Implementation of DDM in the MAK High Performance RTI

Douglas D. Wood
185 Alewife Brook Parkway
Cambridge, MA 02138
dwood@mak.com

Keywords:
HLA, RTI, DDM

ABSTRACT: *The HLA data distribution management (DDM) services have been implemented in the MAK High Performance Runtime Infrastructure (RTI). The implementation of the DDM services has followed a distributed region approach. This paper describes the distributed region approach and its implementation in the MAK High Performance RTI using multicast groups.*

The distributed region approach distributes region information throughout the federation to dynamically match publication and subscription regions. The distribution of region information throughout the federation as well as processing updates, interactions, and interest advisories with respect to regions introduces overhead. The paper examines this overhead in relation to RTI and federate processing and network bandwidth. Suggestions for optimizing the use of DDM both internal and external to the RTI are addressed.

1. Introduction

Data distribution management (DDM) in the High Level Architecture (HLA) controls the distribution, or routing, of data between federates. Through the DDM services, federates express the relevance of data so that the Runtime Infrastructure (RTI) can prevent the delivery of irrelevant data. By preventing the delivery of irrelevant data between federates, the RTI can improve the utilization of processing and network resources. Where and how the RTI establishes the routing of data greatly affects the use and efficiency of the DDM services. Therefore, federate and federation developers must consider the underlying DDM implementation strategy when using these services.

RTI developers have used three approaches to implement DDM; they are receive filtering, fixed grid, and distributed region. Some form of receive filtering is normally required to compensate for less than perfect routing in the other two approaches. The MAK High Performance RTI implements DDM (as defined in the 1.3 specification) using distributed region with receive filtering. This paper describes the distributed region approach, some performance aspects of its implementation, and how the implementation affects the use of DDM services. The fixed grid approach is briefly described to serve as a comparison.

2. DDM Implementation Approaches

DDM implementations seen in RTIs to date have taken either a fixed grid approach [1, 2] or a distributed region approach [2, 3, 4]. Both approaches have used multicast routing as the underlying mechanism for low-level relevance filtering. The chief difference between the two approaches is how the multicast groups are associated with region matches to form data communication channels between federates. In either approach, the principle of using multicast filtering yields less than perfect results in practice. Both approaches require additional receive filtering schemes to ensure that the local RTI component (LRC) performs perfect filtering as required by the HLA interface specification.

2.1 Fixed Grid Approach

The fixed grid approach relies on configuring the assignment of multicast groups in advance of the federation execution. The RTI runtime initialization data (RID) file is generally used to specify a partitioning of the routing spaces along each dimension to form grid cells. In some fashion, the RID parameters specify an assignment of multicast addresses to each grid cell. During the federation execution, the LRC determines the grid cells that a region intersects. The result associates a region with the multicast groups from intersecting grid cells in a fixed and predetermined fashion. This a priori knowledge about multicast group distribution within routing spaces allows LRCs to independently determine the multicast group used for sending or receiving data. The LRC sends attributes or interactions associated with a region to each of the region's multicast groups. In turn, the LRC joins each multicast group of a region that a federate uses for subscription.

