simulation entity and geographic information.
- The *system management service* is responsible for the start, monitoring, termination, and other management work of the whole simulation process.
- The *core service* is responsible for the scheduling and communication of simulation tasks on the simulation node.
- The *distribution service* is responsible for the distributed communication and time synchronization within each simulation node.
- The *application simulation* is composed of a variety of application tasks, which are realized with the cooperation of the core service and distribution service.

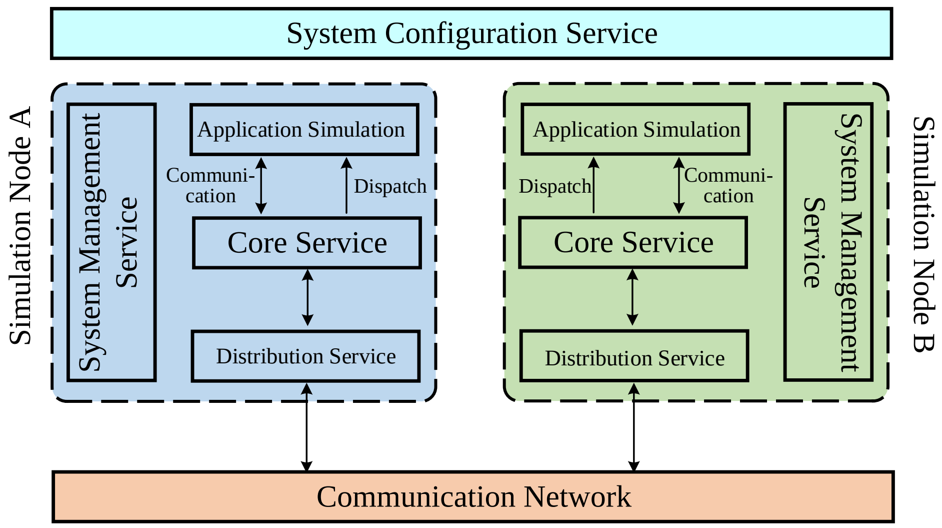

Figure 2. DIS simulation system architecture.

The DIS simulation system is simple to implement and has good fault tolerance. Its shortcomings are as follows:

1. The interoperability between distributed simulation systems can only be realized by standardizing the data communication rules, and systems with complex logical hierarchical relationships cannot be processed.
2. The information exchange between simulation entities is realized by broadcasting, and the utilization rate of bandwidth is very low.
3. Lack of time synchronization between simulators may lead to spatiotemporal inconsistency.

2.3. HLA

In order to solve the interoperability among simulation applications and the reusability of simulation components, the High Level Architecture (HLA) [10] was proposed by DMSO in 1995 and is still widely utilized in various simulation scenarios [16,17]. In 2011, the American Computer Society issued the distributed simulation engineering and execution process standard IEEE 1730 [18], which unified the engineering and execution processes related to distributed simulation such as DIS and HLA.

The HLA is divided into three levels: *federation*, *federate*, and *object*. The *federation* is a simulation system to achieve specific simulation objectives, and it usually consists of multiple *federates*, each of which is composed of several *objects*. Different *federates* cooperate to complete the simulation objectives of the federation.

The HLA is composed of rules, interface specifications, and object model templates, and the relevant contents have formed IEEE 1516 series standards. HLA rules ensure that all *federates* can communicate and interact correctly when the *federation* is running. The HLA interface specification determines all the service functions of the runtime environment (RTI) and the callback functions that the relevant *federates* should provide. The HLA object model template is a description of the *Federation Object Mode* file, providing a common method and model standard for recording information during the running process of the federation.

As shown in Figure 3, the HLA provides general and relatively independent services through an RTI, separating the simulation application from the underlying support environment. Concretely speaking, the HLA separates the specific simulation function implementation, simulation operation management, and the underlying communication transmission, so that each part can be developed relatively independently and make full use of the advanced technology in their respective fields.

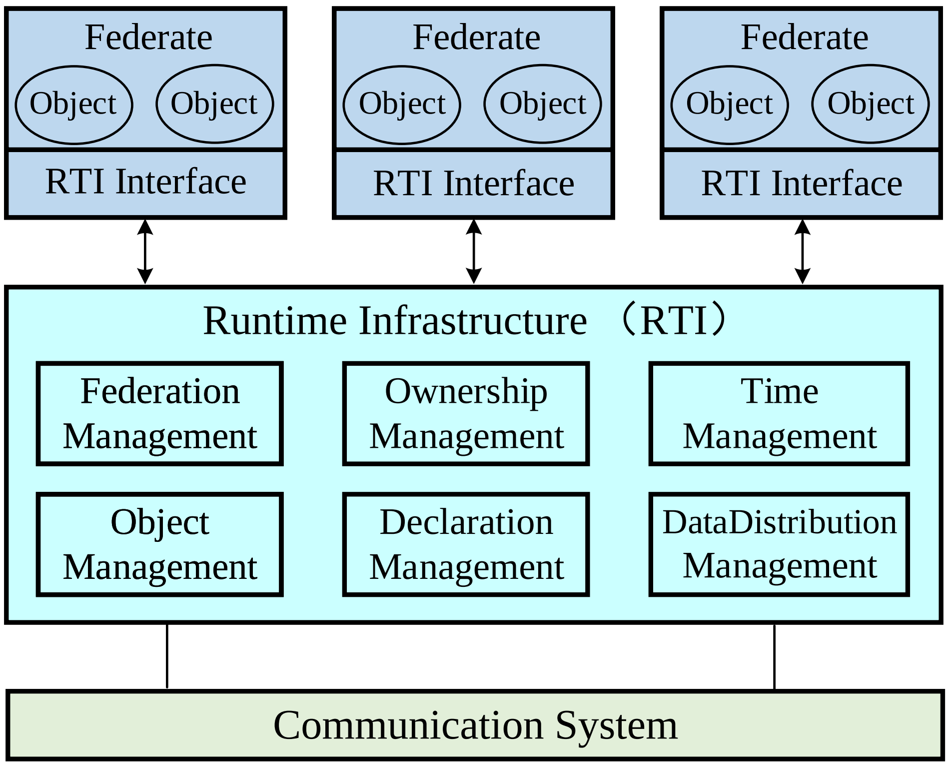

Figure 3. Construction of the federation in an HLA.

