Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012)

An Interest Management in Large-scale Analytic Simulation

Mei Yang, Xinye Zhao, Yun Zhou, Ying Cai
Department of Control Science and Engineering
College of Electromechanical Engineering and Automation
National University of Defense Technology,
Changsha, China
yangmei@nudt.edu.cn

***Abstract*—Using interest management, a lot of irrelevant dada can be reduced during transmission from the publishers to subscribers. This paper proposes to examine this question with the aim of efficiency and scalability of the transmission on how to design a spatial-based approach which combines dynamic quad-tree and sort list, and predicate logic-based approach in large-scale analytic simulation. For analytic simulation of military operation, the interests regions of the spatial entities are formulated as circles or rectangles, and how to predict the entry time and exit time of the regions is studied in detail.**

Keywords-interest management; dynamic quadtree; sort-based; predicate logic; dynamic intersecting test

I. INTRODUCTION

Performance is a vital issue in analytic simulation. In large-scale analytic simulation, the simulation time of the majority of entities are advanced in the forms of event-driven. Sometimes even optimistic approach to synchronize time is adopted. As the entities may be deployed on multiple nodes of cluster in large-scale simulation, the communication between entities will be frequent. This will greatly influence the performance of the system. For an entity in the real system, the data need to be processed according to the requirements and sense area. Information it did not care about or out of reach will be ignored or unknown. Thus how to filter unrelated data transmission and reception to efficiently exchange information between entities is an issue to study in large-scale analytic simulation.

An efficient solution for this problem is to reduce the sending and receiving of irrelevant data, and only data that is necessary for the entities are transmitted to them. For example, when entity A moves from P1 to P2, the position of P2 is updated, and only sent to the entities that can “see” it and are concerned on it. Interest management has provided ability to filter irrelevant data in this way. Publishers register their region of possible varying range and Subscribers register their interest region to interest manager. When an entity updates its attribute value to interest manager, only the entities that interested in it will receive the value from interest manager.

There are four main parts which are responsible for interest management in HLA standards[1]: DM(Declare Management), DDM(Data Distribution Management), OwnM(Ownership Management) and SURR(Smart Update Rate Reduction). Among them, DM provides Class-based data filter scheme, DDM refines it based on value at instance level, OwnM provides the ability to change the producer of the value, and SURR adjust the rate of data updates.

This paper is organized as follows: Section II analyzes some major techniques of interest management; Section III proposes our interest management for large-scale analytic simulation. Section IV gives a briefly introduction to design and implementation of the interest management framework. Section V is about the conclusions and future works.

II. RELATED WORKS

There are many techniques utilized for interest management to transmit information efficiently among entities. The two main types are spatial-based and expression/predicate-based.

A. Spatial-based interest management

Spatial-based interest management provides data filtering by describe and register interest as regions. It is based on the fact that two areas of interests intersect only if the regions overlap in the space. The whole space which contains all of the possible regions is called routing space. In the space for military operation, the entities which register their interests in some areas are normally sensors. These areas can be circles or rectangles, which depends on the type of sensors. Interest management for sensors is responsible for specifying when they will “see” each other, and what they will “see”.

The main idea to filter irrelevant data between senders and receivers is to establish connection when the update region (for senders) is overlapped by subscribe region (for receivers). This kind of interest management is implemented in the High Level Architecture (HLA) by DDM (Data Distribution Management) service.

The simplest approach to detect region overlap is to calculate the intersection between region pairs of publisher and subscriber one by one, which is matching all of the publishing regions and subscribing regions. This approach is also called region-based approach. Once the intersection of P and S1 is none zero, a connection will be built and the data of P will be transferred to S1 when the data is updated. Although the algorithm is easy to be implemented, the compute complexity is too large and many unnecessary matches exist especially in large-scale environment.

To reduce the matching between two regions explicitly far-away from each other, the grid-based approach is introduced by dividing the routing space into grid cells. A region belongs to a grid cell if the region overlaps the cell. Two regions are not overlapped when the grids they belong to are totally different. Optimizing the size of the cell is crucial in improving the performance, because large cells lead to large amount unnecessary data transmitted, while small cells lead to large amount work to refresh the overlapping list when the entities moves frequently from one cell to another[2]. So it is difficult to choose the cell-size for the fixed grid-based method if the region sizes of entities are varying greatly or regions are very different in size. Such problem can be addressed by using multi-resolution grids, e.g. HiGrids[3], Quadtree[4], etc. These structures are available for dynamically moving entity but the "constructing cost" is still left to be solved.