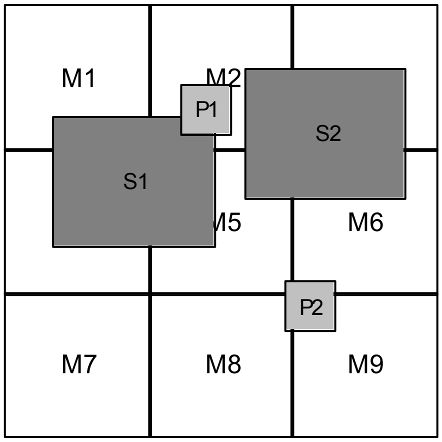

Figure 1. Fixed grid with overlaid regions

The advantage of this approach is that the multicast filtering mechanism requires no exchange of region information between LRCs. It localizes the effects of region associations and modifications on filtering. However, the intersections between regions and grid cells do not always correlate exactly with intersections between update and subscription regions. An LRC will most probably receive attributes and interactions that do not match the federate's region subscription criteria. In Figure 1, each grid has a multicast address M*i* and regions are overlaid on top to show intersections. The publication regions, P*i*, depict regions used for sending interactions and updating objects. The subscription regions, S*i*, depict regions used for subscribing to interaction and object classes. The grids that a publication region overlap determines which multicast address(es) to use for transmitting attributes. The grids that a subscription region overlap determines the multicast groups to join. A federate subscribed using S2 will receive attribute updates associated with both P1 and P2 even though there is no overlap between the two regions. Further, because P2 overlaps both M5 and M6, the attributes must be transmitted to both (M8 and M9 as well) and the subscription with S2 will result in both transmissions being received. The LRC will have to perform some additional filtering to prevent duplicate updates from being delivered to the federate.

Since the fixed grid approach does not perform perfect routing, the LRC must perform receive filtering based on the update and subscription regions. However, this receive filtering requires the exchange of region information. The exchange of region information is also required to support interest advisories (e.g., attributesInScope). Therefore, the perfect filtering mandated by the HLA interface specification diminishes the advantages of a fixed grid approach. A disadvantage of the fixed grid approach is that multiple transmissions are required when a region used for updating intersects multiple grid cells. A work-around for this drawback is to only use point regions for updating, but that requires the region to be updated at a higher rate and is insufficient for non-point objects. Another disadvantage of this approach is that it requires a large number of multicast groups in order for each grid cell to have a unique assignment. The reuse of multicast groups across grid cells is a work-around that relies on receive-filtering to compensate. A variable resolution grid is another technique that can better utilize limited multicast addresses [1]. Areas of high activity are appropriated a finer grid resolution and therefore more multicast addresses to increase filtering. Areas of low activity are defined with a coarser grid and use fewer addresses.

2.2 Distributed Region Approach

In the distributed region approach, each federate's LRC exchanges its region information with remote LRCs (or other RTI components) to distribute the region information among the LRCs. The individual LRCs perform matching between the local and remote regions. The LRC uses the region matches to establish communication channels between publishers and subscribers. In some implementations, a separate component or agent may take part in establishing the channels for a federate or group of federates [2, 3]. The channels can transmit through a multicast group to support low-level relevance filtering.

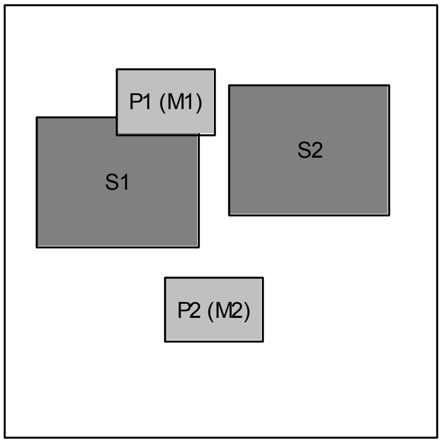

Figure 2. Distributed region using update regions

One scheme for establishing communication channels is to simply assign a unique channel to each region used for updating (update regions). The LRC transmits attributes associated with an update region using its communication channel. The LRC matches its local regions used for subscription (subscription regions) against remote update regions. An LRC listens to an update region channel only if it matches a subscription region. An update and subscribe region match if the regions overlap and if there are any common attributes. In Figure 2, each update region has its own multicast address and subscription regions join the multicast group of overlapping update regions.

A more sophisticated approach for establishing communication channels uses subscription sets. The DMSO 1.3 RTI used this approach [3]. An LRC determines a subscription set by matching its local update regions with remote subscription regions. The subscription sets specify which federates must receive the interactions and object attributes associated with update regions. The LRC establishes a separate communication channel for each subscription set. In Figure 3, both P1 and P2 have a subscription set of S1 and S2 while P3 has a subscription set of S1.