2.4. LVC

The Simulation Interoperability Standards Organization (SISO) [19] was established in 1996 and is committed to promoting the interoperability of modeling and simulation. The LVC was originally proposed by SISO as a classification method for model and simulation, where L, V, and C refer to live simulation, virtual simulation, and constructive simulation, respectively.

In order to realize the joint simulation test that is multirange, multi-agency, and multiplatform in a complex environment, the U.S. military proposed the LVC simulation architecture development roadmap [11] in 2007 to improve the interoperability of a multi-architecture simulation environment. In this roadmap, the ultimate goal of the modeling and simulation field is to create a unified LVC integration architecture, which can quickly integrate models and form an effective LVC environment for tactical training, coordination, and operational evaluation.

In 2012, SISO proposed the Layered Simulation Architecture (LSA) [20], which could unify the distributed simulation architecture such as DIS and HLA through the DDS (Data Distribution Service) [21]. At present, a Spanish company, named Nextel Aerospace Defense & Security, has developed Simware [22] that can fully support the LSA architecture.

Figure 4 shows the architecture of the LSA simulation system, which is composed of layers of *application*, *coupler*, *middleware*, and *protocol*.

- The *application* layer includes two types of simulation applications: the legacy applications based on existing simulation standards (e.g., DIS, TENA [23], and HLA), and new simulation applications based on DDS and other new architectures.
- The *coupler* layer provides various couplers between the *application* layer and the *middleware* layer, including interfaces, gateways, and services. The legacy simulation systems can be seamlessly connected to the middleware through the specific interface (e.g., DIS-DDS, TENA-DDS, and HLA-DDS), gateway, and various simulation services. New simulation systems, such as the ones based on DDS, can connect to the *middleware* layer with public interfaces and gateways.
- The *middleware* layer is the key infrastructure layer for data exchange, through which each simulation system can realize mutual operation. The LSA currently selects the data-centric DDS as the middleware to build a low-latency, high-reliability, and scalable communication architecture for the distributed simulation system.
- The *protocol* layer currently utilizes the interactive real-time publish–subscribe protocol DDSI [24] as the underlying communication protocol for the DDS middleware.

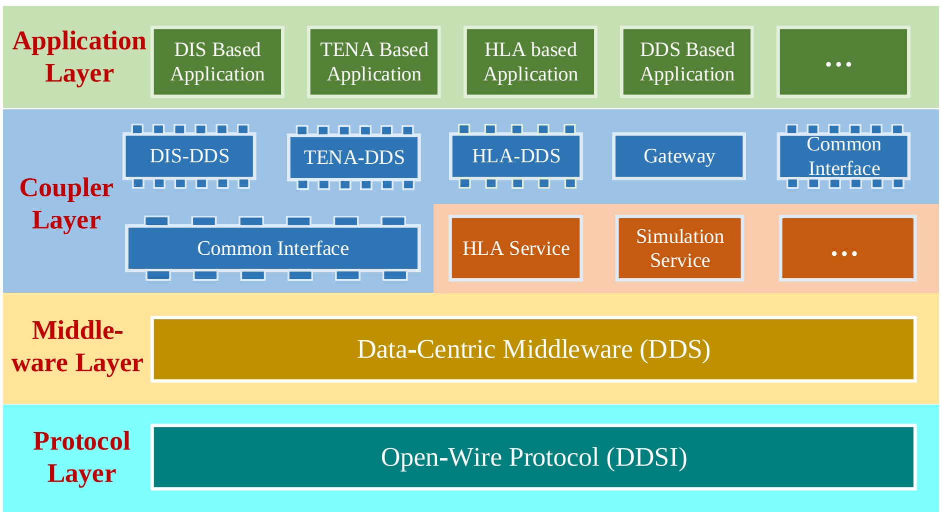

Figure 4. LSA simulation system architecture.

3. Simulation Engine and Simulation Framework

In a simulation system, the simulation engine is responsible for time stepping, operation scheduling, simulation control, and data storage, which performs as the "engine" of the whole system. The simulation framework provides a toolkit (e.g., modeling and resource services) for developing a simulation system and usually includes independent or tightly coupled simulation engines. With respect to the numerous works and products related to simulation engines and simulation frameworks, we selected several typical works to describe the relevant research progress.

3.1. Simulation Engine

According to the SMP2.0 standard, Li et al. [25] designed and implemented a simulation engine compatible with SMP2.0, which included key components such as time management and global event management, and verified the effectiveness of their engine through navigation applications.

In terms of the HLA engine, Ling et al. [26] treated the simulation engine as a special *federate*, including a simulation scheduler, simulation controller, and data recorder, aiming to improve the application development and simulation operation efficiency of a command automation effectiveness simulation and evaluation system.

