

# HLA Run Time Infrastructure: A Comparative Study

AMIR AKRAM\*, MUHAMMAD SHAHZAD SARFRAZ\*, AND UMAR SHOAIB\*

RECEIVED ON 27.03.2017 ACCEPTED ON 19.02.2019

## ABSTRACT

Distributed computer simulation systems use a general-purpose architecture known as HLA (High Level Architecture). HLA aims to provide common architecture for all types of distributed modeling and simulations by providing interoperability and reusability among simulations. A middleware known as RTI (Run Time Infrastructure) provides distributed simulation services under HLA. RTI offers a communication framework which is necessary for federates to interact during simulation process. Many RTI implementations are available commercially and are open source to achieve reusability and interoperability by implementing HLA. However, functionality and performance of RTI is based on its structure. This review presents the performance analysis of multiple RTI implementations, so that the best suitable RTI can be selected for a given simulation scenario and provides the structural comparison of central and service distributed RTI. It also presents the performance analysis of multiple RTI implementations and provides the structural comparison of CeRTI (Central Run Time Infrastructure) and SDRTI (Service Distributed Run Time Infrastructure). Many simulations use HLA, and RTI as basic part of these simulations. This comparative study describes characteristics of different RTIs, their comparisons and implementations that will help the reader to select suitable RTI according to environment and requirements.

**Key Words:** High Level Architecture, Simulations, Run Time Infrastructure, Distributed Simulation Services, Federate, Central Run Time Infrastructure, Service Distributed Run Time Infrastructure.

## 1. INTRODUCTION

Many complex simulations involve several sub simulations of components and usually these sub simulations are heterogeneous in nature. There is a possibility that simulator for some components may already be implemented by others. So re-implementing a system is expensive and risky. Complex simulation cannot be run by ordinary computer system. For this purpose, super

computers are used which is a very expensive option. Distributed systems are introduced to solve complex computational problems that by dividing them into many components. Each of them is solved by one or more computers which communicate with each other through message passing. HLA [1] presents an architecture for distributed modeling and simulation. The main purpose of this architecture is to make possible the reuse of

---

Authors E-Mail: (amirmuhammadakram@gmail.com, shahzad.sarfraz@nu.edu.pk, umar.shoaib@uog.edu.pk)

\* University of Gujrat, Gujrat, Pakistan.

simulation components. HLA simulations can interact with each other irrespective of the computing platform. A single computer in simulation system is known as federate and the composition of different interactive federates is called federation. HLA provides federation formalism by which federates can be modeled to support federation execution using HLA specification (IEEE 1516-2010, IEEE Standard for M&S (Modeling and Simulation) HLA - Framework and Rules) developed to provide reusability and interoperability for simulations [2-3].

RTI [4] provides a set of services that are used for communication between federates during the execution of simulation. RTI services are divided into six major categories: Federal Management, Declaration Management, Object Management, Ownership Management, Time Management and Data Distribution Management. RTI provides the portability benefits (across multiple computing platforms, operating systems, communication systems) and simulation interoperability. RTI is responsible for the communication (data exchange) between federates during execution [5].

There are three parts of the HLA. First is Interface Specification [6] that defines the process of how middleware [7] interacts with the HLA simulation RTI. Second part is HLA Rules [2], the simulation must follow the rules in order to meet the standard. Third and last part is Object Model Template [8] that specifies the rules for message exchange between simulations and defines what kind of information should be communicated [9]. FDK (Federated Development Kit) [10] is based on modules. These modules can be composed and used for different RTI's build. FDK is designed, so the developers can choose the most appropriate FDK module for the development of their RTI implementation [11].

There are many RTIs available in the market but their functionality and performance differs based on the structure and approach of RTIs. Structure of RTIs will be reviewed based on different parameters later in the article which can help to understand the differences of different RTIs. Moreover two different approaches/architecture SDRTI and the CeRTI will be analyzed for better understanding of RTIs. The purpose of this review is to simplify the selection of RTIs to a certain type of simulation based upon particular requirements. For example, if a simulation needs high speed or real time communication among components then it can be done based on provided review.

## 2. SIMULATION

System Simulation [12] means regenerating the behavior of the system over time. The purpose of software simulation in computation is to store the state of a system in computer memory and predict the results with respect to change in time. Without building the actual system, simulation can be used to produce different desired results. Simulation is an approach for analyzing different systems before deploying the actual systems. Especially, the systems for which mathematical models are more complex to solve. Consider, an example if efficient network design is required. Simulations can be used to evaluate different network features like performance and the flow of the network. Network Simulation can include necessary information about the network such as topology, latency and bandwidth characteristics. It also gives an accurate information about network status for a given network load over time [13].

