

Implementing Data Distribution Management System Using Layer Partition-Based Matching Algorithm

Nwe Nwe Myint Thein

Abstract—Simulation has become a popular tool to study a broad range of systems. The growing number and quality of simulation software requires expertise for their evaluation. In High Level Architecture paradigm, the Runtime Infrastructure (RTI) provides a set of services. Data Distribution Management (DDM) service reduces message traffic over the network. DDM services are used to reduce the transmission and receiving of irrelevant data and aimed at reducing the communication over the network. Currently, there are several main DDM filtering algorithms. The paper describes practically testing simulation for the data distribution management (DDM) service of the Battleground simulation with the dynamic fighters by using layer partition-based algorithm. The layer partition-based matching algorithm is based on divide and conquers approach. It selects the dynamic pivot by detecting the regions distribution on the routing space. This system intends to detect the movement of the fighter objects, searches overlap between the fighter object and every battalions (extents). It is large-scale distributed simulation in order to minimize subsequent computations and algorithm complexity. The developed system can be used not only in the research purpose but also in the real-world distributed applications. It provides the low computational time and exact matching result.

Index Terms—Data distribution management, matching algorithm, layer partition-based matching algorithm, modelling and simulation.

I. INTRODUCTION

HLA (High Level Architecture) is a software architecture which can be reusable for execution of distributed simulation applications. HLA consists of rules, interface specification and object model template. The interactions between federation and federates are governed by an interface specification with the Runtime Infrastructure. Object Model Template (OMT) is a role to document major information about simulations.

HLA presents a framework for modeling and simulation within the Department of Defense (DoD). The goal of this architecture is to interoperate multiple simulations and facilitate the reuse of simulation components. HLA allows interconnection of simulations, devices, and human operators in a common federation. It builds on composability, letting designer construct simulations from pre-built components. Each computer-based simulation system is called a federate and the group of interoperating systems is called a federation. HLA specifications—incorporated as IEEE 1516 standard—were developed to provide reusability and

interoperability.

HLA was developed by the Defense Modeling and Simulation Office (DMSO) of the Department of Defense to meet the needs of defense-related projects, but it is now increasingly being used in other application areas. The Department of Defense's policy is to disseminate information about the HLA as widely as possible, both inside and outside the US, and even to provide free supporting software to help new users to evaluate and use the HLA as easily and inexpensively as possible [1], [2].

A hierarchy of components of increasing levels of aggregation is considered as a complex simulation. The lowest level is the model of a system component. This may be a mathematical model, a discrete-event queuing model, a rule-based model etc. The model is implemented in software to produce a simulation. When this simulation is implemented as part of an HLA-compliant simulation, it is referred to as a federate. HLA simulations are made up of several HLA federates and are called federations. There can be multiple instances of a federate, for example several Boeing 747 simulations or F-16 simulations, and the instances' number can change as the simulation continues. The simulations that use HLA are modular in nature allowing federate to join and resign from the federation as the simulation executes. Previous and recent developments of distributed simulation systems have been carried out mainly in the areas of military applications such as commercial war games. In war gaming, a tank may regularly publish its position and another object e.g. an aircraft may be interested to detect the position of the tank by subscribing to receive the position update data published by the tank [3], [4].

Efficient data distribution is an important issue in large scale distributed simulations with several thousands of entities. The broadcasting mechanism employed in Distributed Interactive Simulation (DIS) standards generates unnecessary network traffic and is not suitable for large scale and dynamic simulations. An efficient data distribution mechanism should filter the data and forwards only those data to federate who need the data. Several filtering mechanisms have appeared in the literature and some of them have been implemented in RTI [5], [6].

The goal of the DDM module in RTI is to make the data communication more efficient by sending the data only to those federates who need the data, as opposed to the broadcasting mechanism employed by Distributed Interactive Simulation (DIS). The approaches used by DDM are aimed at reducing the communication over the network, and the data set required to be processed by the receiving federates [7].

In the implementation of distributed interactive simulation,

it requires the specification of routing space and the number of update regions and subscribes regions. This implementation needs to search actual overlap information between the update and subscribe regions. The overlapping result also needs to be known in a time saving manner for the participants of the distributed simulations [8].

