

Design and Implementation of Data Distribution Management in IEEE 1516 HLA/RTI

Jung Hyun Ahn*, Jae Hyun Kim** and Tag Gon Kim***

School of EECS

KAIST

373-1 Kusong-dong, Yusong-gu

Daejeon, Korea 305-701

Tel: +82-42-869-5454

Fax: +82-42-869-8054

jhahn@smslab.kaist.ac.kr*, jhkim@smslab.kaist.ac.kr** , tkim@ee.kaist.ac.kr***

Keywords: Data Distribution Management, HLA, IEEE1516 RTI, Minimized Unicast

The Data Distribution Management (DDM) services, one category of HLA management services, control filters for data transmission and reception of data volume among simulators. In this paper, we propose design concept of DDM and show its implementation for RTI. The design concept of DDM is to minimize total amount of message that each federate and a federation process generate using the rate of RTI service execution. The design of our proposed DDM follows that a data transfer mechanism is differently applied as the rate of RTI service execution. A federate usually publishes or subscribes data when it starts. The federate constantly updates the data and modifies associated regions while it continues to advance its simulation time. Therefore, the proposed DDM design provides fast update or region modification in exchange of complex publication and subscription. This paper describes how to process the proposed DDM and experiment a variety of scenarios while modifying a region, changing a overlap ratio, and increasing data volume.

1 INTRODUCTION

The High Level Architecture (HLA) is the IEEE 1516 standard for interoperation between heterogeneous simulators which are developed with different languages and platforms [1] [2] [3]. The Runtime Infrastructure (RTI) is a software which implements the IEEE 1516.1 Federation Interface Specification which defines APIs for HLA services. The RTI provides commonly required services to the federate for cooperating in their execution [4].

The Data Distribution Management (DDM) services, one category of the HLA management services, filter the amount of data volume to reduce both data transmission and reception of irrelevant among simulators. The goal of DDM services is to enhance the data communication more

efficiently by sending data only to those simulators that need the data.

To communicate with each distributed simulator, it is needed to select an appropriate data communication [5]. The data communication includes broadcast, unicast (point to point) and multicast (multipoint to multipoint). The broadcast makes sending fast anywhere, but the receiver expends time on discarding irrelevant data. the broadcast is adequate for small systems. The unicast communicates point to point that receivers only get a relevant data. The multicast only delivers messages to the member of the multicast group. The multicast strikes a balance between broadcast and unicast by reducing the time to send and the amount of data received.

The previous research of DDM emphasizes multicast communication in order to communicate with each other in group. In several approaches of previous research that use multicast communication, Grid-Based DDM communicates with a cell which is partitioned by grid and provides a group communication per same cell [7]. It is more widespread and scalable than previous other approaches. On the other hand, there are some drawbacks when the multicast is applied to HLA/RTI. Even though the multicast has the potential of improvement of performance, a multicast hardware currently supports a limited number of multicast groups. And it is not suitable for Wide Area Network (WAN) environment if the router does not support multicast communication. It has join/resign overhead and group's maintenance overhead. Especially in Grid-Based DDM, it needs the Receiver-Sided filtering when there is no region intersection in same cell. It gets poor performance under non-optimized cell size.

Therefore, in this paper, we propose a design concept about the DDM which is applicable with RTI and show its implementation in the HLA Interface Spec. This is the minimized unicast communication. The design which is proposed follows that a data transfer mechanism is differently applied as the calling frequency.

2 BACKGROUND

2.1 Data Distribution Management in HLA/RTI [6] [8] [9] [10]

Consider a distributed simulation that has different residence of simulators, necessitating data exchange for interoperation. Data Distribution Management (DDM) enforces that the simulator sends and receives a relevant data they need during the simulation. One of the HLA Interface Spec., The DDM services is an important category service making the data communication more efficient by sending a data to those federates requiring the data.

These DDM services allow a federation to control routing and delivery of attribute updates and interactions on a specific instance with additional information, region. The publication and subscription of Declaration Management (DM) services allow the federate to promise to send the data, class attribute values and interaction parameters, previously. A federate which publishes (or subscribes) to a interaction parameter will send (or receives) the specific parameter in federation. This type of filtering is the class-based filtering. On the other hand, the DDM services allow a value-based filtering. The DDM services allow a federate to receive the subscribed attributes after associated region is intersected with the publishing federate's associated region.

The Class-based filtering may give more suitable than the value-based filtering at small federation. However it is necessary for large federation to filter more elaborate for data exchange during the simulation execution. Therefore The DDM, the value-based filtering, can be required for large federation with numbers of data volume.