Another approach is to maintain the publishing regions and subscribing regions in a sorted list[5]. All of the extents are listed in some order, and the extents can be efficiently resorted by "bubble over" or other resort methods when the entity is moving around[6]. This is particularly useful when the entities are in larges spatial environment and most of them are stationary[7]. However, the resort will be expensive when most of the entities are moving and the original sorting process may be very costly if comparing a lot of entities that are not near-by.

There are some hybrid approaches which are combined with some basic approaches mentioned above, such as a hybrid approach based on both the region-based and grid-based approaches proposed by Gary Tan [8], and an adaptive DDM[9]. The goal of these approaches is to refine one method by another method to make comprise between the advantages and disadvantages of the two methods.

B. *Predictate logic-based interest management*

An alternative interest management is to express the interest in terms of expression/predicate. It is a powerful method which can express all most all kinds of interest by connecting the conditions using "AND", "OR", "NOR", etc. Although the matching process is hard to be improved, the predicate logic-based approach is convenient to describe the area of interests that are in logic. For example, the commander in the base register interest in the airplanes in combat that is under control of but is far away from it geographically.

III. OUR PROPOSED INTEREST MANAGEMENT

In this section, we will propose an adaptive interest management framework for varies types of entities in large-scale analytic simulation.

The use cases can be divided into two kinds: by geography and by logic (see Figure 1). Different methods can be applied to them. The former includes the cases that entities such as Radar obtain attributions like position or velocity of other entities such as airplanes in their detection range which can be statically or dynamically geography. And the spatial-based interest management is utilized which will be shown in the following part. The latter one takes apart regions of interest that are at a distance. It includes the entities connected tightly by "bridge" in logic, such as communication between two radio stations. And predicate logic-based interest management is used to build some major communication network among entities in different battle side. The division of the two catalogs makes it possibly to efficiently apply both methods.

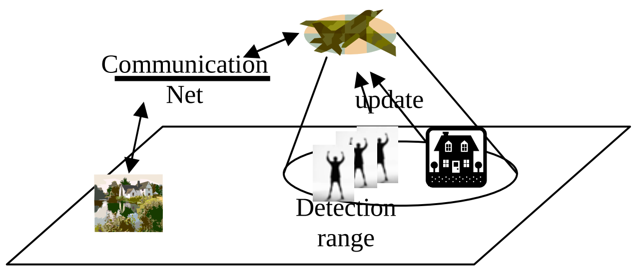

Figure 1. Two kinds of interest

A. *A hybrid spatial-based approach*

Considering the reality that there are huge entities simulated in our system, and most of them are moving ones, the traditional approaches are not suitable. The multi-resolution grid reduces matching of regions in different cells but still faces "list cost", and the sort-based approach can reduce rematch by the use of history information[6] but remains "unnecessary cost". So, there might be a combination of them. This is similar to the hybrid approach mention in [4].

Firstly, instead of using quadtree-based approach to perform spatial partition, the space is divided dynamically into grid cells according to the scales of entities by a dynamic quadtree, with the branching condition that there are $k>m$ entities in the grid cell and end-branching condition that the branch level satisfies $l>n$ or the number of entities in the cell satisfies $k<m$.

Secondly, a sorted-based approach is used for each leaf in the quadtree.

In the spatial partition stage, the node in the quadtree can be combined dynamically when the amount of entities is less than the threshold of m. The construction of the quadtree is shown in Figure 2.

Each leaf node in the tree represents a grid cell in the space, and the sizes of cells are not uniformed with a fix amount of regions in it. The parent node represents the combination of the child nodes, and will have a lower resolution.