The advantage of distributed region matching is that the LRC can establish communication channels that carry relevant data only to receiving federates. Additional unsubscribed attributes may be carried through the channel, but this multiple association problem exists in practically any approach including the fixed grid approach. Also, the transmission of attributes and interactions are always only sent to a single channel. The disadvantage is the considerable overhead of distributing the region information and performing matching between regions. With the distributed region approach, the overhead required to maintain regions becomes similar to objects (with reliable attributes). A federation that naively uses regions and region updates can easily overwhelm any advantages of DDM implemented this way. As pointed out in [5, 6], region matching scales poorly with the number of regions. However, matching can be optimized using computational geometry techniques [7].

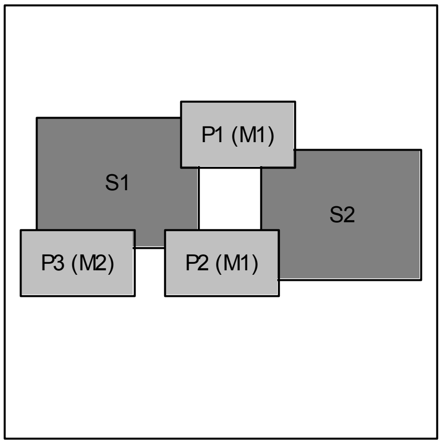

Figure 3. Distributed region using subscription regions

2.3 Multiple Region Associations

One facet of DDM that the HLA 1.3 interface specification does not make clear is that attributes can be associated with multiple regions. In fact, existing RTIs only allow attributes to be associated with a single region [8]. Under 1.3, associating an attribute with multiple regions provides little benefit in that attributes can only be associated with regions from a single routing space (fixed set of dimensions) and a region can have multiple extents (disjoint volumes within a single region). However, the 1516 specification removes the routing space concept (dimensions specified per region) and limits a region to a single extent (single volume per region) [9]. While the changes to DDM in the 1516 specification made it more flexible, it specifically and functionally mandates supporting attribute associations with multiple regions or region sets. Region set associations are necessary to ensure the same level of functionality provided by a single region association in the 1.3 DDM specifications. Looking ahead at the transition to a 1516 implementation, the MAK RTI implements DDM to support attribute associations with region sets versus a single region.

3. Distributed region Implementation

The MAK RTI implements DDM using the distributed region approach with receive filtering and communication channels based on update regions. The implementation supports associating attributes with multiple regions. The LRC builds collections of regions into region sets as the federate uses them in associations (update region sets) and subscriptions (subscription region sets). It assigns a unique channel identifier to each update region set. Each LRC distributes its region set information along with its region information to all other LRCs. At any given time, each attribute instance is associated with exactly one update region set and each class attribute is associated with exactly one subscription region set. Interactions can be associated with multiple update region sets, but with only a single subscription region set at any given time.

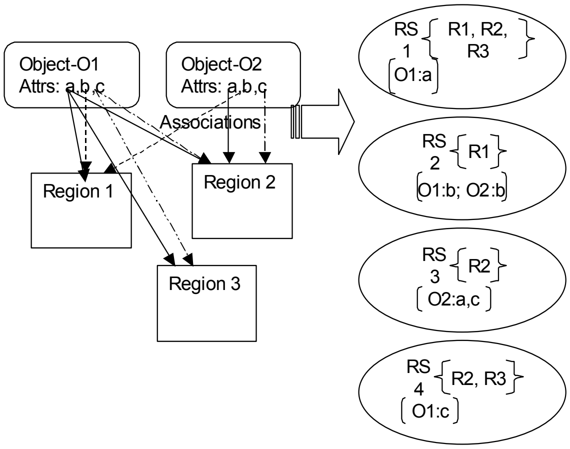

Figure 4 Region set associations

The region sets can be associated with multiple attributes from different objects. In Figure 4, attributes from two objects are associated with three regions to create four region sets. No restrictions are placed on the class of the object, but filtering is diminished if a region is associated with attributes from different classes. For interactions, the region sets are created separately for each interaction class.

The MAK RTI supports relevance filtering using multicast groups assigned to the update region sets. An LRC assigns a multicast address from a given pool of available addresses to each update region set. The LRC distributes the multicast assignment information with the region set to all remote LRCs. When a LRC determines that a remote update region set matches a local subscription region set, then it joins the update region set's multicast group.