### 2.1 Discrete Event Simulation

Usually a discrete event simulation pattern is used by network simulations. In discrete event simulation system

event occurs at discrete points in time and considered as change in state. To design efficient network example, events may correspond to sending or receiving a message or beginning the processing of a message. In continuous simulations, state changing events are viewed as occurring continuously over time. Each event has an event handler, which specifies how the state variables change on occurrence of that event. When a message is received, an event occurs so system state variables change their values [14].

## 2.2 Parallel and Distributed Discrete Event Simulation

Multiple processors are called in parallel for distributed discrete event simulations in discrete event simulation program. [15]. These types of simulations are executed on tightly coupled computer systems such as shared memory multiprocessors (clusters) in case of parallel discrete event simulations [16]. All processors connected with each other use high-speed switches. The advantage of using high-speed switches is low communication latencies. Loosely coupled computer systems are required in order to execute distributed discrete event simulations. Workstations, interconnected via LANs (Local Area Network) or WANs (Wide Area Network), are example of this type of system [17-18]. In loosely coupled systems, communication latencies are higher than the tightly coupled systems [19].

## 3. HIGH LEVEL ARCHITECTURE

The HLA has been developed by the United States Department of Defense as a standard for common architecture of distributed modeling and simulation [3]. HLA is an integrated technique to provide a common framework for the inter connection of different interacting

simulations. The basic purpose of HLA is to provide a common simulation infrastructure that can support interoperability and reusability for simulations. HLA is used in a number of simulation application areas such as education and training, engineering and analysis.

The HLA consists of three parts

- HLA Rules
- OMT (Object Model Template)
- Interface Specification

There are ten HLA Rules the must be followed by simulation entities (federates) in order to attain proper interaction during a simulation. The HLA rules are categorized into two groups; first five rules for federations and the next five rules for federate are given. The federation rules are for creation of federation, including documentation requirements, data interchange, objects representation, interfacing requirements and attribute ownership. The federate rules are defined for a single federate, which include documented HLA SOM (Simulation Object Model), local time management; SOM attributes values update, acceptance and transfer of attributes ownership [20].

OMT has defined asset of common objects used by a federation, attributes of the objects and relationships among different attributes. OMT also describes MOM (Management Object Model), FOM (Federation Object Model) and SOM [8]. FOM defines all the objects, interaction of objects and data encoding schemes. SOM presents object and interactions for federates that are used externally. Federates internal operation are also focused by the SOM. MOM identifies objects and interactions that are used to handle a federation [21].

The interface specification describes standards for RTI. It provides a generic communication interface for RTI. Interface Specification ties with federate during model execution [6].

#### 4. RUN TIME INFRASTRUCTURE

A collection of libraries that provide services to the simulation systems is called RTI. Computer simulations can interact with each other regardless of their computing platforms by using RTI. The design of RTI is scalable and extensible. There are six services provided by RTI; Federation Management, Object Management, Declaration Management, Ownership Management, Time Management, and Data Distribution Management [20].

Federation Management is used for the creation and the deletion of federation executions, joining and resigning federates, control checkpoints, resume, pause and restart the federations [22-23]. Ownership Management deals with exchange of ownership between the object attributes [24-25]. Declaration Management is for subscribing and publishing object attributes and interactions [26]. Object Management is responsible for sending object attribute updates and interactions, creation and deletion of object instances and object reflections [9]. The Time Management service is used to coordinate the advance simulation time and relationship between the real time and the advance simulation time [27]. Data Distribution Management service is responsible to rout data efficiently between federates or federations [28]. To increase the scalability, common techniques are used including hierarchies and reduction networks that are standardized with the architecture by components. The RTI architecture is able to adopt the changes and ready to adopt changing requirements and technologies [29].

The RTI is system software that provides interaction of federates using a common FOM. Federate actions are defined as HLA events; these include registering an object, sending an interaction or updating attributes of an object. RTI Exec (Executive Process) manages the creation and destruction of Federation executions in a simulation. During simulation a federate uses RTI Amb (Ambassador) methods and the FedAmb (Federate Ambassador) for communication as shown in Fig. 1. RTI Ambassador and Federate Ambassador owned by each federate [9].

#### 4.1 RTI Implementations

RTI is a basic part of HLA that provides runtime services to federate. It provides portability benefits across operating systems, multiple computing platforms and communication systems. For reusability and open source RTI implementations are available. Functionality and performance of different RTIs is different in different scenarios. Different RTI implementations, their release years, standard use by them, binding and developers are shown in Table 1.