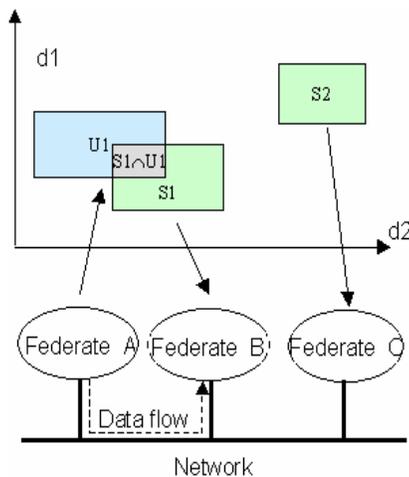


Figure 1 Data Distribution Management [10]

Figure 1 shows that Federate A has update region U1 and Federate B and Federate C have subscription region S1 and S2 individually within a two dimensional region. Update region is associated with published data and subscription region is associated with subscribed data.

In this example, U1 of Federate A and S1 of Federate B overlap and therefore attributes and interactions associated with U1 will be routed by the RTI to the Federate B that created S1. However U1 of Federate A and S2 of Federate C do not overlap and attributes and interactions will not be routed from Federate A to Federate C

2.2 Related Work

Over the past few years, several researches have been devoted to the DDM mechanisms in HLA/RTI. The grid-based DDM is one of DDM mechanisms that emphasizes multicast communication in order to communicate with each other.

In the grid-based DDM, the RTI initially partitions the space into a grid of cells along each dimension, and then assigns a multicast group to each cell. The grid-based DDM provides a group communication per same cell. Each cell manages a group list that contains update regions and subscription regions [7].

A federate joins to a multicast group associated with each cell. For each cell, the multicast group including subscribers whose regions overlap with that cell is maintained. When a publisher's update region overlaps with a cell, the data associated with the update region will be delivered to the multicast group for that cell. Only when the subscription region and update region occupy at least one common cell, the data associated with the update region will be routed to the federate which created the subscription region. It is more widespread and scalable than other previous approaches.

On the other hand, there are some drawbacks when the multicast communication is applied to the HLA/RTI. It is not suitable for Wide Area Network (WAN) environment if a router does not support the multicast communication. It has join/resign overhead and group's maintenance overhead. Especially in grid-Based DDM, it does not prevent irrelevant messages and needs the receiver-side filtering when there is no region intersection in same cell. There is still a possibility that unnecessary data flows in the network.

Moreover, the larger the cell size, the greater is the amount of unnecessary data that is communicated. Whereas, the smaller the grid cell size, the larger is the amount of work required to update the cell list. It gets poor performance under non-optimized cell size. So there should be an optimal size for the cells.

Therefore, in this paper, we propose a design of DDM which is applicable with the light-weighted RTI that is quick and simple to handle and show its implementation in the HLA Interface Spec. This is a minimum connected unicast communication, so that we can reduce total amount of generating messages that each federate and a Federation execution (Fedexec) which processes a federation and facilitates data exchange between participating federates.

Therefore the proposed DDM design provides fast update or region modification in exchange of complex publication and subscription using the rate of service execution.

3 PROPOSED DDM

3.1 The Rate of Services

The rate of DDM services is divided by the frequency which is called during whole simulation time. The low rate service means that it is rarely called on the middle which it simulates. As it accomplishes in only the phase which initializes the information of simulation, it is not called generally in simulation. The kind of low rate services is createRegion(), subscribeInteractionClassWithRegion() and so on in DDM services

In the other side, the high rate service means that it is plentifully called on the middle of simulation. Like this services continuously are called in simulation which is advanced. As it accomplishes in only the phase which initializes the information of simulation, it is not called generally in simulation. The kind of high rate services is commitRegionModification(), sendInteractionWithRegion() and so on in DDM services.

3.2 Data Transfer Mechanism

We consider two different mechanisms for data transfer mechanisms, a centralized approach and a distributed approach depending on where a data store.

The centralized approach is stored and managed information by center (i.e. Fedexec). It offers easy data management but the Fedexec may become the bottleneck.

Whereas the distributed approach is stored and managed information by a local components. It has data individually and send direct to destination but has difficult data management in a sparse environment.

The Figure 2 shows the centralized approach. If individual federate want to communicate with each federate, the Fedexec is a coordinator and controls the data flow. And Figure 2 shows the distributed approach. When a federate communicate with each other federate, the federate directly sends the data to other federates.