The LRCs must still perform receiver-based filtering even though the LRCs distribute all of the region information and can perform perfect region matching. The pool of multicast addresses available to any given LRC may be limited. The LRC may have to assign a multicast address to more than one update region set. Any given region set may have multiple associated attributes. A subscription region set may match only a subset of these attributes; however, matching any one requires that the LRC add the update region set to the accepted list. A receiving LRC must filter out any unwanted attributes.

3.1 Region Distribution and Matching

When a federate creates a region, the LRC does not immediately distribute the region information. It is not until the federate invokes the notifyAboutRegionModification service that the LRC transmits the region description to other remote LRCs. Each time the federate invokes this service, the LRC retransmits the region description.

LRCs perform no matching with the region by itself. The LRC performs matching using region sets. The LRC creates a region set when the federate associates a region (update region) or subscribes with a region (subscription region). Update region sets are created by these services:

- RegisterObjectInstanceWithRegion
- AssociateRegionForUpdates
- UnassociateRegionForUpdates (by subtraction)
- SendInteractionWithRegion

Subscription region sets are created by these services:

- SubscribeObjectClassAttributesWithRegion
- UnsubscribeObjectClassWithRegion
- SubscribeInteractionClassWithRegion
- UnsubscribeInteractionClassWithRegion
- RequestClassAttributeValueUpdateWithRegion

The LRC immediately distributes new region set and region set association information (both subscription and update) to remote LRCs.

An LRC matches its local region sets against any remote region sets (update against subscription and vice versa). Region set matching involves determining region overlap and attribute matches. The overlaps between local and remote update and subscription region sets are determined irrespective of attribute associations when region sets are created or discovered. The LRC also maintains matches between region sets (e.g., region sets that are associated with the same class attributes). When two region sets are both overlapping and matching, they satisfy the relevance criteria. The LRC also evaluates overlaps for each region set that contains a region updated by the notifyAboutRegionModification service. The LRC evaluates overlaps only when it creates region sets or when the federate modifies a region contained in the region set. The LRC evaluates matching only when attribute associations change. When the federate invokes the deleteRegion service (only on regions that are not in an active region set), the LRC destroys any region sets that contain that region.

3.2 Receive Filtering

The LRC performs efficient receive filtering with no region matching. The LRC performs region matching between the local and remote region sets only during region associations or region updates. For each remote update region set that matches a local subscription region set, it collects the channel identifiers in an accept list. When an LRC transmits an interaction or object attributes, the appropriate channel identifier is included in each message header. This channel identifier is the only region information included in the interaction and update messages. When receiving an update or interaction, the receiving LRC uses a simple lookup in the accept list, filtering any message with a channel identifier not in the list. The LRC constructing the update or interaction message also does not perform any region matching. Each interaction and attribute instance maintains a reference to the channel (region set) to be used for transmission. The LRC uses the reference to directly retrieve the channel information. The LRC must evaluate received attribute lists to filter unsubscribed attributes. The LRC sends all attributes associated with an update region set together. A subscribing federate may only be subscribed to some subset of these attributes and the LRC must filter the unsubscribed attributes that it receives.

3.3 Attribute Passelization

When a federate invokes an updateAttributeValues service, the LRC separates the submitted attributes into the designated region set channels formed from previous region associations. Therefore, the LRC may transmit more than one message for a given update even though all the attributes have the same transport type and order type. The LRC shares region sets across all object classes but creates them separately for interaction classes. A federate should use separate regions when associating regions across multiple objects of different classes. Otherwise, the LRC will transmit the attributes from these objects using the same channel and filtering will be diminished.

4. Costs and Benefits of DDM

The intent of DDM is to reduce the delivery of irrelevant data both to federates and onto the network. However, the use of DDM introduces overhead that can diminish, if not overtake, the benefits of data routing and interest advisories. The cost of DDM is the amount of processing and bandwidth required to set up communication channels for data routing and any additional overhead added to data transmission. The benefit is the reduction in delivery of irrelevant data and the processing required to filter it. This section provides more detail on how the implementation factors into the cost of using DDM.