Modeling and simulations have always been a major part of human history. Modeling can be defined by “the process of producing a model; a model is a representation of the construction and working of some system of interest” and simulation can be defined as “A simulation of a system is the operation of a model of the system” simply. Modeling and Simulation(M&S) is used to simulate real system’s objectives by modeling components, simulation steps and process and implementing produced models in a time flow. The area of M&S was extended from War Game to Task Request, Weapon Acquisition, Decision, Analysis and Military Training. Also, efficient operation of massive simulation and interoperability between complex systems has been studied. [9] The remainder of the paper organizes as follows. Section 2 describes the theory background of DDM and matching algorithms. The layer partition-based matching algorithm explains in section 3. Section 4 represents the simulation model by using layer partition-based matching algorithm. Finally, section 5 offers conclusion.

II. LITERATURE REVIEW

Data Distribution Management (DDM) is a set of services defined in HLA to distribute information in distributed simulation environments. HLA’s Run Time Infrastructure (RTI) is a software component that provides commonly required services to simulation systems. There are several groups of services, which are provided by RTI to coordinate the operations and the exchanges of data between federates (simulations) during a runtime execution. The interaction of object instances supports by the function of RTI, which is like a distributed operating system. The evolution of the DDM service provides solutions to problems by using filtering mechanism that is suitable for large-scale simulation. These services rely on the computation of the intersection between “update” and “subscription” regions. When calculating the intersection between update regions and subscription regions, the higher computation overhead can occur. Therefore, many DDM filtering algorithms are proposed [10].

A. Region-Based Algorithm

The region-based algorithm checks all the pairs of regions until an intersection found for each pair of update regions and subscription regions or the end of the regions list reached. The implementation of this algorithm is straightforward, but the performance is varying greatly. If there is N update regions and M subscription regions. “There are N*M pairs to check in the worst case” [11]-[13].

B. Grid-based Algorithm

In the grid-based approach, the routing space partition into a grid of cells. Each region mapped onto these cells. If a subscription region and an update region intersect with the

same grid cell, they assumed to overlap with each other. Although the overlapping information is not exact, the grid-based algorithm can reduce the computation complexity than the region-based algorithm. The amount of irrelevant data communicated in the grid-based filtering depends on the grid cell size, but it is hard to define the appropriate size of grid cells [12]-[14].

C. Hybrid Approach

The hybrid approach is an improvement approach over the region-based and the grid-based approaches. The matching cost is lower than the region-based approach, and this advantage is more apparent if the update frequency is high. It also produces a lower number of irrelevant messages than that of the grid-based approach using large cell sizes. The major problem is that it has the same drawbacks as the grid-based approach: the size of the grid cell is very crucial to the behavior of the algorithm [14], [15].

D. Sort-Based Algorithm

The sort-based algorithm used a sorting algorithm to compute the intersection between update and subscription regions. The algorithm’s performance degraded when the regions are highly overlapped, and it needed to optimize the sorting data structure for the efficient matching operation [16].

E. Binary Partition-Based Algorithm

The binary partition-based algorithm takes a divide-and-conquer approach. It performs binary partitioning which divides the regions into two partitions that entirely cover those regions according to the midpoint (pivot value) of the routing space. It matches and compares the pivot partition with the left and right partitions. The overall overlap information can be obtained by combining the overlap information of each dimension. The selection of the pivot partition is a determinant factor of the performance of the algorithm. The overlap information of all region is obtained by the pivot partition. If there is no overlapping at all in these partitions, the comparing process of the pivot partition with the left and right partitions is the most time-consuming operation. With the greater overlap rate, the binary partition-based algorithm performs well. The partitioning process of the algorithm are stated in Fig. 1 [17], [18].

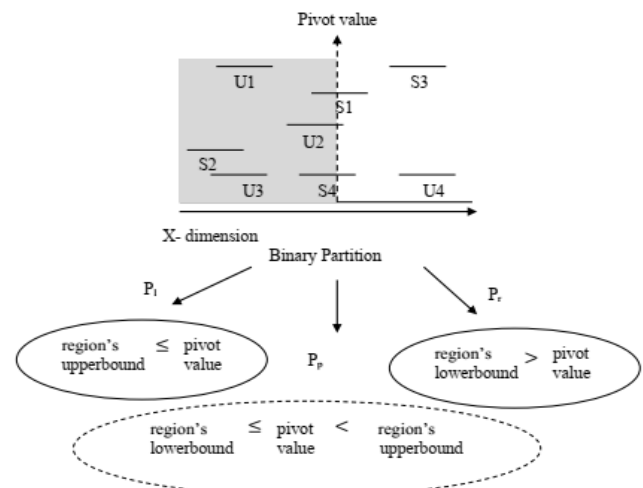


Fig. 1. Partitioning process of binary partition-based algorithm.