Unity [27] and Unreal Engine [28] (referred to as UE) are famous commercial game engines, which have been widely used to build high-fidelity visual simulation systems. For example, Zheng et al. [29] built a full-size high-immersion visual simulation environment based on the Unity engine for the attack–defense confrontation of an aircraft. For another example, Ginchev et al. [30] developed a low budget simulator of an M1A2 Abrams main battle tank based on UE4 and realized the tank's virtual representation.

Besides commercial game engines, there are many open source engines, such as OGRE [31], which have been used in a large number of production projects in diverse areas, such as games, simulators, educational software, interactive art, scientific visualization, and others. For instance, the OGRE engine was utilized by Tang et al. [32] in their framework named TPL.Frame to tackle the existing difficulties of engineering simulation platform development through open-source components and homemade resources.

3.2. Simulation Framework

3.2.1. FLAMES

FLAMES [6] (Flexible analysis and modeling effectiveness system) is a simulation software framework developed by American Ternion Company in the mid-1980s, which was released in 2001; it can provide customized weapons and equipment simulation services for the U.S. military. As shown in Figure 5, we can develop simulation systems conforming to the LVC architecture of the U.S. military based on the FLAMES framework. It is capable of supporting the development of simulation systems for weapon platforms and tactical and campaign scenarios and providing the development framework of the equipment model, cognitive model, message model, and environment model.

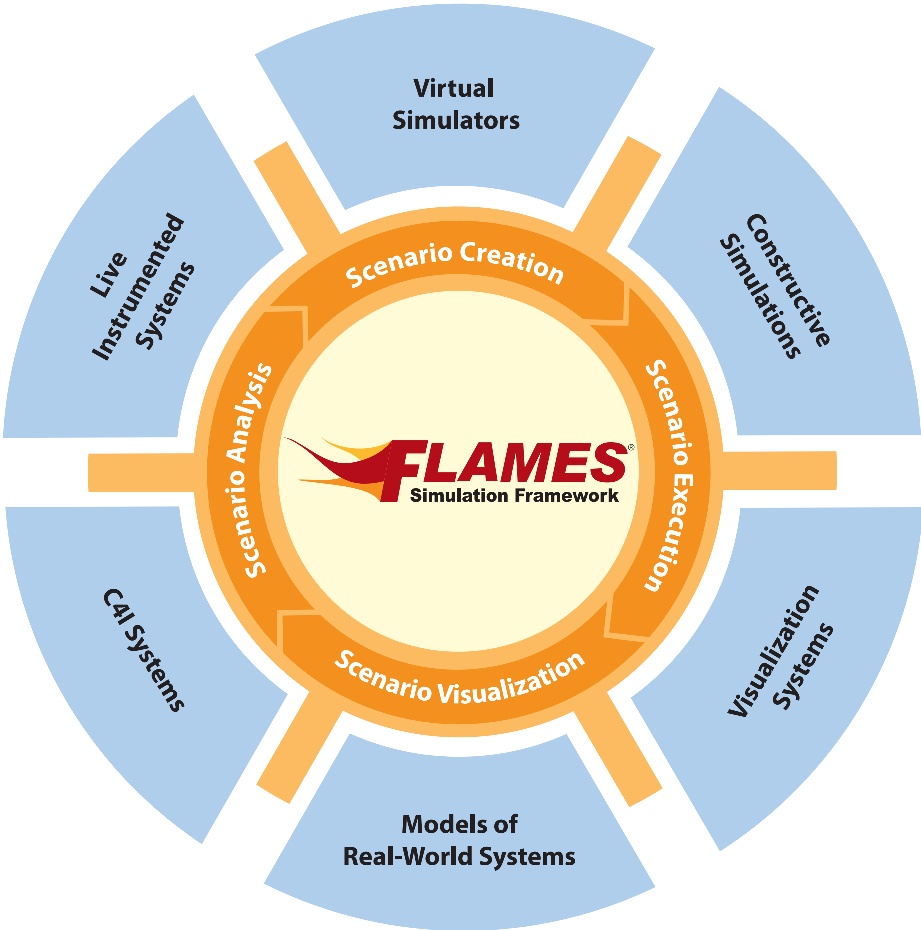

Figure 5. FLAMES-based simulation system [33].

3.2.2. XSim Studio

XSim Studio (XSIM for short) [34] is a scalable simulation framework for military simulation developed by Huaru Technology Co., Ltd. It takes the component-based modeling and parallel discrete event simulation as core technologies and supports C4ISR system modeling. It provides comprehensive support, such as integrated development, operation management, and resource services at all stages of model preparation, scheme formulation, system operation, analysis, and evaluation.

3.2.3. AFSIM

The AFSIM (Advanced framework for simulation, integration and modeling) [5] was developed by MDS [35] and can support various mission objectives for Air Force, Navy, and joint-sponsored initiatives. It can be utilized in the areas of multiplatform and multisensor simulation for manned, UAV/UAS, and other defense platforms in conventional and anti-access/area-denial scenarios. Its capabilities include resource management, variable mission effectiveness, and a physics-based analysis of the sensor performance, such as algorithms related to radar search, finding, fixing, and tracking and layered fusion techniques. As shown in Figure 6, the AFSIM is characterized by a component-based architecture, which utilizes an underlying generic component class to derive all components. This technique allows the access of existed components via naming and can ease the addition and removal of certain component types, which provides the means to maintain a release version with no weapons or electronic warfare capabilities. The AFSIM is very attractive, and many domain experts [36–38] have performed various investigations based on the AFSIM framework.