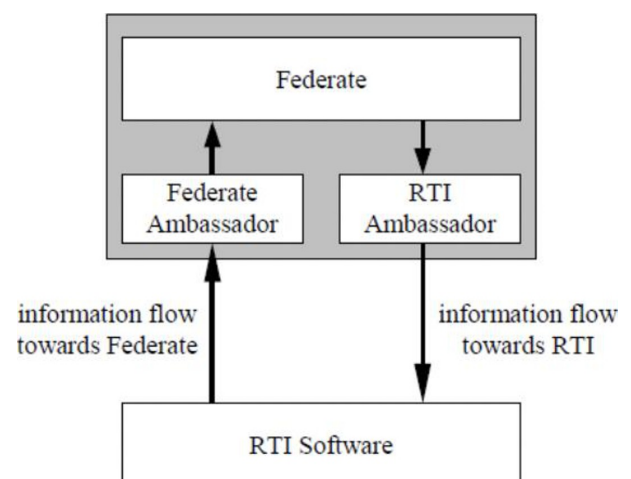


FIG. 1. RTI AMBASSADOR PARADIGM

FFDK [11] is a HLA based RTI software that is developed at Georgia and taught by the PADS research group led by Professor Richard Fujimoto. Its modules can be composed which are used to build run time infrastructure, by using it integration of different simulations could be made possible [30]. The FDK design allows RTI developers to take the FDK modules and choose from the set of these FDK modules that are most suitable for developing their particular implementation of RTI. RTI-kit is a main component of FDK, which is a collection of libraries [31]. Detailed RTI (DRTI is a sample implementation of the RTI based on the RTI-Kit [32].

The RTI prototype was developed in the 1990's by MIT (Massachusetts of Technology) Lincoln laboratories. All the implementations done in that period was a proof to the concept for HLA. During this period, HLA was

under development so all HLA Services [35,37-38] could not be implemented by developers of RTI prototype [33]. Time management service was not included in RTI Prototype.

RTI 1.3 is also developed by RTI developers group at Lincoln labs. This version RTI 1.3 is successor of RTI prototype and name RTI 1.3 is because of the implementation of the version 1.3 of HLA specification. The previous versions are no longer supported. 1.3 was partially released and did not support all the functionality of HLA [34]. RTI 1.3NG was developed by SAIC, (Science Applications International Corporation) and it was sponsored by DMSO (Defense Modeling and Simulation Office), which is based on competitive industry designs. DMSO RTI 1.3NG supports HLA specification 1.3 [35].

TABLE 1. RTI IMPLEMENTATIONS

RTI Implementation	Release Year	Binding	Standard	Developers
RTI 1.3	In 1996	C/C++	HLA 1.3	MIT Lincoln Labs
DMSO RTI	In 1996	C/C++	HLA 1.3	DoD
GMU RTI	In 1997	C++	HLA 1.1	George Mason University
FDK-DRTI	In 2001	C++	IEEE 1516	Georgia Tech
HPC RTI	In 2002	C++	IEEE 1516	RAM Labs
MÄK RTI	In 2002	C/C++, Java	HLA1.3, IEEE 1516-2000, IEEE 1516-2010 (HLA Evolved)	MÄK Technologies
XRTI	In 2003	Java	IEEE 1516	Open Source
RTI-NG 4.0.4	In 2006	C++, Java	HLA1.3, IEEE 1516-2000, IEEE 1516-2010 (HLA-Evolved)	SAIC & VTC
pRTI 3.2.2	In 2007	C++, Java, Web Services	HLA 1.3, IEEE 1516-2000, IEEE 1516-2010 (HLA Evolved)	Pitch AB
Portico RTI 1.0.2	In 2010	C++, Java	HLA 1.3, IEEE 1516, IEEE-1516e	Open Source
CERTI 3.4.0	In 2011	C++, Fortran90, Java, Matlab, Python	HLA 1.3 partial, IEEE 1516partial	Open Source
RTI-s	In 2012	C++, Java, Ada	HLA 1.3 partial, 1516e partial	US Government

MAK RTI was developed by MAK Technologies. “MAK Technologies” based in Massachusetts and in Cambridge [36]. “MAK Technologies” is a popular provider for network simulation software. It also does not implement all HLA services. The GMU (George Mason University) light-weight RTI [39] was the project of GMU. The focus of GMU light-weight RTI is on Data Distribution Management and Declaration Management. In GMU light-weight RTI two services of HLA were not implemented. One is Time Management and other is Ownership Management. GMU light-weight RTI is best for real time simulation in small and medium size federation [40].