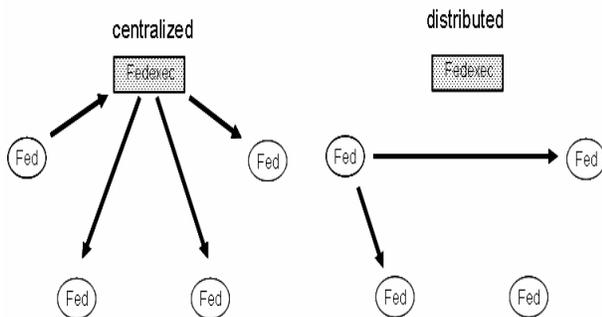


Figure 2 centralized approach vs distributed approach

Table 1 Service Rate

Service	Rate of service	Approach
Associate(Unassociate) Region For Updates	Low	Centralized
Create Region / Delete Region	Low	Centralized
Commit Region Modification	High	Distributed
Register Object Instance With Region	High	Distributed
Request Class Attribute Value Update With Region	High	Distributed
Send Interaction With Region	High	Distributed
Subscribe Interaction (Object Class) Class With Region	Low	Centralized
Unsubscribe Interaction (Object Class) Class With Region	Low	Centralized

3.3 Design Concept

The proposed DDM designs to minimize of total number of messages using the rate of service execution. This is a minimum connected unicast communication, so that we reduce total amount of generating messages that each federate and a Federation execution (Fedexec) which processes a federation and facilitates data exchange between participating federates. The design of our proposed DDM follows that a data transfer mechanism is differently applied as the calling frequency.

Our approach is based on the centralized approach at low rate services and the distributed approach at high rate services. Combined centralized and distributed approach can provide the advantages of both. As the high rate services are applied to the distributed approach the federate sends directly a message to the destination federate. The Table 1 presents applied data transfer approaches per the service

In detail, first, the data transfer mechanism for low rate services is a centralized approach that manages data at center (i.e. Fedexec). A federate usually publishes or subscribes data when it starts. The service frequency is low in whole federation. Our design is to use complex message sequence for low rate services. In that manner, we obtain additional data such as the addresses of subscribers. The addresses help a federate directly send a message for the minimum connected unicast communication.

Second, data transfer method for high rate services is a distributed approach that manages data at local part (i.e. Federate). The federate constantly updates the data and modifies associated regions while it continues to advance its simulation time. The service frequency is high in whole federation. Our design is to use simple message sequence for high rate services. To do so, a publisher, a producer of message, has the addresses of subscribers which are

consumers and directly sends the data to its subscribers. Also the sender-sided filtering is achieved. The proposed DDM design provides fast update or region modification in exchange of complex publish and subscribe services. Figure 4 depicts the data transfer mechanism through the rate of service execution. Therefore, in this paper, we apply and develop the proposed DDM in the light-weighted RTI. This makes the federate get the minimum connection with each other federate.

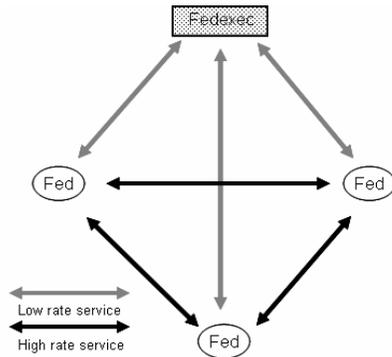


Figure 3 proposed DDM approach

3.4 Message Sequence

We observe the sequence diagram How the federate and Fedexec interact. The sequence diagram shows the stage of process in finding the destination federate with Publish(P) and Subscribe(S) information and in sending actually to the destination federate. The objective of our design is the minimum message delivery. To do like that we obtain light-weighted RTI, which it is quick and simple to handle.

In various RTI services, we are concerned with publishInteractionClass() which a federate declares to begin generating interactions of a specified class and sendInteractionWithRegion() which a federate declares a federate's interest in receiving a specified class of interactions with region.