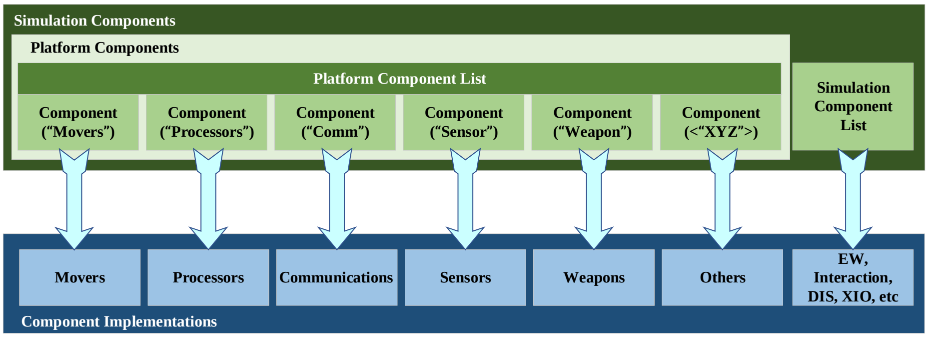

Figure 6. Component-based architecture.

3.2.4. E-CARGO

There are many challenging requirements in combat simulation software, e.g., trust establishment among soldiers and commanders, highly autonomous and well-equipped individual soldiers, and situation awareness in the simulation systems. E-CARGO (Environments—classes, agents, roles, groups, and objects) [7] is an abstract model providing a framework for role-based collaboration (RBC) [39] that can address the aforementioned requirements. E-CARGO can help analyze, design, and simulate complex systems by formalizing each of them to a nine-tuple: $< C, O, A, M, R, E, G, H, s_0 >$, whose elements correspond to classes, objects, agents, messages, roles, environments, groups, human users, and the system's initial state, respectively.

- The first-class components in E-CARGO are classes, objects, groups, agents, environments, and roles. They are the fundamental structures and depict the constructions or compositions of a complex system.
- The second-class components include messages, the initial state, and human users, which reveal the complexity of systems. For instance, messages reflect the numerous ways for the components to communicate, interact, coordinate, and cooperate.

E-CARGO and RBC are a good fit for simulating complex systems and have been utilized in various scenarios [7,40,41]. For example, Zhu and Yu [42] investigated the team management problem with the E-CARGO framework and revealed interesting conclusions that could help decision makers understand the complexity of the assignment and choose a pertinent path when conducting team management.

3.2.5. Comparison and Analysis

From the perspective of developing distributed combat simulation systems, we selected four metrics (i.e., distributed ability, modularity, scalability, and generality) and performed a general comparison among the aforementioned four simulation frameworks. Here, *distributed ability* represents the ability to support a simulation on distributed computers linked via a communicating network; *modularity* measures how well the framework can be decomposed into smaller pieces with standardized interfaces; *scalability* denotes how easy it is to grow or shrink the framework with respect to the simulation model, problem size, and secondary development; and *generality* describes the application's ability to be applied to other scenarios.

As shown in Table 4, we marked a number of stars on the four selected metrics for each software. Since we could not obtain the FLAMES, AFSIM, and E-CARGO software, the results were evaluated according to the public information released by the relevant software developers. Our future work includes trying to obtain the software for experimentation and verification.

Table 4. Comparison between the simulation frameworks.

Framework	Distributed Ability	Modularity	Scalability	Generality
FLAMES	★★★★	★★★★	★★★★	★★★
XSIM	★★★	★★★★★	★★★	★★★
AFSIM	★★★★★	★★★★★	★★★★★	★★★★
E-CARGO	★★★	★★★★	★★★★★	★★★★★

We observed the following based on the results listed in Table 4:

- AFSIM and FLAMES have a very good distributed ability.
- Software modularization has become an important design principle in software engineering, and all four frameworks have good modularity.
- AFSIM and E-CARGO perform relatively better than the other two frameworks.
- E-CARGO outperforms the others in generality, since it can simulate complex problems in collaboration and complex systems.

4. Typical Combat Simulation Software

4.1. EADSim

EADSIM (Extended Air Defense Simulation) [12] is a multi-scenario simulation system for missile and space operations, providing warfighters with an overall solution for analysis, training, and operational planning. After nearly 30 years of development, EADSIM has become one of the most widely used simulation systems in the world. For example, Jordan et al. [43] utilized EADSIM for high detail virtual prototyping of defensive systems, and Hong et al. [44] generated a fictitious asset-based defence scenario to illustrate the solution-space exploration across three simulation phases.

Figure 7 shows the architecture of the EADSIM, which is divided into three parts: the simulation setup, the run-time models, and the post-simulation analysis.

(1) Simulation setup. With the help of a graphical interface, the users can set the basic parameters, check the parameter consistency and boundary, and generate operational scenarios.

(2) Run-time model. In the C3I decision-making process, the C3I model deals with an entity's command and decision making; the flight processing model is responsible for updating the movement status of battlefield entities; the detection model simulates the working condition of the sensor; and the propagation model undertakes the simulation of network connectivity. In terms of system models, EADSIM supports operations such as air defense, air-to-ground, air warfare, electronic warfare, and cyber warfare and can model weapons, sensors, and a battlefield environment.

(3) Post-simulation analysis. This module carries out statistical analysis on the simulation data, can realize the visualization of the combat simulation process, and can automatically generate reports such as engagement statistics, communication statistics, detection statistics, etc.