Pitch Corporation in Linköping, Sweden developed the Pitch portable RTI [41]. Implementation of all services of HLA Interface Specification version 1.3 is available in pRT [42]. CERTI is a RTI for distributed discrete event simulation developed at ONERA [43]. CERTI is recognizable for communicating processes that is original architecture. CERTI is distributed system that has two processes one is local (RTIA) and the other is RTIG (Global) also has a libRTI (library), each federate is linked by libRTI. Each federate has local interaction through a Unix-domain socket with an RTIA (Ambassador) [43]. On network, various distributed algorithms can be run. RTI services, RTIA exchanges messages with RTIG process via TCP sockets [44].

Portico RTI is an open source and cross-platform RTI that is fully supported HLA implementation [45]. Portico RTI has following components: LRC (Local Runtime Component), Connection Binding and RTI. The basic communication mechanism is isolated and hidden from the actual functionality of the RTI [46]. HLA has been

gaining the popularity in the United States Department of Defense SPEEDES [47] so it is improved to serve as RTI. However, RTI selection should be according to its compatibility and specification of HLA. This improvement results in the HPC-RTI [48]. HPC-RTI was the first effort to modify existing simulation engine into HLA RTI characteristic of HPC-RTI from others, it contains the implementation of Time Management Service [49]. HPC-RTI provides a time management service to all HLA services so that HLA does not need the data distribution management, ownership management or declaration management services to be time-managed [50].

## 5. COMPARISON

HLA implementations (RTI) support interoperability and reusability as their basic purpose [51-52]. Currently available implementations of RTI respond differently to HLA exercises. All RTIs perform same task that is acting like a middleware during simulation. However, all have some distinct features, for example, some perform best in small-scale simulation and some in large-scale simulations. Some have specific best implementation of one or two RTI services, which makes them unique. Selection of a RTI is strictly based upon your simulation nature. If simulation demands faster communication between federates you probably select the RTI with the best implementation of data distribution management service.

RTIs can be compared with one another on the basis of their basic features like HLA compliance, presence of RTI Executive process and the transportation mode of data that has been used to interact with federates [53]. HLA compliance means that the RTI implementation is

Compliant with the HLA evolved standards. With respect to RTI’s functionality, many aspects required some centralized knowledge. To manage this centralized knowledge there are two strategies. One is to make RTI LRC (Local Component) that is responsible for one federate and the other is to make a central server application known as RTI Executive or rtiexec. Transportation mode can be the best effort, reliable or both. In Best effort transport mode federates data obtains unspecified variable bit rate and delivery time depending on the current traffic load, but the reliable transport mode built on top of best-effort possibly without latency.

The comparison of RTI implementations based on the features they provided shown in Table 2. The performance of federates in a simulation also depends upon many factors like type of network, hardware platform, operating system and the chosen RTI’s capability.

### 5.1 Service-Distributed Run Time Infrastructure vs. Central RTI

The structure of a RTI can be central, distributed or hierarchical. Most RTIs are based on central structure.

Services are provided through an interface as a special programming language binding and platform binding as shown in Fig. 2. Therefore, these RTIs have some drawbacks. RTI is dependent on specific platform and on special programming language. In central structure, RTI can face bottleneck because of increase in simulation data and processes on central server that can cause simulation to h. This load balancing deficiency creates heavy load on central server as simulation nodes increases. Most RTIs use static linking of federates that decreases the portability of federates in RTI. RTIs only support distributed applications on local area network or a special network [54].

To overcome the issues a SDRTI is purposed in 2013 [55], which aims to provide many excellent characteristics such as load balancing, reusability, heterogeneous communication, high security, loosely coupled services, wide area network, and efficient and stable network communication speed.