III. LAYER PARTITION-BASED ALGORITHM

The Layer partition-based matching algorithm supports to search the overlapping information for data distribution management of HLA. It executes in dimension by dimension. This algorithm accepts all regions in the routing space. It also generates all regions randomly. Then it sends these regions to the Layer partition-based matching algorithm. The LPM algorithm firstly chooses the optimal pivot to define the matching area. The efficiency and performance of the divide and conquer approaches depends on the choice of the pivot value. Some algorithms choose the middle point as the pivot value. The LPM algorithm accepts the projected regions list and select one point of that list as the pivot value. At that point, the most subscriber regions and updater regions are converged in the projected regions list [19].

To define the exact matching area, a region distribution detection algorithm mainly used in the first layer of layer partition-based matching algorithm. The LPM algorithm firstly calculates the regions distribution. Then, the partitioning among regions performs based on the result of choosing pivot based on region detection and defines the matching area that entirely covers all regions, which need to match with regions at pivot point. The process of optimal pivot choosing and defining the exact matching area is shown in Fig. 2. The algorithm guarantees low computational overheads for matching process and reduces the irrelevant message among federates [20].

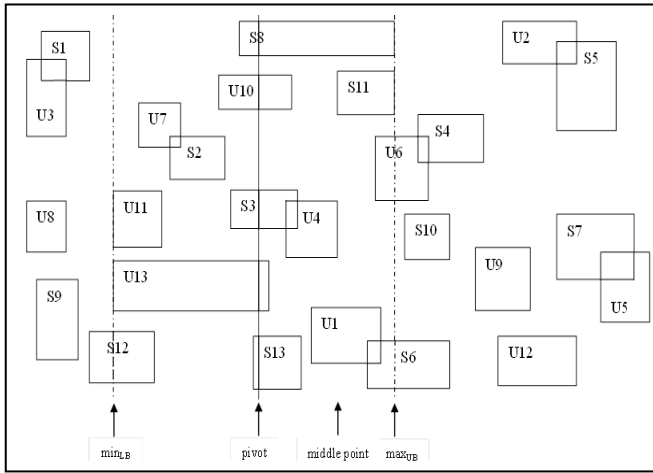


Fig. 2. Sample routing space.

The LPM algorithm promises the lower number of pivot point choosing. It also reduces the number of matching process between the updater regions and the subscriber regions of the routing space. The LPM algorithm reduces the half of matching process by defining the exact matching area. It also assures the lower number of pivots choosing for partition the routing space.

In the second layer, the specific decision of the region's selection performs to calculate the matching data between the three sets. This layer also supports the subtracted region lists. These lists subtract from the input regions set for next matching calculation. The classification of regions carries out in the region classifier algorithm. The actual matching between the updater regions and the subscriber region execute in intersection calculation algorithm. The subtracted

region lists use to reduce the next calculation. The final overlapping information produces by observing the result of two matrixes for two-dimensional routing spaces. The LPM algorithm complete when all dimensions are covered. [20] The process of layer partition-based matching algorithm is shown in the Fig. 3.

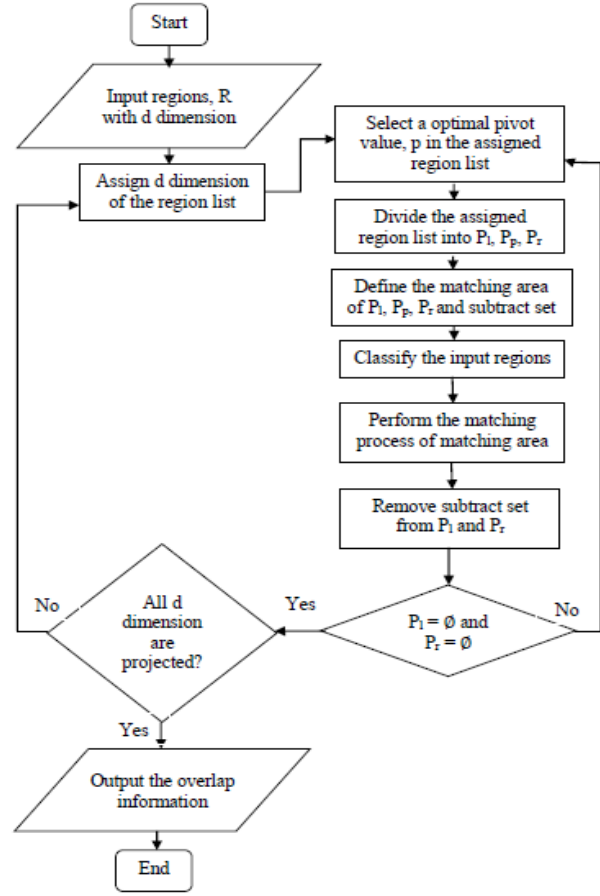


Fig. 3. Layer partition-based algorithm.