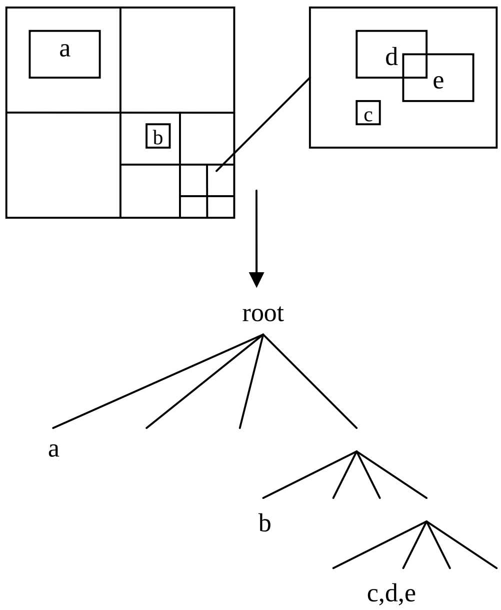

Figure 2. Construction of quadtree

During the sort stage, n lists are maintained at every leaf node for n-dimension space, e.g. n is 2 in the scenario shown in Figure 1. Every list for the same node stands for different dimension and is dependent from each other. The lists are composed of the endpoints of the regions in all of the dimensions, and are sorted in ascending order. An example of sorts of 3 regions in one-dimension is shown in Figure 3.

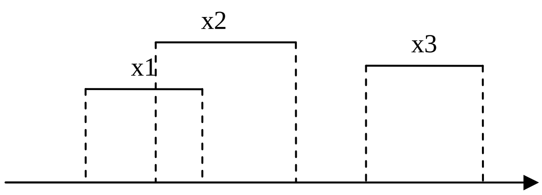

Figure 3. Sorts of list for the region sorted in one dimension

As the sizes of areas of interest may be quite different, there will be large region which represents interest of large sensor range. These regions may overlap multiple grid cells and may move fast among the cells, so they may not be contained in a particular cell but be related to multiple cells. Then we can refer to some higher-level node where such entities move around. Taking an aircraft as an example, the region is varying at the third level and the list contained in node 1 (see Figure) will be inserted into the lists of child nodes when calculating the overlaps.

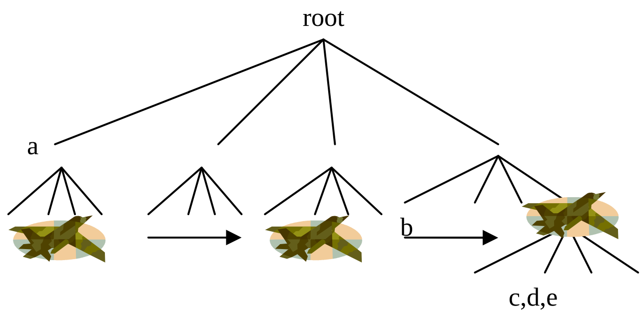

Figure 4. List in different level of tree

Then the computation will be reduced by simply adding the list in the small cells. And our proposed algorithm will have better computation performance compared with other algorithms in this situation.

B. Hierarchical communication net

In SURR mechanism in HLAE, the data is published at a certain frequency, the subscribers provide their requirements for this data in the FOM with a value referred to the frequency which is less than the publishing frequency[10]. During the runtime, different update rates will be sent at different rate possibly using multicast implementations. This work has solved the problem of adapting the rate according to the various subscribed rate.

In our research, the entities which are related to each other logically are divided into a hierarchical tree. The communication nets between nodes in some sub-tree share a communication net. Several communication nets are connected through the backbone net. The basic communication network is shown in Figure 5.

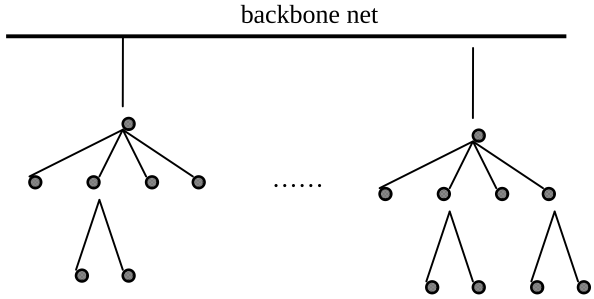

Figure 5. Basic logic communication network

IV. Design and Implementation

There are three major objects in the interest management framework: Interest Manager (IM), Region Manager (RM), and Logic Comm Manager (LCM) .IM is responsible for coordinating the other two objects and predicting when one entity might “see” another entity (see Figure).

Interest Manager
- Region Manager
- Logic Comm Manager

Figure 6. Relationship of three major objects

The RM is responsible for matching regions in the space, including dividing the space into dynamically quadtree and sort and resort of the list. The details of the algorithms will be introduced in the following part. The LCM majors in matching expressions of predicate logic, creating and maintaining communication net according to the logic communication network.