TABLE 2: COMPARISONS OF RTIS

RTI	HLA Compliant	RtiExec Required	Transport Mode	Other Properties	Reference
MSO RTI NG	Yes	Yes	Best Effort and Reliable	Most Stable RTI among all tested	Fujimoto [33]
FDK Detailed RTI	No	No	Reliable	Second Most Stable RTI, Objects cannot be Destroyed, when environment Variables are set easiest to run	SAICTR Group [34]
MAK RT RTI	No	Yes	Best Effort and Static Reliable	Processes Manual Cleanup Required, Sensitive to Frequency of Tick	MÄK RTI [36]
GMU light-weight RTI	Yes	Yes	Reliable	Best for small simulations	GMU Light-Weight RTI [39]
Pitch RTI	Yes	Yes	Best Effort and Reliable	Least Stable, Threaded RTI	Pitch Portable RTI [41]
CERTI	No	Yes	Best Effort and Reliable	Support High Performance Distributed Simulation, Multi-resolution	Noulard et. al. [44]
Portico RTI	No	Yes	Best Effort and Reliable	Widely-used services of HLA evolved	Web Site [45]
HPC-RTI	Yes	Yes	Best Effort and Reliable	Implementation of time management	Steinman [49]

The main advantage of SDRTI is that RTI services are decoupled and distributed in different nodes on internet is shown in Fig. 3. The other advantage

is that it overcomes the drawbacks of central RTI and provides effective load balancing solution on the internet.