For the matching algorithms of DDM, the impact of network speed on the algorithm does not care and, there are no messages transferred in the network in all the approaches. Thus, a single computer used to make experiments. One of the important experimental parameters is the number of regions. The overlap rate defines as the proportion of the scene volume occupied by the regions. Therefore, we define the overlap rate as shown in equation (1):

$$\text{overlap rate} = \frac{\sum \text{area of regions}}{\text{area of space}} \quad (1)$$

where $\sum \text{area of regions} = \text{number of regions} \times \text{high of region} \times \text{width of region}$.

The performance of the LPM algorithm analyze on same size regions and different size regions, which generate randomly. The execution time of same size regions can be reduced about two third than the other matching algorithms for the overlapping rate is 1. LPM algorithm is more efficient than the existing matching algorithms of DDM at any overlapping degree using different size regions. The main advantage of this algorithm is its support for scalability very well, when the overlapping rate is large. The overlap rate defines as the proportion of the scene volume occupied by the regions.

The computational complexity of the LPM algorithm, we

suppose that there are N regions with the number of dimensions, $d=2$ in the multidimensional space. The optimal pivot algorithm requires $O(N)$ computation for the size of region is M . The first layer partition algorithm requires $O(N)$ computation. The total number of recursions for matching algorithm requires $O(\log N)$ computation.

Moreover, the second layer partition algorithm also needs $O(n)$ computation and the matching process of comparing the intersection of regions between partitions requires $O(n^2)$ computation (where n is the number of regions in each partition). The complexity of the intersection calculation procedure is proportional to n . It seems that the most important points are the exact matching partitions. It is obvious that the number of regions, n , is a determinant factor. Because the overlap information of all regions obtains by the pivot partition, it is not necessary to compare their overlap information in the left and right partitions of pivot partition. Therefore, the computational complexity of the LPM algorithm is $n^2 \times N \times O(\log N)$ computation. If the number of regions, n , is normally very small in a large-scale spatial environment, so the LPM algorithm should be very efficient. Therefore, the actual computational complexity depends on how the exact matching partition well achieved. The proposed algorithm is also sevaluated in firstly calculates the X-dimension for matching result. Then modify the input region of Y-dimension based on the matching result of the previous dimension. The proposed algorithm is more efficient most of overlapping degree but sometime the proposed algorithm has more execution time than the original layer partition-based algorithm [21], [22].

IV. SIMULATION

The proposed system focuses on the battleground simulation by using new layer partition-based matching algorithm for data distribution management system. The overall system design and architecture of implementing distributed data management system in battleground by using layer partition-based matching algorithm is shown in the following Fig. 4 and Fig. 5. It is implemented the simulation using java programming language. The system consists of four distributed federates and communicates with them via RTI.

A. Scope and Architecture of the Proposed System

The system consists of centralized FighterController, OverlapDetector, and Coordinator and distributed local DDM_Managers of each battalion groups. The system consists of four distributed groups. In the proposed system, there are two kinds of entities: fighter objects and federates (battalions). The system checks the movement of fighter at each time step and calculate the overlapping algorithm. When overlapping is detected, it informs the corresponding battalion.

In order to perform the overlapping detection, the system comprises three sub-models:

- FighterController: It is used to control the moving of the user defined fighter objects.
- OverlapDetector: It uses layer partition-based matching algorithm to detect the overlapping condition between fighter objects and 12 battalions.

- Coordinator: It performs the message transportation duty.