4.1 Creating and Distributing Regions

Regions, region sets, and region associations act like additional objects in the federation. For now, the MAK RTI uses a naïve strategy for distributing regions so that changing extents or associations requires reliably transmitting the changes to all the LRCs (missing a change would corrupt the routing configuration). There are no recursive DM or DDM strategies to limit the distribution and processing. Other implementations have used strategies to aggregate or localize regions so a reduced set of data is distributed [2, 3]. However, these strategies reduce the filtering performed by the multicast routing mechanism and require the actual extent information to be carried with the transmission since the complete region information is not distributed.

Since distributing region information is a costly operation, the federate and federation should be designed to minimize the number of regions and the frequency that they are modified. Using fixed regions is a good example of a different way to attack DDM [4]. Rather than moving regions to follow an object, a set of fixed regions are placed so that the object changes associations from region to region. There is still some overhead involved with changing region associations, but it may be less than changing extents and reevaluating overlaps.

The typical approach of creating one or more separate regions for each object seems to place too much reliance on the RTI to optimize the management of regions. For large exercises with thousands of objects, this strategy seems too simplistic, especially since most of the objects are generated by computer generated forces systems with large numbers of objects apiece. When it comes down to the actual mechanics, DDM operates on the exchange of data between federates rather than between objects. When the number of systems is in the tens to hundreds, it seems like overkill to create thousands of regions to define the connection topology. Also, the federate is better equipped to take advantage of the semantic knowledge of the routing dimensions. It can use this information to aggregate objects into common region associations to balance maximizing relevance filtering while minimizing the number of regions and modifications to them.

4.2 Region matching

The MAK RTI uses a simple approach for region matching. Remote and local regions are evaluated against their corresponding update or subscription region. The cost of these operations is polynomial for the number of regions (each local region must be evaluated against every remote region). There are a few strategies for reducing the cost of overlap calculations such as quad trees or d-rectangle intersection [2, 3, 7, 10]. In theory, these optimizations will perform well. In practice, there is considerable overhead for setting up the data structures to support them, especially since attribute matching and multiple region associations must also be considered. However, given a large enough population, these strategies will provide their theoretical results.

4.3 Communication Channels

As long as the number of update regions is reasonable, assigning multicast addresses to update region sets should provide significant relevance filtering. Contrary to what is indicated in [3], a unique multicast address for each update region set would result in essentially perfect filtering, if attributes from differing classes are not associated with the same region. Using subscription sets would also provide perfect filtering with fewer multicast addresses.

4.4 Interest Advisories

Through scope and registration controls, the RTI uses the interest advisories to direct federates to reduce the data that they put on the network. The use of multicast groups for data routing reduces the data received and processed by an LRC; however, without low-level multicast filter configuration of routers across WANs there is no similar reduction in bandwidth. The interest advisories are where significant bandwidth reduction can be realized directly through the RTI. The interest advisories can also throttle publishing federates internal processing when the published model is only intended for remote consumption. For example, a target generator can use a lower fidelity model to maintain an approximate state for objects that are not subscribed or in scope and then switch to a higher fidelity model when they are in scope.

Since distributed region approaches already distribute DDM (and subsequently DM) information, the necessary information is at hand to support interest advisories. A fixed grid approach requires a supplemental mechanism to pass the DDM and DM information between the federates that results in similar sharing of information as the distributed region approach. The DMSO RTI NG uses a periodic update mechanism to distribute this information. The impact of this mechanism is shown in [11].

4.5 Flexibility of Distributed Regions

The distributed region approach can be adapted in many ways to optimize how communication channels are established. The current implementation in the MAK RTI bases its communication channels on update regions. The processing of regions and region matching is performed in each LRC. Because the MAK RTI distributes both update and subscription regions, it can be adapted to establish communications based on subscriber sets as in the DMSO RTI 1.3 [3]. The use of multicast addresses could be monitored and reconfigured to optimize their assignment to the communication channels. Through adaptation (e.g., recursive DDM or region clustering), the distributed region approach can be optimized to reduce the overhead of distributing the region information. Optimizing the perfect filtering required by the HLA Interface Specification has the greatest flexibility with a distributed region approach.