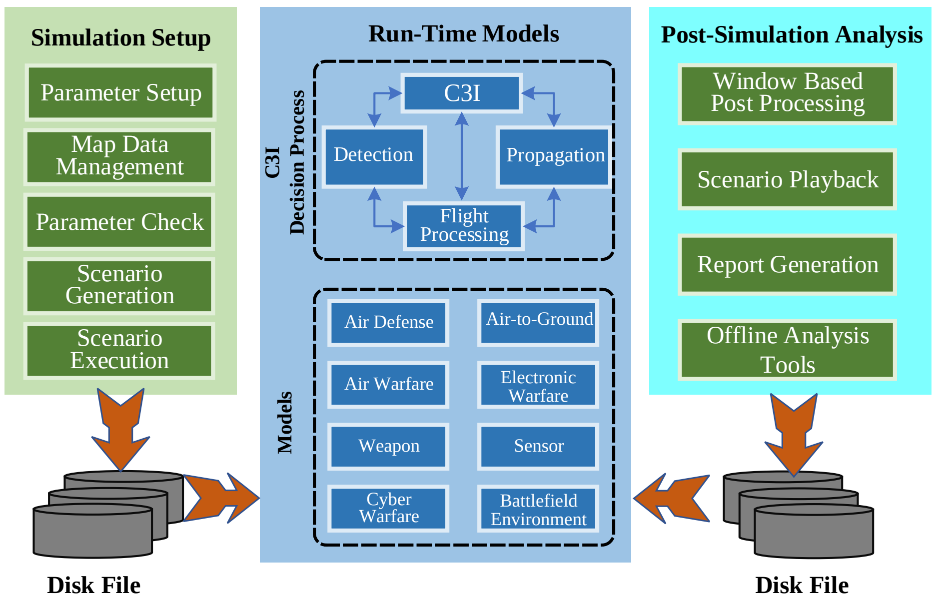

Figure 7. EADSIM architecture.

4.2. JTLS-GO

The JTLS-GO (Joint Theater Level Simulation—Global Operations) [13,45] is an interactive internet-enabled simulation software that models multisided air, ground, and naval civil–military operations with logistical and intelligence support. It is frequently used as a training support model that is theater-independent and does not require knowledge of programming to operate effectively. Its primary focus is conventional joint and combined operations at the operational level of war.

With respect to the evolution of its software architecture, JTLS changed from the C/S architecture to the web-enabled architecture in 2005. Concretely speaking, the simulation personnel can use the "interface program WHIP" and "interactive recall and playback program" based on the web host to carry out planning preparation, simulation operation, and result analysis.

JTLS-GO provides its functions through six models, namely force control, ground operations, air operations, maritime operations, logistics support, and C3I.

(1) Force control. This includes command authority, relations between all parties, and engagement rules. The force commander can assign operations to the subordinates involved in the combat simulation. The participants need to determine the relationship between each other. Each combat unit has a specific set of engagement rules for different adversaries.
(2) Ground operations. The commander can set engagement rules, establish a new ground maneuver route, and issue commands such as attack, defense, and fire support.
(3) Air operations. There are three ways to implement air operations, namely automatic air mission command generator, manual input of instructions, and the combination of the two methods. The air mission command generator can create task packages composed of different types of aircrafts.
(4) Marine operations. This can simulate ship-to-ship missile attacks, naval gunfire, amphibious attacks, regional patrol and anti-submarine warfare, air operations based on aircraft carriers, etc.
(5) Logistics support. This includes automatic resupply, integrated supply, emergency supply, supply transportation, force transportation, etc.
(6) C3I. The information obtained by any combat unit can be used by the whole force, so that the commander can master the enemy situation and make a thorough and timely plan.

4.3. Fight Tonight

"Fight Tonight" [46] is a project launched by the United States Air Force Research Laboratory in December 2021 and was awarded to a Raytheon BBN-led team in December 2022. The Air Force is looking for ways to combine artificial intelligence and interactive gaming technology to revolutionize and dramatically shorten the process of planning complex air attack operations, namely from 36 h or more to four. The developed prototype system will eventually be interconnected with external systems to support operational decision making in the full scenario. It is an important feature of the project to use commercial game technology to build an intuitive and interactive interface to improve human understanding of the complex combat environment.

Figure 8 demonstrates the conceptual function composition of the "Fight Tonight" project, which includes interactive plan refinement and plan gaming/outcome analysis.

(1) Interactive plan refinement. This function generates plans through resource allocation and task priority setting, which can help users design force allocation schemes, understand the available schemes, and evaluate the impact of priority adjustment. The system generates a plan prototype according to the guidelines provided by users; then, users employ the automatic analysis function to evaluate other alternatives of the plan. At the same time, the system continues to optimize the alternatives and improve the accuracy of the plan through calculation.

(2) Plan gaming and result analysis. The deduction platform will provide in-depth insights and experience for operational planning, which enables planners to analyze the expected performance and update the operational plan according to the simulation results.

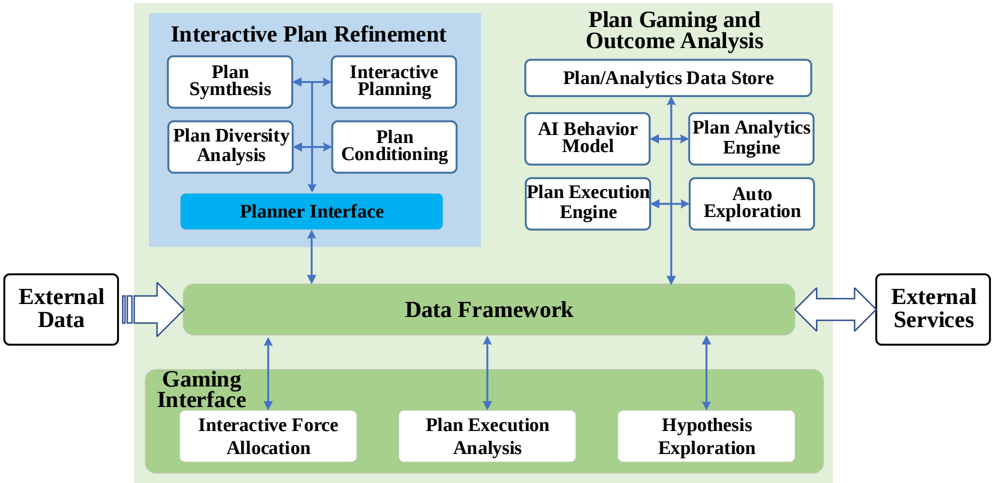

Figure 8. Conceptional function diagram.