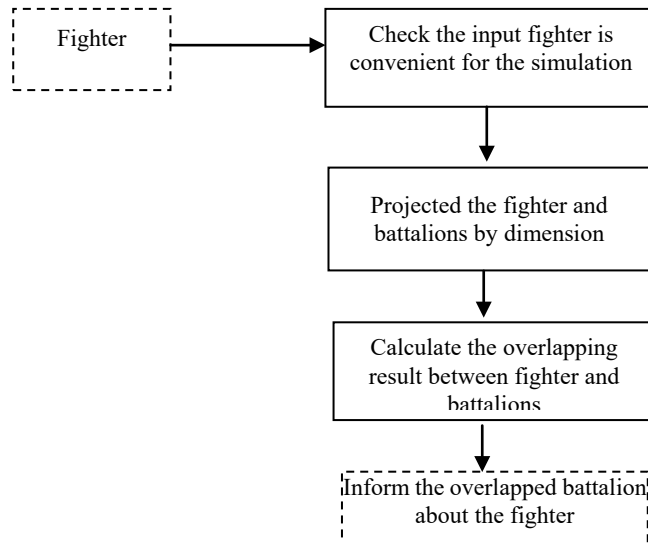


Fig. 4: Fundamental of proposed system.

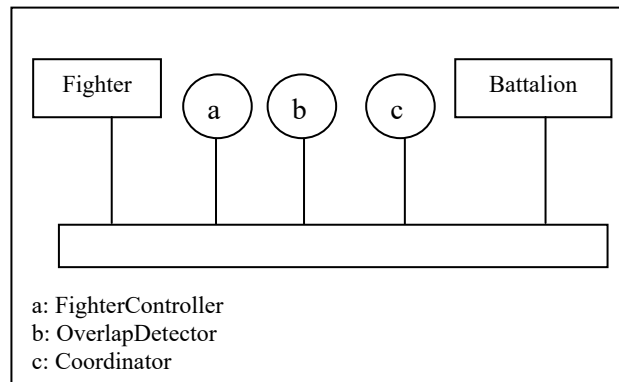


Fig. 5. The system architecture.

B. Overview of the Proposed System

The system will be implemented for dynamic fighter simulation object moving in a battleground, travelling at constant speeds in various directions. The number of objects is initially placed at user’s input boundary. It is assumed that the fighter moves constant velocity and constant direction for several time-steps. The position and movement of the fighter can be detected by investigation of FighterController. The regional battalions act as subscribers and fighter object acts as updater. To search the overlap information of fighter and regional battalions, the system uses layer partition-based algorithm. OverlappedDetector has detected the position of fighter; it continues to distribute the fighter’s information to the overlapped battalion which needs to know that information by means of Coordinator.

For simplicity, the fighter moves at the start of the random position on user’s defined four boundary side of the battleground, North, East, South and West. When the fighter object is moving to the destination boundary, the central regiment server detects the fighter’s position and search the overlapping result between the fighter and the 12 battalions. When the fighter object is reached on another side of the boundary of the battleground, this fighter is not considered in the detecting fighter and searching the overlapping information. All fighter objects pass through the boundary of

the battle ground the simulation is terminated. In the proposed system, twelve battalions are grouped in order to perform the relevant actions. Assume that these groups are in distributed manner. The Fig. 6 shows twelve regiments, their grouping.

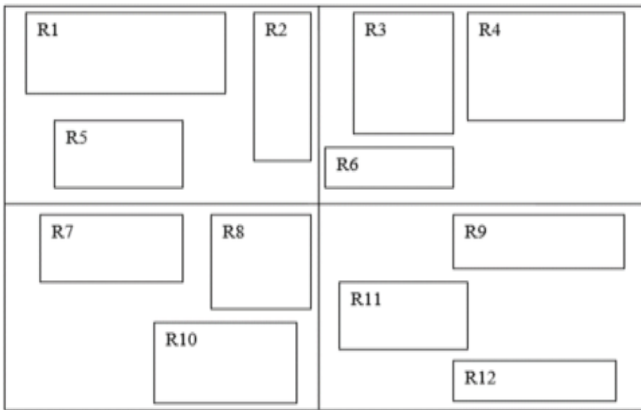


Fig. 6. Twelve regiments divided into four co-operated group.

C. Operational Components of the Proposed System

The system implements the simulation of detecting enemies' moving fighter objects by using layer partition-based algorithm. In this platform, the fighter objects are simulated using the FighterController sub-model. This sub-model is responsible for the creation and deletion of the simulated fighters from the runtime infrastructure. It also in control of the movement of the simulation objects within the routing space and it calculates the boundary point for each fighter and control the fighter object's movement. The fighter objects move according to the random position of the user's assigned direction and at each time step it checks for meeting the boundary of battleground area. When the fighter objects reach the other side of the boundary of the battleground, these fighters are removed from the current runtime infrastructure. Sequence diagram for the interaction between fighter object, FighterController and user is shown in Fig. 7.