Major algorithms are described by pseudo code as follows:

```
void ResortWhenMovesHigh (Node *pOrigNode, Node *pNewNode, Object *pObj)
{DELETE  pObj FROM pOrigNode->pObjList;
 pNodeList1 = GetLeafNode(pOrigNode);
 pNodeList2 = GetLeafNode(pNewNode);
 IF (IsContainedInArea( pObj, GetRangeOfNode(pNodeList1[i]))==FALSE)
    {DELETE pObj->x, pObj->y FROM pNodeList1[i]->pSortList;
     DELETE pObj FROM pNodeList[i]->pObjList;
    }
  INSERT pObj INTO pNodeList2[i]->pObjList;
  DO ResortList(pObj->x, pObj->y, pNodeList2[i]->pSortList[k]);
}

void MergeChildRegion(Node *pParentNode)
{pNodeList=GetLeafNode(pParentNode);
 pParentNode->pChild[i]=NULL;
 AddObj(pObjList, pNodeList[i]->pObjList);
 CombineSortList(pNodeList[1]->pSortList[k], …, pNodeList[n]->pSortlist[k]);
}

void CombineSortList(pSortList1, pSortList2)
{LengthOf(pSortList1)> LengthOf (pSortList2);
 n=1;
 l=0;//l is the start point to insert a value into a list;
  l=ResortList(pSortList1, pSortList2[0], l);
  WHILE(l<GetLength(pSortlist1) AND n<GetLength(pSortList2))
   DO(l=ResortList(pSortList1, pSortList2[n],l);n++);
}
```

Where, the node of the tree is defined as the following structure:

```
Struct Node {
 Node *pChild[4];
  Object *pObjlist;
  SortList *pSortList[2];
}
```

V. DYNAMIC INTERSECTING TEST FOR REGIONS

As described in the above section, interest management in the 3D space is based on representation of spatial region. For example, sensors register their interest in the forms of detection area, while entities in the operation space are expressed as the shapes of the platform where they reside in. For a sensor in an aircraft, the shape of the sensor is equal to the shape of the aircraft which can be abstract to rectangle to simplify computation.

So far, the discussion has involved the static regions and one moving region. However, what should we do when two or more regions are moving? This section concentrates on testing intersects for several dynamically moving regions. It aims to predict when a shape will contact with another shape at the first time and the last time, which means entry time and exit time.

In the analytic simulation for military operation, the regions are usually expressed as rectangle and circle. The problem becomes testing two shapes of region whether or not the shapes intersect with each other and when will they intersect. Due to the relativity of movement, one of the regions can be set as stationary, while the other one moves at the relative speed. And the cases are classified by rectangle vs. rectangle, circle vs. rectangle, and circle vs. circle.

Firstly, let’s consider the case that two rectangles are moving. Assuming that region A and region B are moving from position PA1 $(x_{A1}, x_{A2}, y_{A1}, y_{A2})$ and PB1 $(x_{B1}, x_{B2}, y_{B1}, y_{B2})$ at the speed of VA and VB separately. The problem can be solved at each dimension such as X-axis and Y-axis by orthogonal decomposition, as shown in Figure 7.

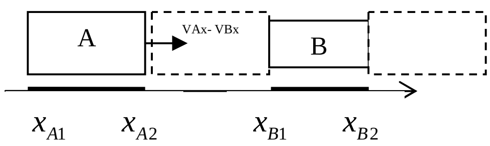

Figure 7. Moving rectangles at X-axis

Published by Atlantis Press, Paris, France.

According to relativity theory, region A moves against region B with relative velocity of VAx-VBx. Then the entry time and exit time can be got as:

$$t_{entry} = (x_{B1} - x_{A2})/(v_{Ax} - v_{Bx})$$

$$t_{exit} = (x_{B2} - x_{A1})/(v_{Ax} - v_{Bx})$$

Secondly, in the case that a circle and a rectangle move around, it is computed by setting that the rectangle is immobile. It is illustrated in Figure 7 that circle A with radius r is shifting from position PA1 (x_A, y_A), which is specified by the center-point of the circle, at the speed of VA. The rectangle is expressed in the forms of PB1 $(x_{B1}, x_{B2}, y_{B1}, y_{B2})$.