In contrast, the fixed grid approach is limited in how it can be adapted and optimized. It is by nature a static architecture. The basic premise requires a static configuration of multicast groups to the fixed grid. It might be envisioned that the multicast address assignment or even the grid itself could be dynamic. However, such a scheme might easily approach the complexity and overhead of distributing the regions. In fact, one of the biggest drawback of the fixed grid approach is that it must distribute region information to adhere to the HLA interface specification to support perfect filtering and interest advisories. The result is that distributing region information must be implemented in some way thereby diminishing the chief benefit of the fixed grid approach (i.e., not distributing region information). The fixed grid approach achieves the best results when perfect filtering is relaxed and the only guarantee is that received data will fall within grid cells overlapped by subscription regions.

5. Summary

The MAK RTI implements a distributed region approach to support the DDM services. The distributed region approach inherently supports the perfect filtering required by the HLA Interface Specification. The use of region sets conforms to both to the HLA 1.3 draft and the HLA 1516 standard requirements for the DDM services. The implementation separates region matching from routing and filtering the data. The implementation minimizes the overhead for sending the update and interaction messages with only a single 32-bit identifier added to update and interaction messages. While the distributed region approach does incur overhead for distributing and matching region information, sophisticated algorithms can be used to optimize those factors. The distributed region approach favors a federation that efficiently manages the number of regions and region modifications. The distributed region approach is flexible and can support various communication and connection strategies.

6. References

[1] S. Rak, M. Salisbury, R. McDonald: "HLA RTI Data Distribution Management in the Synthetic Theater of War" Simulation Interoperability Workshop, paper 119, Fall 1997

[2] D. Van Hook, S. Rak, J. Calvin: "Approaches To RTI Implementation of HLA Data Distribution Management Services" 15th Distributed Interactive Simulation Workshop, paper 084, Fall 1996

[3] D. Van Hook, J. Calvin: "Data Distribution Management in RTI 1.3", Simulation Interoperability Workshop, paper 206, Spring 1998

[4] M. Hyett, R.Wuerfel: "Effective DD Usage within the RTI-NG" Simulation Interoperability Workshop, paper 091, Fall 2001

[5] B. Helfinstine, D. Wilbert, M. Torpey, W. Civinskas: "Experiences with Data Distribution Management in Large-Scale Federations", Simulation Interoperability Workshop, paper 032, Fall 2001

[6] M. Petty: "Geometric and Algorithmic Results Regarding HLA Data Distribution Management Matching" Simulation Interoperability Workshop, paper 072, Fall 2000

[7] M. Petty: "Computational Geometry Techniques for Terrain Reasoning & Data Distribution Problems in Battlefield Simulation" Ph. D. Dissertation, University of Central Florida, Summer 1997

[8] DMSO, SAIC, VTC, OSC, DAS: "High Level Architecture Run-Time Infrastructure RTI 1.3-Next Generation Programmer's Guide"

[9] IEEE: "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification", 2001

[10] M. Petty: "Experimental Comparison of d-Rectangle Intersection Algorithms Applied to HLA Data Distribution", Simulation Interoperability Workshop, paper 016, Fall 1997

[11] K. Morse, F. Hodum, P. Wickes: "Attribute Level Advisories and Scalability", paper 066, Fall 2001

Author Biographies

DOUGLAS D. WOOD is a Senior Programmer-Analyst at MÄK Technologies. Mr. Wood is leading the development of an optimized HLA environment; a major part of which is enhancing and optimizing the MÄK High Performance RTI. Over the past 12 years, he has performed research and software development in distributed simulation, including DIS-HLA translation, electronic warfare protocols, emergency management training and computer generated forces. Mr. Wood received an MS and BS in Computer Science from the University of Central Florida. His interests are in simulation and distributed systems.