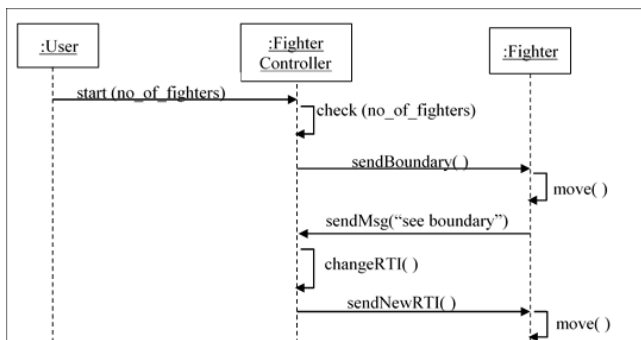


Fig. 7. Sequence diagram for the action of fighter controller component.

The OverlapDetector oversees data filtering strategy in this system. The position of the fighter objects may change at every time steps. The OverlapDetector search the overlapping information between the battalions and the fighters by using LPM. When it knows the overlapping information, it connects Local DDM_Manager of the regional battalions (federates). Then it sends the command directly to overlapping federates. If overlapping happens, it

produces actual overlap battalion name and sends information to that battalion. Therefore, the battalion may perform required action in timely manner.

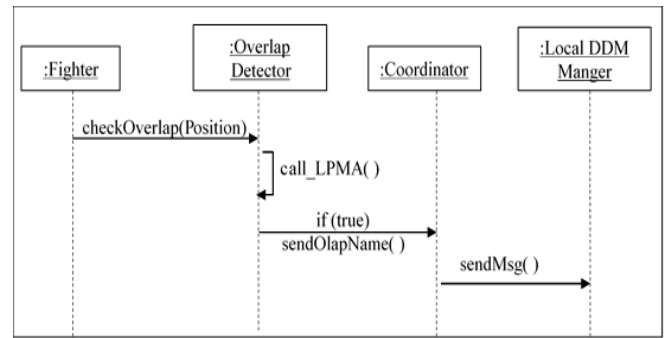


Fig. 8. Sequence diagram for the action of overlapdetector component.

In this sub-model, OverlapDetector require to communicate with Coordinator. When it knows the overlapping information, it connects Local DDM_Manager of the regional battalions (federates). Then it sends the command directly to overlapping federates. In the Fig. 8 shows the sequence diagram for interaction between Fighter, OverlapDetector, Coordinator and Local DDM_Manager, the fighter objects always send the position of the fighters to OverlapDetector to calculate the overlapping information. If OverlapDetector detected the overlapping results between battalions and fighter objects, it sends all overlapping fighters list to Coordinantor. When the fighters passed through the boundary of the battleground, these are not interested by the OverlapDetector and all these fighters are not considered in the runtime infrastructure. The Fig. 9 shows the algorithm incorporated in the system for performing the time-stepped simulation of the battleground.

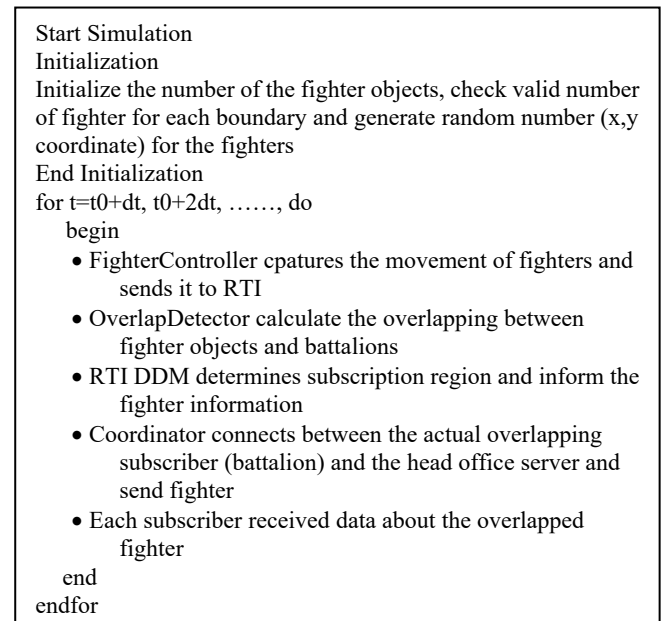


Fig. 9. A time-stepped distributed simulation algorithm.