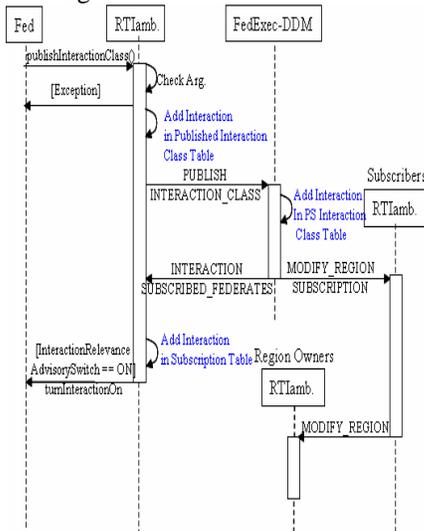


Figure 4 publishInteractionClass()

The Figure 4 depicts the sequence diagram of publishInteractionClass(). With the HLA spec., the publishInteractionClass() that is called in compliance with the user sends a message with Fedexec-DDM through the RTIambassador. This service can publish the interaction which a federate wants to send the interaction. This service informs Fedexec-DDM that the federate intends to give the specified interaction. When the argument check is done, the RTIambassador adds the interaction handle in Published Interaction Class Table and then the RTIambassador sends the message of PUBLISH_INTERACTION_CLASS to Fedexec-DDM. As followed, Fedexec-DDM sends the message of INTERACTION_SUBSCRIBED_FEDERATES to the publisher's RTIambassador and the message of MODIFY_REGION_SUBSCRIPTION to the subscriber's RTIambassador. This service is completed when the federate receives the callback function of turnInteractionOn() if the InteractionRelevanceAdvisory Switch is on.

This sequence is very complex. But, to do so, the federate has the information of subscribers' address. Although these services of sequence are complex, it is called low rate and the message sequence does not increase the number of total messages.

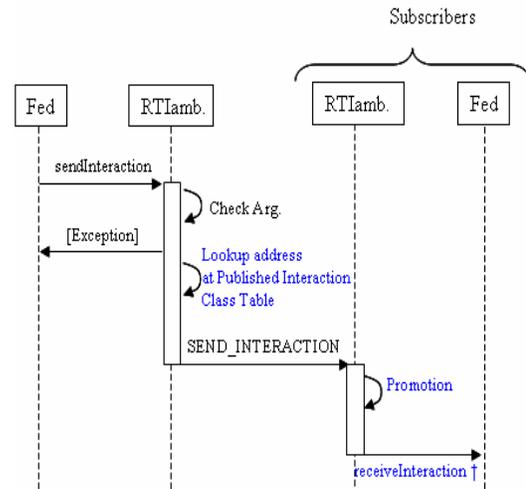


Figure 5 sendInteractionWithRegion()

The Figure 5 shows the sequence diagram of sendInteractionWithRegion(). This service can send the interaction actually by address of subscribers. This service is high rate service so that a federate passes through the subscribers. As a federate has the information of the subscribers' address, the federate send directly a message to the subscriber. This sequence is very simple. Although these services is frequently called, the message sequence does not increase the number of total messages because of simple message sequence.

4 EXPERIMENT AND RESULT

In order to check the performance in the proposed DDM, we develop the sender and receiver federates in test federation. The experimental result of the test federation is then evaluated and compared to study the performance of our proposed DDM design. The test federation was carried out on three Pentium IV 2.8GHz PCs ran Microsoft Windows XP: one PC is used to run rtiexec (SMSRTI) [11] and the other two PCs for the two separate federates. The three PCs have 512MBytes of RAM memory and interconnected by a dedicated Fast Ethernet. The experiment described in this section are exercised with eight federates in test federation. We just focus on the latency with processing time in test federation and the spent-time on network.

Our federation have a test environment to periodically send and receive interactions. To measure the latency from sending time to receiving time, the federation was repeatedly executed for several times to compute their average latency. We can check the time at both start point and end point of the interval callback function in the sender federate and the receiver federate. The latency is just difference between two time values.

To represent the performance of our proposed DDM, three different parameters are introduced [12]. The first is Data Volume. The Data Volume is the number of data which is used in simulation. A lot of data will require more memory and more processing time. The second is Modification Rate. The Modification Rate is the changed region number per total region number within some time-interval. Whenever the region is modified, the RTI must recalculate the region's intersections. The higher Modification Rate, the more consuming-time when compare with modified regions. The third is Overlap Ratio. The Overlap Ratio is the intersected region number per total region number. Since all region intersections must be recheck when the region is modified, the intersected region will require more processing time.

We experiment variable scenarios while modifying a region, changing a overlap ratio, and increasing data volume. We create region statically and an interaction is associated with each region in test federation.

The first experiment measures the latency with 20 interactions which is defined in Federation Object Model (FOM) and second experiment measures the latency with 200 interactions. The Figure 7-8 show the difference between the latency with 20 interactions and the latency with 200 interactions, respectively.