5. Development Trends of Combat Simulation Software

5.1. Combat Simulation Software Should Be More Intelligent

Driven by a new round of technological revolution, industrial revolution, and military revolution, the outstanding performance of unmanned combat platforms in local conflicts has been deeply rooted in the hearts of the people. Intelligent warfare is taking shape, showing characteristics such as sudden outbreak, transparent environment, fuzzy space, and system integration [47].

With regard to the advanced warfare mode with intelligence as its main feature, Europe and the USA have begun to utilize artificial intelligence (AI) technology to promote "independent adaptation" and "independent action" of weapons and equipment. For example, AI techniques have been integrated into existing weapon systems to improve the autonomy of weapons and equipment in finding, identifying, tracking, and destroying targets. Another example is to vigorously develop the unmanned combat systems so that they can undertake reconnaissance, tracking and surveillance, target indication, battlefield material distribution, fire attack, and other tasks on the battlefield. In terms of operational concepts, the US military has proposed "loyal wingman", manned/unmanned intelligent cooperation, "swarm" intelligent cooperation, and other operational styles based on intelligent UAV clusters [48].

Therefore, the combat simulation software should be deeply combined with AI technology so as to gradually realize the intelligent combat simulation software, to simulate intelligent weapons in the future intelligent war, and to improve the accuracy of combat simulation.

5.2. Combat Simulation Software Should Be More Suitable for LVC Systematization

Systems confrontation is the main form of current local war and future war. The LVC integrated architecture proposed by the U.S. military is suitable for building a more complex and realistic training scenario required for systems confrontation training. The LVC is the best way to achieve systematic combat training and the best technical path to verify the contribution rate of an equipment system in the future. The construction of an LVC is a complex system project. The U.S. military implemented it in accordance with the three lines of "L/LC", "VC", and "LVC", simultaneously. For example, the U.S. military verified the ability of the existing fighters and simulators of the U.S. military to join the LVC through the "Secure LVC Advanced Training Environment" project [49] and applied it to the "Red Flag" military exercise in 2018.

In a sense, in the process of adapting to the LVC system, the combat simulation software can learn from metaverse technologies. The metaverse is essentially an advanced cyberspace, which is based on the digital information world, and can express the criss-cross of reality and reality. With the influx of high-tech companies and industrial capital, the metaverse technologies will surely achieve explosive development, which can catalyze and accelerate the process of adapting the combat simulation software to the LVC system.

5.3. Combat Simulation Software Will Provide a Much More User-Friendly Experience

In recent years, emerging computer game rendering, extended reality, and other technologies [50] have been continuously applied to the development of combat simulation software. With the help of 3D animation effects and real-time interactive functions, participants can experience a realistic battlefield environment through multiple senses, realize immersive drills, and improve technical and tactical capabilities. The United States Army has conducted a lot of work in the development of the application of a game-based simulation training system. For instance, VIRTSIM [51] is a highly immersive virtual reality system, which can pit squads of players against each other in a realistic virtual reality combat environment, and Virtual Battlespace 3 [52] can support a variety of training objectives, including terrain reconnaissance, convoy operations, counter IED, and fire support.

On the other hand, military games [53–56] are becoming more and more practical and perform as new platforms for combat simulation. For example, the COMMAND [53] series game developed by Warfare Sims has been highly valued by the military and defense agencies of many countries. It develops expansion packages based on the changing political and military activities. For example, the "Chains of War battlefield" package is based on hypothetical conflicts between China, the USA, and their respective allies in the future.

6. Conclusions

New equipment such as hypersonic weapons and new combat modes such as intelligent unmanned combat introduce a lot of new functional requirements to existing combat simulation software. From the perspective of software engineering, we systematically summarized the research progress in the top-level specification and simulation frameworks of combat simulation software. We also selected three typical combat simulation software systems to analyze their software architectures. Combining the prospects of artificial intelligence, metaverse, and other new technologies in combat simulation, we believe that the development trends of combat simulation software are intellectualization, adaptation to an LVC (live, virtual, and constructive) system, and a more game-based experience. Based on a comprehensive comparison between the mentioned simulation frameworks, we state that AFSIM and E-CARGO are appropriate candidates for developing distributed combat simulation software.

Our future work includes two aspects. On the one hand, we will develop distributed combat simulation software addressing the newly emerging requirements; on the other hand, we will perform experiments on and verifications of combat simulation software.