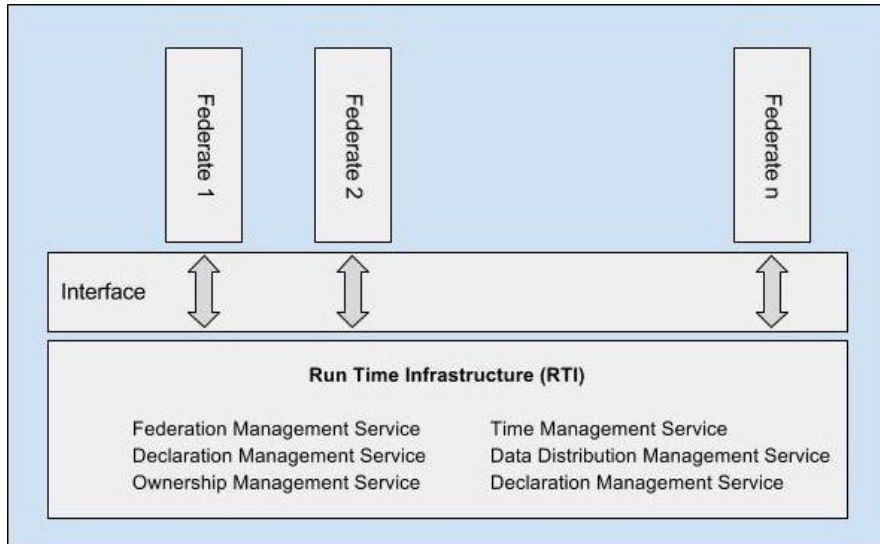


FIG. 2. CENTRAL RTI ARCHITECTURE

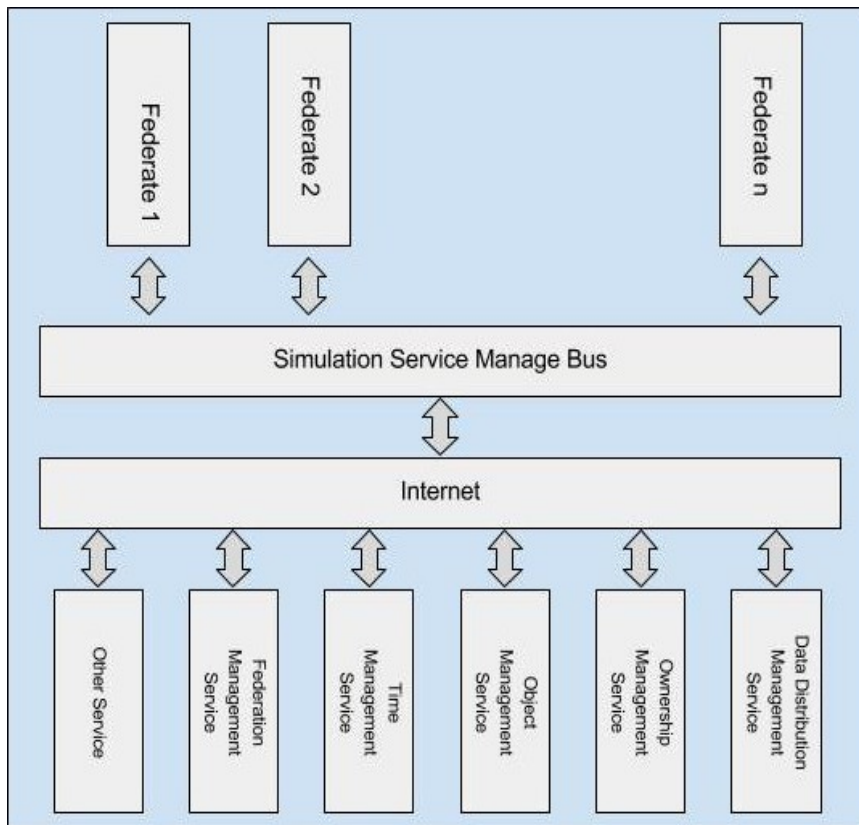


FIG. 3. SDRTI ARCHITECTURE



## 6. CONCLUSION

All HLA implementations provide interoperability and reuse facility, as it is a basic purpose of HLA. Each implementation has its own pros and cons. As RTI provides services to federates in runtime execution so performance and stability of federates depends upon the selection of RTI. Different implementations of HLA RTI are available in the market as open source and closed source. However, RTI selection will be based on the need of the simulation. For example, to configure GUI (Graphical User Interface) easily, tracking MAK RTI can be used. Some RTIs show better performance for large-scale simulations, some for small simulations. The three RTI implementations have similar behavior for small simulations. MSO RTI NG are declared as most stable, FDK Detailed RTI is second stable and Pitch RTI is least stable among all RTIs tested.

RTIs still have deficiencies including load balancing, reusability, heterogeneous communication, high security, loosely coupled services, WAN, and efficient and stable network communication speed. To deploy RTI simulation services on the Internet a SDRTI is introduced which helps in cost reduction, load balancing, reusability and security aims to overcome RTI deficiencies. There is a need to develop a generic module based RTI which has the capability to overcome all the existing deficiencies in current available RTIs. Another solution is to combine the available ones using hybrid approach to get maximum benefits.

## ACKNOWLEDGEMENT

Authors would like to express very great appreciation to Dr. Muddesar Iqbal, Kingston University, UK and Komal Intizar University of Gujrat, Gujrat, Pakistan for their valuable and constructive suggestions during the planning and development of this research work. Their willingness to give time so generously has been very much appreciated.

## REFERENCES

- [1] Dahmann, J.S., and Morse, K.L., "High Level Architecture for Simulation: An Update: In Distributed Interactive Simulation and Real-Time Applications", Proceedings. IEEE 2nd International Workshop on Distributed Interactive Simulation and Real-Time Applications, pp. 32-40, Canada, 19 July, 1998.
- [2] IEEE-Standards Association, "1516-2000 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules", 2010 <http://standards.ieee.org/> (Accessed date: March 2018).
- [3] Dahmann, J.S., Fujimoto, R.M., and Weatherly, R.M., "The Department of Defense High Level Architecture", Proceedings of IEEE 29th Conference on Winter Simulation, Computer Society, pp. 142-149, December 1997.
- [4] Fujimoto, R., McLean, T., Perumalla, K., and Tacic, I., "Design of High Performance RTI Software: In Distributed Simulation and Real-Time Applications", Proceeding of 4<sup>th</sup> IEEE International Workshop on Distributed Simulation and Real-Time Applications, pp. 89-96, 2000.
- [5] Bachinsky, S., Mellon, L., Tarbox, G., and Fujimoto, R., "RTI 2.0 Architecture", Proceedings of Spring Simulation Interoperability Workshop, March, 1998.
- [6] Sung, C.H., Hong, J.H., and Kim, T.G., "Interoperation of DEVS Models and Differential Equation Models Using HLA/RTI: Hybrid Simulation of Engineering and Engagement Level Models", Proceedings of Spring Simulation Multi Conference, Society for Computer Simulation International, 2009.
- [7] Emmerich, W., "Software Engineering and Middleware: A Roadmap", Proceedings of ACM Conference on the Future of Software Engineering, pp. 117-129, May, 2000.
- [8] Fujimoto, R.M., "Parallel Simulation: Distributed Simulation Systems", Proceedings of 35th Conference on Winter Simulation: Driving Innovation. Winter Simulation, pp. 124-134, December, 2003.