The coordinator sub-model is responsible for communication between battalions and their local DDM managers. In the proposed system, it is to be assumed that four federate groups are in distributed manner. Each group

has three battalions. In order to connect them Coordinator uses RMI methodology.

The proposed algorithm selects the dynamic pivot by detecting the regions distribution on the routing space. Firstly, it generally defines the left set, right set and the pivot set. Before matching the left set and right set with the pivot set, the proposed system defines the exact matching area of regions for the left set and right set and calculate the subtract set for each left and right set upon the exact matching area. So, the proposed algorithm matches the exact matching area of the left set and right set with the pivot set. After matching the left and right sets with the pivot set, the two subtract lists are omitted in the input regions lists for the next repetition of the LPM. It is the detection and producing the overlapping results of the battleground simulation of large-scale distributed simulation in order to minimize subsequent computations and algorithm complexity.

V. CONCLUSION

The Efficient data distribution is an important issue in large scale distributed simulations with several thousands of entities. The broadcasting mechanism employed in Distributed Interactive Simulation (DIS) standards generates unnecessary network traffic and is unsuitable for large scale and dynamic simulations. In this paper present the performance of the layer partition-based matching algorithm based on the previous calculated dimension. The proposed algorithm also provides the minimum matching cost between the updaters and the subscribers within the routing space. The system can be used not only in the research purpose, but also in the military and civil applications. This system is developed to search the overlapping information between the data updaters and the data subscribers based on the time-stepped simulation's infrastructure. Due to the Layer Partition-based algorithm, the implementation can decrease the message traffic over the communication network. It can grow up the efficiency to distribute the updater's information. It can use in any distributed simulation in large scale distributed system. The small execution time for matching process can get from the simulation system with the use of this proposed algorithm when the overlapping degree is lower.

CONFLICT OF INTEREST

The author declare no conflict of interest.

AUTHOR CONTRIBUTIONS

The author, Nwe Nwe Myint Thein, studied thoroughly High Level Architecture's Services. The author interested in Data Distribution Management (DDM) Service. Firstly, the author applied existing algorithm of DDM and investigated them. Then, the author proposed the new algorithm for DDM Services. The author analyzed the proposed algorithm and compare with the existing algorithms. The author published many papers according to the analyzed result. During the research journey, the author was advised by the author's supervisor - Dr Nay Min Tun. The author had approved the final version.

ACKNOWLEDGMENT

I firstly thank my parent for their support in my pursuit of the research. I especially thank to my rector, Dr. Saw Sanda Aye and Dr. Ei Chaw Htoon, (Head – Faculty of Computer Science), for giving me the opportunity to continue my research at University of Information Technology (UIT). Heartfelt thanks are extended to my respectful Director, Dr Nay Min Tun, Ministry of Education, Myanmar, for his invaluable recommendations, detailed guidance, patient supervision and keen interest throughout the preparation of this research.