Author Contributions: Conceptualization, F.L. and X.J.; investigation, X.H. and B.Z.; resources, J.L. and Z.W.; writing—original draft preparation, F.L.; writing—review and editing, F.L. and Z.W.; visualization, B.Z.; supervision, X.J.; project administration, X.J.; funding acquisition, D.L. and X.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Scientific Research Projects under grant no. JK20211A010109.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. US Operations Research Office. Software: Carmonette. 1953. Available online: https://handwiki.org/wiki/Software:Carmonette (accessed on 25 April 2023).
2. Kessler, A. Russian Hypersonic Glide Vehicles: What to Know and What to Fear. *Orbis* **2022**, *66*, 213–223. [CrossRef]
3. Laghari, A.A.; Jumani, A.K.; Laghari, R.A.; Nawaz, H. Unmanned aerial vehicles: A review. *Cogn. Robot.* **2023**, *3*, 8–22. [CrossRef]
4. Snyder, H. Literature review as a research methodology: An overview and guidelines Hannah Snyder. *J. Bus. Res.* **2019**, *104*, 333–339. [CrossRef]
5. 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). In Proceedings of the 13th International Conference on Scientific Computing, Las Vegas, NV, USA, 27–30 July 2015; IEEE: Las Vegas, NV, USA, 2015.
6. Ternion Corporation. FLAMES Simulation Framework. 2023. Available online: https://flamesframework.com/ (accessed on 10 January 2023).
7. Zhu, H.; Kennedy, T.; Ma, H.; Zhang, H.; Lei, B.; Gningue, Y. A Simulation System for Flexible Transit Services based on E-CARGO. In Proceedings of the 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 27–29 March 2018; IEEE: New York, NY, USA, 2018.
8. Lei, Y.; Wang, W.; Li, Q.; Zhu, Y. A transformation model from DEVS to SMP2 based on MDA. *Simul. Model. Pract. Theory* **2009**, *17*, 1690–1709.
9. Rackl, G.; de Stefani, F.; Héran, F.; Pasquarelli, A.; Ludwig, T. Airport simulation using CORBA and DIS. *Future Gener. Comput. Syst.* **2000**, *16*, 465–472. [CrossRef]
10. Dahmann, J. High Level Architecture for simulation. In Proceedings of the First International Workshop on Distributed Interactive Simulation and Real Time Applications, Eilat, Israel, 9–10 January 1997; pp. 9–14.
11. Amy E.; Henninger, D.; Cutts, M.L. *Live Virtual Constructive Architecture Roadmap (LVCAR) Final Report*; Technical Report; Institute for Defense Analyses: Alexandria, Egypt, 2008.
12. Teledyne Brown Engineering. EADSIM Homepage. 2023. Available online: https://www.tbe.com/missionsystems/eadsim (accessed on 27 April 2023).
13. ROLANDS & ASSOCIATES Corporation. JTLS-GO Homepage. 2023. Available online: https://www.rolands.com/jtls/j_over.php (accessed on 20 April 2023).
14. European Space Agency. *SMP 2.0 Handbook*; Technical Report EGOS-SIM-GEN-TN-0099; European Space Agency: Paris, France, 2005.
15. Open DIS. DIS: The Missing Handbook. 2023. Available online: https://open-dis.github.io/dis-tutorial/DIS_Background.html (accessed on 25 April 2023).
16. Magoua, J.J.; Wang, F.; Li, N. High level architecture-based framework for modeling interdependent critical infrastructure systems. *Simul. Model. Pract. Theory* **2022**, *118*, 102529. [CrossRef]
17. Celik, T.; Tekinerdogan, B. S-IDE: A tool framework for optimizing deployment architecture of High Level Architecture based simulation systems. *J. Syst. Softw.* **2013**, *86*, 2520–2541. [CrossRef]
18. *IEEE Std 1730.1-2013*; IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO). SISO: New York, NY, USA, 2013; pp. 1–91.
19. SISO. Simulation Interoperability Standards Organization Homepage. 2023. Available online: https://www.sisostds.org/ (accessed on 26 April 2023).
20. Martinez, J.R. Structural Organization of LSA Architecture. 2023. Available online: https://www.sisostds.org/DigitalLibrary.aspx (accessed on 26 April 2023).
21. OMG. Data-Distribution Service Homepage. 2023. Available online: https://www.omg.org/omg-dds-portal/ (accessed on 27 April 2023).
22. Nextel Aerospace Defence & Security. Simware Homepage. 2023. Available online: http://www.simware.es/ (accessed on 13 January 2023).
23. TENA-SDA Organization. TENA Homepage. 2023. Available online: https://www.tena-sda.org/ (accessed on 27 April 2023).
24. OMG. About the DDS Interoperability Wire Protocol Specification Version 2.5. 2023. Available online: https://www.omg.org/spec/DDSI-RTPS/2.5/About-DDSI-RTPS/ (accessed on 27 April 2023).
25. Li, Q.; Wang, C.; Wang, W.; Zhu, Y. Design and Implementation on Simulation Engine Compliant with SMP2.0. *J. Syst. Simul.* **2008**, *20*, 6622–6626+6630.
26. Ling, Y.; Zhang, X.; Liao, H.; Gu, X.; Qiu, D. Design and Implementation of Simulation Engine Based on HLA Federate Template. *J. Syst. Simul.* **2005**, *17*, 2629–2632.
27. Unity Technologies. Unity Homepage. 2023. Available online: https://unity.com/ (accessed on 27 April 2023).
28. Epic Games Incorporation. Unreal Engine Homepage. 2023. Available online: https://www.unrealengine.com/en-US (accessed on 27 April 2023).
29. Zheng, X.P.; Teng, Y.; Gao, L.; Zhou, Z.H. Simulating the Aircraft Attack-defence Confrontation Based on Unity3D. *Command Control Simul.* **2016**, *38*, 102–106.
30. Ginchev, D.; Stavrev, S. A low-cost battle tank simulator using Unreal Engine 4 and open-hardware microcontrollers. In Proceedings of the IEEE XXIX International Scientific Conference Electronics, Sozopol, Bulgaria, 16–18 September 2020.
31. OGRE Organization. OGRE Homepage. 2023. Available online: https://www.ogre3d.org/ (accessed on 27 April 2023).