- [9] Rocha, R.V., Araújo, R.B., Campos, M.R., and Boukerche, A., "HLA Compliant Training Simulations Creation Tool", Proceedings of 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, Computer Society. pp. 192-198, October, 2009.
- [10] Riley, G.F., Ammar, M.H., Fujimoto, R.M., Park, A., Perumalla, K., and Xu, D., "A Federated Approach to Distributed Network Simulation", ACM Transactions on Modeling and Computer Simulation, Volume 14, No. 2, pp. 116-14, April, 2004.
- [11] FDK-Federated Simulations Development Kit: [www.cc.gatech.edu/computing/pads/fdk/](http://www.cc.gatech.edu/computing/pads/fdk/), (Accessed date: 12 March 2018).
- [12] Harrison, J.R., "The Concept of Simulation in Organizational Research", Journal of Richard Harrison, School of Management, University of Texas at Dallas, 1999.
- [13] Fujimoto, R.M., "Parallel and Distributed Simulation Systems", Proceeding of Winter Simulation Conference Volume 300, No. 1. pp. 147-157, USA, January, 2000.
- [14] Banks, J., John, S., Carson, I.I., Barry, L.N., and David, M.N., "Discrete-Event System Simulation", 5th Edition, 2013.
- [15] Ferscha, A., and Tripathi, S.K., "Parallel and Distributed Simulation of Discrete Event Systems", October, 1998.
- [16] Fujimoto, R.M., "Parallel Discrete Event Simulation", Communications of the ACM, Volume 33, No. 10, pp. 30-53, October, 1998.
- [17] Bagrodia, R., Chandy, K.M., and Liao, W.T., "A Unifying Framework for Distributed Simulation", ACM Transactions on Modeling and Computer Simulation, Volume 1, Volume 4, pp. 348-385, 1991.
- [18] Bizarro, P., Silva, L.M., and Silva, J.G., "JWarp: A Java Library for Parallel Discrete-Event Simulations", Concurrency - Practice and Experience, Volume 10, No. 11-13, pp. 999-1005, 1998.
- [19] Bagrodia, R., Meyer, R., Takai, M., Chen, Y.A., Zeng, X., Martin, J., and Song, H.Y., "Parsec: A Parallel Simulation Environment for Complex Systems", Computer, Volume 31, No.10, pp. 77-85, October, 1998.
- [20] Wang, F., Turner, S.J., and Wang, L., "Agent Communication in Distributed Simulations" International Workshop on Multi-Agent Systems and Agent-Based Simulation, pp. 11-24, Springer, Berlin Germany, July, 2004.
- [21] Lutz, R.R., "Migrating the HLA Object Model Template to an IEEE Standard", Johns Hopkins APL Technical Digest, Volume 21, No. 3, pp. 337-347. 2000.
- [22] Yuan, Z., Cai, W., Low, M.Y.H., and Turner, S.J., "Federate Migration in HLA-Based Simulation", International Conference on Computational Science, pp. 856-864, Springer, Berlin Germany, June, 2004.
- [23] Boukerche, A., and De Grande, R.E., "Optimized Federate Migration for Large-Scale HLA-Based Simulations: In Distributed Simulation and Real-Time Applications", 12th IEEE/ACM International Symposium, pp. 227-235, October, 2008.
- [24] Chen, L.F., Gu, H., Cao, Z.M., and Li, H.Y., "Practical Methods on Data Distribution Management of Large Complex System, Acta Simulata Systematica Sinica, Volume 5, pp. 15, 2004.
- [25] Moradi, F., Ayani, R., and Tan, G., "Some Ownership Management Issues in Distributed Simulation Using HLA/RTI", Parallel and Distributed Computing Practices, Volume 4, No. 1, 2000.
- [26] Joshi, R., and Castellote, G., "A Comparison and Mapping of Data Distribution Service and High-Level Architecture", Real-Time Innovations Inc. August 2010.
- [27] Fujimoto, R.M., "Time Management in the High Level Architecture", Simulation, Volume 71, No. 6, pp. 388-400. 1998.
- [28] Pan, K., Turner, S.J., Cai, W., and Li, Z., "Implementation of Data Distribution Management Services in a Service Oriented HLA RTI", Winter Simulation Conference, pp. 1027-1038, December, 2009.