As shown in figure, when the number of interactions increased and the Overlap Ratio and the Modification Rate changed higher, the latency increased as well. The experimental results show that performance can vary when the scenario have changed. That suggests the dynamic parameters affect the performance of simulation. If some scenarios need DDM services, the more attention is required to apply with DDM services.

While the proposed DDM with SMSRTI does not provide good performance for actually implementing and checking DDM within test federation, it can serve the tendency characteristic which follows in various DDM scenarios with dynamic parameters.

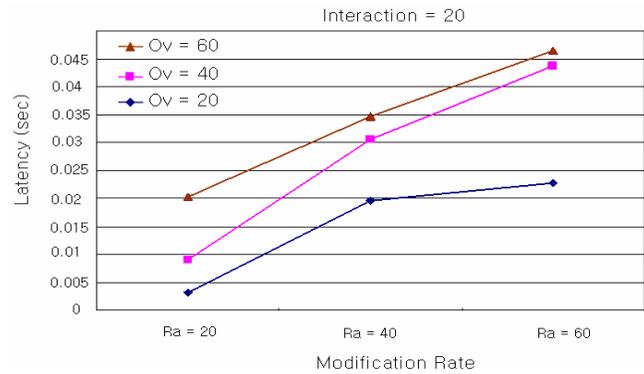


Figure 6 The latency with 20 interactions

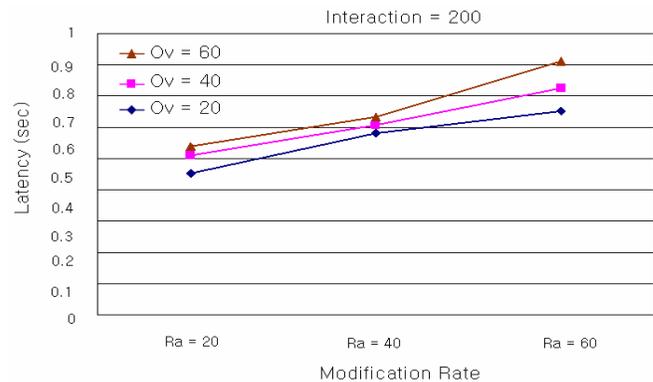


Figure 7 The latency with 200 interactions

5 CONCLUSION

We have represented the appropriate design concept for light-weighted RTI. The proposed DDM provides the minimized unicast communication, parameterized using the rate of RTI service execution. Low rate RTI services are a centralized approach while high rate RTI services are a distributed approach for data transfer mechanisms. Our Combined centralized and distributed approach can supports that the high rate services sends directly message the destination federate with help of the low rate services. The proposed DDM is suitable for the lighted-weighted RTI (i.e. SMSRTI). We describe how to process the proposed DDM and measure its performance of each scenario which may vary.

6 REFERENCES

- [1] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules, Std 1516, 2000.

- [2] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification, Std 1516.1, 2000.
- [3] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT), Std 1516.2, 2000.
- [4] Defense Modeling and Simulation Office, 1998, High Level Architecture Interface Specification, Version 1.3. Washington D.C.
- [5] Fujimoto, 2000, "Parallel and Distributed Simulation Systems," John Wiley and Sons, New York.
- [6] K. L. Morse and J. S. Steinman, "Data Distribution Management in the HLA: Multidimensional Regions Physically Correct Filtering," Proceedings Spring SIW Workshop, 1997, 97S-SIW-052
- [7] Gary Tan, Rassul Ayani, Yusong Zhang and Farshad Moradi, 2000, "Grid-based Data Management in Distributed Simulation," in Proceeding of 33rd Annual Simulation Symposium.
- [8] Daniel J. Van Hook et al; 1996, "Approaches to RTI Implementation of HLA Data Distribution Management Services," Proceeding of the 15th DIS Workshop.
- [9] Van Hook, Calvin, 1998, "Data Distribution Management in RTI 1.3," Proceedings of the Spring Simulation Interoperability Workshop; 98S-SIW-206.
- [10] Petty, M. D. and K. L. Morse. "The Computational Complexity of the High Level Architecture Data Distribution Management Matching and Connecting Processes", Simulation Modelling Practice and Theory, Volume 12, Issues 3-4, July 2004, pp. 217-237.
- [11] J. H. Kim, 2006, "Proposal of High Level Architecture Extension and Run-Time Infrastructure Implementation," Ph.D. Thesis, KAIST
- [12] K. L. Morse, "Characterizing Scenarios for DDM Performance and Benchmarking RTIs," Proceedings of the Spring Simulation Interoperability Workshop; 99S-SIW-054.