REFERENCES

- [1] G. J. Jense, N. H. L. Juijpers, and A. C. M. Dumay, "DIS and HLA: Connecting people, simulations and simulators in the military, space and civil domains," in *Proc. 48th International Astronautical Congress*, Italy, 1997.
- [2] S. D. Judith, M. F. Richard, and M. W. Richard, "The department of defense high level architecture," in *Proc. the 29th Simulation Conference*, Washington, DC, 1997, pp. 142-149.
- [3] J. R. Steven, "HLA/RTI data distribution management in the synthetic theater of war," in *Proc. Fall Simulation Interoperability Workshop*, Orlando, FL, September 1997.
- [4] A. Martin and S. Pierre, "Air-ground combat simulation through the ONERA HLA run-time infrastructure," in *Proc. the Advanced Simulation Technologies Conference*, April 2000.
- [5] M. F. Richard. (2014). *Dynamic Data Distribution.ppt*. [Online]. Available: <http://www.cc.gatech.edu/~fujimoto/>
- [6] M. F. Richard. (2014). *Static Data Distribution.ppt*. [Online]. Available: <http://www.cc.gatech.edu/~fujimoto/>
- [7] L. M. Katherine and S. S. Jeffrey, "Data distribution management in the HLA multidimensional regions and physically correct filtering," in *Proc. Spring Simulation Interoperability Workshop*, March 1997.
- [8] G. Tan, R. Ayani, Y. Zhang, and F. Moradi, "An experimental platform for data management in distributed simulation," in *Proc. Simulation Technology and Training Conference*, Sydney, Australia, 2000, pp. 371-376.
- [9] T. Gary, A. Rassul, Z. Yusong, and F. Moradi, "Grid-based data management in distributed simulation," in *Proc. the 33rd Annual Simulation Symposium*, Washington, April 2000, pp. 7-13.
- [10] Y. Zhang, "A simulation platform for investigating data filtering in data distribution management," M.Sc Thesis, School of Computing, National University of Singapore, Singapore, February 2000.
- [11] B. Azzedine and L. Kaiyuan, "A novel approach to real-time RTI based distributed simulation system," in *Proc. the 38th Annual Symposium on Simulation*, 2005, pp. 267-274.
- [12] K. Pamela, C. Aaron, and L. Ron. *Evaluation of Run Time Infrastructure (RTI) Implementations*. [Online]. Available: <http://www.scs.org/confernc/hsc/hsc02/papers/hsc017.pdf>
- [13] "Evaluation of an Efficient Algorithm for Data Distribution Management in High Level Architecture," Computer Science Technical Report CS-TR-05-12, School of Computer Science, University of Central Florida, Orlando, FL, 2005.
- [14] A. Rassul, M. Farshad, and T. Gary, "Optimizing cell-size in grid-based DDM," in *Proc. the 14th Workshop on Parallel and Distributed Simulation*, Bologna, Italy, May 2000, pp. 93-100.
- [15] T. Gary, Z. Yusong, and A. Rassul, "A hybrid approach to data distribution management," in *Proc. the 4th IEEE International Workshop on Distributed Simulation and Real-Time Applications*, 2000, pp. 55-61.
- [16] J. Yu, R. Come, and T. Gary, "Evaluation of a sort-based matching algorithm for DDM," in *Proc. the 16th Workshop on Parallel and Distributed Simulation*, Washington, DC, May 2002, pp. 68-75.
- [17] A. Junghyun, S. Changho, and G. K. Tag, "A binary partition-based matching algorithm for data distribution management," in *Proc. Winter Simulation Conference (WSC)*, Phoenix, AZ, 2011, pp. 2723-2734.
- [18] M. T. Nwe Nwe and T. Nay Min, "Optimization of region distribution using binary partition-based matching algorithm for data distribution management," *International Journal of Engineering research and Technology (IJERT)*, vol. 2, no. 2, February 2013, pp. 1582-1587.
- [19] M. T. Nwe Nwe and T. Nay Min, "Dynamic pivot for layer partition-based matching algorithm of DDM based on regions distribution," in *Proc. the 4th International Conference on Science and Engineering (ICSE)*, Yangon, Myanmar, December 9-10, 2013.

- [20] M. T. Nwe Nwe and T. Nay Min, "Layer partition-based matching algorithm of DDM," Presented at the 3rd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI), Singapore, February 11-12, 2014.
- [21] M. T. Nwe Nwe and T. Nay Min, "Layer partition-based matching algorithm of DDM based on dimension," *International Journal of Advanced Computer Technology (COMPUSOFT)*, vol. 3, no. 4, April 2014, pp. 691-695.
- [22] M. T. Nwe New, "Performance analysis of layer partition-based matching algorithm in data distribution management," in *Proc. the 2nd International Conference on Advanced Information Technologies (ICAIT)*, Myanmar, November 1-2, 2018, pp. 114-119.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).



Nwe Nwe Myint Thein was born at Yangon, Myanmar in April 1983. She passed the matriculation exam in March 2000. She completed B.C.SC.- Bachelor of Computer Science degree, B.C.SC. (Hons) - Bachelor of Computer Science (Honours) degree and M.C.Sc. - Master of Computer Science degree from the University of Computer Studies, Yangon (UCSY) in 2004, 2005 and 2008 respectively.

Nwe Nwe Myint Thein has a doctorate - Doctor of Philosophy (Information Technology) from University of Technology (Yatanarpon Cyber City), PyonOoLwin, Myanmar in 2014. She has more than ten years of teaching experience in Programming Logic and Design, Data Structures, C++ Programming, Java Programming, Android Programming, Computer Organization, Programming Language, Compiling Technique, Distributed Systems, Computing Paradigms in IOT. Now she is Lecturer of University of Information Technology, Yangon, Myanmar. She has served on conference and workshop program committees and served as the Technical Program Committee (TPC) member. Her main research field includes modelling and simulation, real-time infrastructure, data distribution management, distributed system, discrete event simulation, cloud computing, fog computing, blockchain, distributed ledger and virtualization technology.