- [29] Gupta, P., "Resource-Constraint and Scalable Data Distribution Management for High Level Architecture" in School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, USA, 2007.
- [30] Fullford, D., and Wetzel, D., "A Federation Management Tool: Using the Management Object Model (MOM) to Manage, Monitor, and Control an HLA Federation", Spring Simulation Interoperability Workshop, 1999.
- [31] Riley, G.F., Mostafa, H.A., Richard, M.F., Alfred, P., Kalyan, P., and Xu, D., "A Federated Approach to Distributed Network Simulation", ACM Transactions on Modeling and Computer Simulation, Volume 14, No. 2, pp. 116-148, April, 2004.
- [32] Fujimoto, R., McLean, T., Perumalla, K., and Tacic, I., "Design of High Performance RTI Software: In Distributed Simulation and Real-Time Applications", 4<sup>th</sup> IEEE International Workshop on Distributed Simulation and Real-Time Applications, pp. 89-96, 2000.
- [33] Siron, P., "Design and Implementation of a HLA RTI Prototype at ONERA", Fall Simulation Interoperability Workshop, 98F-SIW-036, September, 1998.
- [34] SAICTR Group, "High Level Architecture Run-Time Infrastructure RTI 1.3-Next Generation Programmer's Guide Version 5." 1999.
- [35] Defense Modeling and Simulation Office, <https://www.msco.mil/> (Accessed date: March 2018).
- [36] MÄK RTI: <https://www.mak.com/products/link/mak-rti> (Accessed date: March 2018).
- [37] MÄK RTI 3.1.2 Release Notes, MÄK Technologies, 2007
- [38] Watrous, B., and Granowetter, L., and Wood, D., "The MÄK High-Performance RTI: Performance by Design", MÄK Technologies. 2009.
- [39] GMU Light-Weight RTI: <http://netlab.gmu.edu/rti/> (Accessed date: March 2018).
- [40] Pullen, J.M., and Kakarlamudi, N., "Performance Issues for the Light-Weight RTI", Proceedings of IEEE 3<sup>rd</sup> Workshop on Simulation Interoperability, Orlando September, 1998.
- [41] Pitch Portable RTI (pRTI): <http://www.pitch.se/products/prti/> (Accessed date: March 2018).
- [42] Knight, P., Corder, A., Liedel, R., Giddens, J., Drake, R., Jenkins, C., and Agarwal, P., "Evaluation of Run Time Infrastructure (RTI) Implementations", Huntsville Simulation Conference, Volume 18, 2002.
- [43] Bréholée, B., and Siron, P., "CERTI: Evolutions of the Onera RTI Prototype", Fall Simulation Interoperability Workshop, September, 2002.
- [44] Noulard, E, Rousselot, J.Y., and Siron, P., "CERTI, An Open Source RTI, Why and How", Spring Simulation Interoperability Workshop, pp. 23-27, 2009.
- [45] The Portico Project Web Site, <http://www.porticoproject.org> (Accessed date: March 2018).
- [46] Tu, Z., Zacharewicz, G., and Chen, D., "Developing a Web-Enabled HLA Federate Based on Portico RTI", Proceedings of IEEE Simulation Conference on Winter Simulation Conference, pp. 2289-230, 2011.
- [47] Steinman, J.S., "A Multiple-Synchronization Environment for Parallel Discrete-Event Simulation", International Journal in Computer Simulation, USA, January, 1992.
- [48] Gary, B., Scott, S., and Jeff, W., "Design and Implementation of the HPC-RTI for the High Level Architecture in SPEEDES 0.81", Proceedings of Spring Simulation Interoperability Workshop, Paper 00S-SIW-153, 2000.
- [49] Steinman, J.S., "WarpIV Kernel: Real Time HPC-RTI Prototype", Proceedings of Spring Simulation Interoperability Workshop, 2005.
- [50] Joe, C., Sebastian, C., Chris, B., and Larry, P., "The Development of an HLA Compliant High Performance Computing Run-Time Infrastructure", Proceedings of Spring Simulation Interoperability Workshop, Paper 02S-SIW-016, 2002.

- [51] Topçu, O., Umut, D., Halit, O., and Levent, Y., “High Level Architecture”, *Distributed Simulation*, pp. 39-66, Springer International Publishing, 2016.
- [52] Bouanan, Y., Gorecki, S., Ribault, J., Zacharewicz, G., and Perry, N., “Including in HLA Federation Functional Mockup Units for Supporting Interoperability and Reusability in Distributed Simulation”, *Proceedings of 50th Computer Simulation Conference*, 2018.
- [53] Malinga, L., and Le, R.W.H., “HLA RTI Performance Evaluation”, *Proceedings of European Simulation Interoperability Workshop on Society for Modeling & Simulation International*, pp. 39-44, 2009.
- [54] Brutzman, D., Zyda, M., Pullen, J.M., Morse, K.L., Fouskarinis, S., Drake, D., Moen, D., Blais, C., Kapolka, A., and McGregor, D., “Extensible Modeling and Simulation Framework (XMSF) Challenges for Web-Based Modeling and Simulation”, *Findings and Recommendations Report of the XMSF Technical Challenges Workshop and Strategic Opportunities Symposium*, 2002.
- [55] Wang, Z., Zhang, H., Zhang, R., Li, Y., and Xu, B., “A Run-Time Infrastructure Based on Service-Distributed Architecture”, *Applied Mathematics & information Sciences*, pp. 595-604, June, 2013.