32. Tang, Z.; Zhang, W.; Wu, P. A holistic framework for engineering simulation platform development gluing open-source and home-made software resources. *Adv. Eng. Softw.* **2014**, *76*, 99–109. [CrossRef]
33. Ternion Corporation. FLAMES Overview. 2023. Available online: https://flamesframework.com/wp-content/uploads/2021/08/FLAMES-Constructive-Simulation-Framework-Brochure.pdf (accessed on 11 January 2023).
34. Huaru Tech. XSimStudio Scalable Simulation Platform. 2023. Available online: http://www.huaru.com.cn/technology/file.html (accessed on 26 January 2023).
35. MDS. MDS Homepage. 2023. Available online: https://www.mdsse.com/ (accessed on 25 April 2023).
36. Thompson, J.S.; Hodson, D.D. AFSIMs pseudo-realtime hybrid simulation software design. *J. Def. Model. Simul.* **2022**, *19*, 575–587. [CrossRef]
37. Allison, D.; Shimmin, K.; Schley, W.; Bryson, D. Automated 6DOF Model Generation and Actuator Sizing within AFSIM. In *2019 SAE AeroTech Americas Congress and Exhibition*; United States Air Force: Charleston, SC, USA, 2019.
38. Hanlon, N.; Garcia, E.; Casbeer, D.W.; Pachter, M. AFSIM implementation and simulation of the active target defense differential game. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018; AIAA: Kissimmee, FL, USA, 2018.
39. Zhu, H.; Zhou, M. Role-based collaboration and its kernel mechanisms. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2006**, *36*, 578–589.
40. Zhu, H. Computational Social Simulation with E-CARGO: Comparison Between Collectivism and Individualism. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 1345–1357. [CrossRef]
41. Zhang, C.; Cui, H.; Cao, L.; Wang, F.; Yang, Y. Group role assignment strategies in microservices team based on E-CARGO model. *Knowl.-Based Syst.* **2022**, *248*, 108910. [CrossRef]
42. Zhu, H.; Yu, Z. Team Performance due to Agent Conflicts: E-CARGO Simulations. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Melbourne, VIC, Australia, 17–20 October 2021; pp. 2816–2821.
43. Jordan, W.; Cooper, A.; McAnally, M.; Binner, M. Virtual prototyping with EADSIM. In *Proceedings of the SISO European Simulation Interoperability Workshop 2007*; SISO: Genoa, Italy, 2007; pp. 109–116.
44. Hong, L.J.; Wee, T.C.; Kiat, L.Y. Tradespace exploration for military simulations. In *Proceedings of the Simulation Series*; The Society for Modeling and Simulation International: San Diego, CA, USA, 2012; Volume 44, pp. 106–113.
45. Boiling, R.H. The Joint Theater Level Simulation in Military Operations Other Than War. In Proceedings of the 1995 Winter Simulation Conference, Arlington, VA, USA, 3–6 December 1995.
46. Air & Space Forces Magazine. AFRL's 'Fight Tonight' to Prototype AI and Gaming Tech for Attack Planning. 2023. Available online: https://www.airandspaceforces.com/afrls-fight-tonight-to-prototype-ai-and-gaming-tech-for-attack-planning/ (accessed on 27 April 2023).
47. Liu, C.; He, J.; Li, Q.; Wang, M. An Air Combat Simulation System for Intelligent Decision-Making. In Proceedings of the 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 22–23 August 2020; IEEE: Daegu, Republic of Korea, 2020; Volume 2, pp. 104–108.
48. Chen, S.; Li, D.; Sun, P.; Fang, J. Analysis on the Development and Influence of Intelligent Unmanned Aerial Vehicle Cluster in U. S. Army. *J. CAEIT* **2021**, *16*, 1113–1118.
49. The Air Force Research Laboratory. Secure Live Virtual Constructive Advanced Training Environment. 2023. Available online: https://afresearchlab.com/technology/human-performance/secure-live-virtual-constructive-advanced-training-environment/ (accessed on 27 April 2023).
50. Wu, Z.; Pan, L.; Yu, M.; Liu, J.; Mei, D. A game-based approach for designing a collaborative evolution mechanism for unmanned swarms on community networks. *Sci. Rep.* **2022**, *12*, 18892. [CrossRef]
51. Lang B. VIRTSIM Is the Virtual Reality Platform That Gamers Crave but Can't Have. 2023. Available online: https://www.roadtovr.com/virtsim-virtual-reality-platform/ (accessed on 27 April 2023).
52. Lucas, R. Virtual Battlespace 3 Provides Training Opportunities During Coronavirus Restrictions. 2023. Available online: https://www.dvidshub.net/news/372080/virtual-battlespace-3-provides-training-opportunities-during-coronavirus-restrictions (accessed on 27 April 2023).
53. Warfare Sims. Command: Modern Operations. 2023. Available online: https://command.matrixgames.com/ (accessed on 27 April 2023).
54. Microsoft. Microsoft Combat Flight Simulator Homepage. 2023. Available online: https://www.flightsimulator.com/ (accessed on 27 April 2023).
55. Terasynth, Inc. TWS ATC/GCI: Next Level Simulation Experience. 2023. Available online: https://www.terasynth.com/tws-news/tws-atc-gci-next-level-simulation-experience (accessed on 27 April 2023).
56. Eagle Dynamics SA. Digital Combat Simulator World. 2023. Available online: https://www.digitalcombatsimulator.com/en/products/world/ (accessed on 27 April 2023).