The cases that the circle and the rectangle intersect are shown in Figure 8-9. The circle overlaps the rectangle with two kinds of critical condition. One is that the edge of the rectangle is the tangency of the circle, then the rectangle B is expanded to rectangle C by r. The other one is that the circle is intersect with B at the vertex, then the vertexes of B is wrapped by four quarter circles, which is shown in Figure 8. The quarter circle is specified by the center-point which is one of the vertexes and the radius of r. According to the relativity, the problem can be abstract to the movement of the center-point of A by expanding the rectangle.

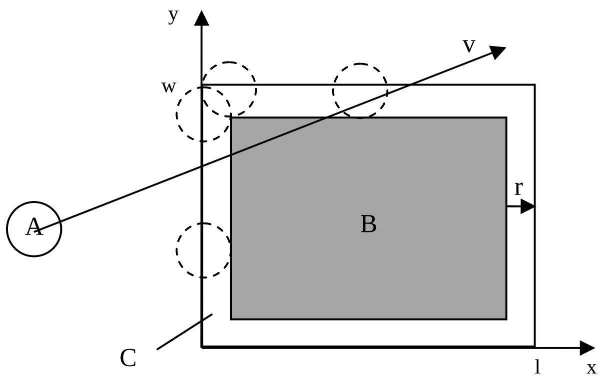

Figure 8. Point CA moves across rectangle C which is expanded from B

In the first case illustrated in Figure 8, the center-point of A is moving through the expanded rectangle C. The entry time and exit time are the intersecting time with C if and only if they are not intersecting with vertexes shown in Figure 8.

Let: $v = (v_x, v_y) = v_A - v_B$,

$S = (x_S, y_S) = (x_A - x_{B1} + r, y_A - y_{B1} + r)$

The length and width of C are:

$l = (x_{B2} - x_{B1} + 2r)$, $w = (y_{B2} - y_{B1} + 2r)$

Then, the point of intersecting P = $(x_p, y_p) = (x_S + v_x \cdot t, x_y + v_y \cdot t)$ satisfies:

$$\begin{cases} x_p = 0 \\ 0 \le y_p \le w \end{cases}, \text{ or } \begin{cases} x_p = l \\ 0 \le y_p \le w \end{cases}, \text{ or } \begin{cases} 0 \le x_p \le l \\ y_p = 0 \end{cases}, \text{ or } \begin{cases} 0 \le x_p \le l \\ y_p = w \end{cases}$$

where t is the intersecting time.

A is not intersected with B at any vertexes if :

$$\begin{cases} r < x_p < w + r \\ r < y_p < l + r \end{cases}$$

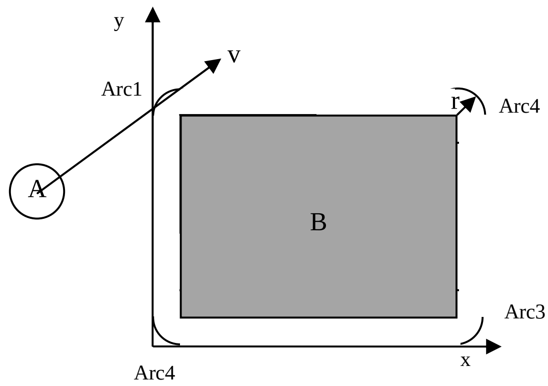

Figure 9. Point CA moves across one of the four arcs

Otherwise, it is test whether A intersects with B at one of the vertexes, which means the center-point of A moves through one of the four quarter-circles (which are specified by Arc1-Arc4 in Figure 9). The intersecting arc can be further specified by judging the range of x_p and y_p. For example, when the intersecting arc is Arc4, P = $(x_p', y_p') = (x_S + v_x \cdot t', x_y + v_y \cdot t')$ satisfies:

$$\begin{cases} 0 \le x_p' \le r \\ 0 \le y_p' \le r \\ \sqrt{(x_p' - r)^2 + (y_p' - r)^2} = r \end{cases}$$

Where, t' is the intersecting time.

Lastly, two circles (One is A at CA with a radius of r_A, while the other one is B at CB with a radius of r_B) intersects with each other, only when the distance of the center-points is no more than the trigger distance of $r_A + r_B$. The coordinates of CA and CB are (x_A , y_A),and

(x_B , y_B).The velocities of the circle are v_A and v_B separately. The relative velocity of A is $v_A - v_B$ when the reference object is B, and the problem can be equivalent to a point moves to a circle of a radius of $r_A + r_B$, as shown in Figure 10.

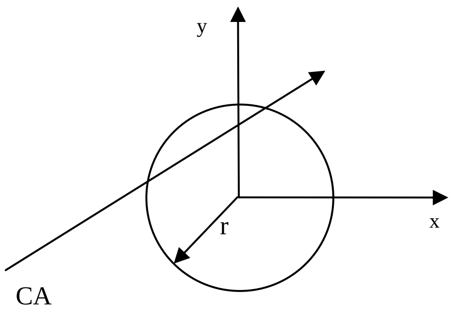

Figure 10. Point CA moves across a circle

Let: $v = v_A - v_B$, $tD = r = r_A + r_B$,

$$s_0 = (\Delta x, \Delta y) = (x_A - x_B, y_A - y_B),$$

$$\Delta p = (\Delta p_x, \Delta p_y) = (\Delta x - r, \Delta y - r)$$

As the critical condition for intersecting of the point and the circle is the distance between CA and CB is tD, we get:

$$s_0 + v \cdot t = r$$

To avoid finding square root of a negative, both sides of the equal sign are squared.

$$(s_0 + v \cdot t)^2 = r^2$$

Then, $t = \dfrac{-b \pm \sqrt{k}}{2a}$

where,

$$a = (\Delta v_x)^2 + (\Delta v_y)^2, \quad b = 2(\Delta p_x \cdot \Delta v_x + \Delta p_y \cdot \Delta v_y),$$

$$c = (\Delta p_x)^2 + (\Delta p_y)^2 - r^2, \quad k = b^2 - 4a \cdot c$$

In the complementation for these equations in a function, the value of k should be tested whether it is positive before return the entry time $t_{entry} = \min(t)$ and exit time $t_{exit} = \max(t)$.

VI. Conclusion

In this research, the problem of interest management is divided into geographical and logic parts, and a hybrid spatial-based approach and a predicate logic-based approach are utilized to describe and match interest region respectively. The hybrid approach adopts dynamic quad-tree to partition spaces and some sorted lists are assigned to nodes in the tree in order to refine matching process further. The interests expressed in predicate logic are matched through logic communication net with smart update rate. After the intersecting area is computed, the entry time and exit time for the moving object are predicted using the geometry methods.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NSF) Grant #61074108.

References

[1] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules," ed, 2010, p. 38.

[2] R. Ayani, et al., "Optimizing Cell-size in Grid-Based DDM," in Proceedings of 33rd Annual Simulation Symposium, Washington, U.S.A., 2000.

[3] L. Craig and S. Jeff, "JSIMS CCSE Performance Analysis," in In proceedings of the Fall 2004 Simulation Interoperability Workshop, 2004.

[4] O. Eroglu, et al., "Quadtree-Based Approach to Data Distribution Management for Distributed Simulations," presented at the SpringSim, 2008.

[5] C. RACZY, et al., "A Sort-Based DDM Matching Algorithm for HLA," ACM Transactions on Modeling and Computer Simulation, vol. 15, pp. 14-38, January 2005.

[6] Y. Yao, et al., "Moving-Intersecting-Information-Based Matching Algorithm for Dynamic DDM," in Proceedings of the 2009 Summer Computer Simulation Conference, Istanbul, Turkey, 2009.

[7] K. Pan, et al., "An Efficient Sort-Based DDM Matching Algorithm for HLA Applications with a Large Spatial Environment," 21st International Workshop on Principles of Advanced and Distributed Simulation (PADS'07), 2007.

[8] Y. Z. a. R. A. Gary Tan, "A Hybrid Approach to Data Distribution Management," presented at the proceedings of 4th International Workshop on Distributed Simulation and Real-time Applications (DSRT'00), 2000.

[9] C. Raczy, et al., "Adaptive Data Distribution Management for HLA RTI," presented at the European Simulation Interoperability Workshop, 2002.

[10] B. Möller and M. Karlsson, "Developing Well-Balanced Federations Using the HLA Evolved Smart Update Rate Reduction."