

# High Level Architecture

## Module 2

### Advanced Topics

---



Roy Crosbie  
John Zenor  
Simon Goberstein

California State University, Chico

# **High Level Architecture**

## **Module 2**

### **Advanced Topics**

---

## **Data Distribution Management 2: Physically Correct Filtering**



California State University, Chico

11/4/99

2

# **Data Distribution Management (DDM):**

## **2. Physically Correct Filtering**

---

- This lesson and the previous one are based on K. L. Morse and J. S. Steinman's paper "Data Distribution Management in the HLA", 1997 Spring Simulation Interoperability Workshop



## The Need for Developing Physically Correct Filter Strategies

---

- Factors that affect filtering:
  - Network latencies restrict how tightly in real time one federate can interact with another
  - Some routing spaces have *dynamic coordinates* (those changing continuously over time) for describing update and/or subscription regions
  - Other kinds of routing space coordinates, the so-called *discrete coordinates*, change their values unpredictably
- In this lesson we discuss strategies for overcoming problems faced by DDM in real-time execution



California State University, Chico

11/4/99

4

1) Filtering strategies must sample dynamically changing routing space coordinates at rates that are not too high. Otherwise, the filter strategy might require too many computations and changes in the filter specifications. On the other hand, if the (changing) routing space coordinates are not sampled often enough, the filter will not be as responsive or as efficient as it otherwise could be.

2) An efficient filtering strategy must not sample subscription regions too often or too infrequently for the same reasons. The filter strategy used by a federation must take into account the fact that true update regions and true subscription regions are not accurately represented by the sampled regions (which are represented as static regions at discrete points in time). To get around this problem one might widen subscription regions and/or update regions in a physically correct manner (of course, this must be done in a way that preserves true overlaps between the update and subscription regions).

3) In the case of *discrete coordinates*, the routing space coordinate values remain constant unless they are changed to new values at discrete points in time. In some cases, the values never change. In other cases, they may change abruptly at any time without prior warning. In short, *they do not change smoothly in a continuous manner*. For these kinds of routing space coordinates, there may be no way to predict when their values will change. Filter strategies, using discrete coordinates in multidimensional routing spaces, will discover objects late in simulation time due to network latencies when changes are made to a routing space region.

## Filtering on Discrete Coordinates

---

- Discrete coordinates make large jumps when they change; they can be thought of as step functions
- If changes to discrete coordinates can be predicted, the update and subscription regions can provide physically correct filtering by
  - Adding a new extent indicating the predicted change
  - Removing the old extent when it is no longer valid
- If it is not possible to predict when the values of discrete coordinates change, then the filtering will still be logically correct but not physically correct



California State University, Chico

11/4/99

5

Filtering on discrete coordinates can be very useful in describing categories of objects to which a federate might want to subscribe. For example, a federate might want to subscribe to objects that are radar-detectable. This could be one of the dimensions in a routing space. Radar detectability could be a dimension with two bins (True or False) which are set during initialization and never change. This kind of coordinate dimension can be useful in specifying the static characteristics of objects beyond just their class type, which is the most basic type of filtering that is provided by the DM services.

## Filtering on Dynamic Coordinates

---

- Dynamically changing coordinates, unlike discrete ones, are represented not as step functions but as continuous curves that smoothly change over time
- It is assumed that:
  - At any point in time the equations can change form or characteristic parameters
  - One can bound how much a dynamically changing routing space coordinate can vary over a specified time which makes it possible to establish efficient filtering practices that also provide physically correct filtering



California State University, Chico

11/4/99

6

It is assumed that the continuous curves representing dynamically changing routing space coordinates can be evolved over time in software according to known equations, integrals, etc.

The RTI does not automate filtering based on known functions. It is responsibility of the federates to change routing space regions when coordinate values in the routing space change noticeably.

An assumption about the possibility of bounding how much a routing space coordinate value can change during a specified time means the following. If a routing space coordinate has a value  $v$  at time  $t$ , then it can change by at most  $\Delta v$  at time  $t + \Delta t$ . To define filter specifications correctly with latency, it may also be important to bound the worst case  $\Delta v$  for *any* object in the federation. This has to be done for each dynamically changing dimension coordinate in the routing space.

## On Efficient Filtering Strategies

- The key questions for developing efficient filtering strategies are:
  - How much to artificially widen the subscription regions of dynamically changing routing space coordinates?
  - How often to sample the interest regions and the related update regions?
- Considering at first only the sampling of update and subscription regions, we must recognize that:
  - Subscription regions are expanded to always include their true region of interest within their sampling period
  - Update regions can be widened or subscription regions of other federates may have to be further expanded



California State University, Chico

11/4/99

7

1) It is important to recognize that, in practice, one would never want to sample an object's routing space coordinates independently but rather their values would always be sampled together in a coordinated manner. For the sake of clarity, our current discussion focuses on a single routing space coordinate and how to determine its natural sampling rate.

2) Federates need to take all their routing space dimensions into account when determining their sampling rates. Federates may further coordinate this process so that all of their local objects modify their update regions together with subscription regions.

3) Update regions can be widened or the subscription regions of other federates may have to be further expanded to account for the fact that update regions, represented as points in the routing space, are also sampled.

## Sampling Subscription Regions

---

- Suppose a federate's true subscription region is specified by a low and high value,  $[v_{lo}, v_{hi}]$
- When that subscription region is sampled  $t$  time units later, the old high and low values might have changed by as much as  $\Delta v$ , so in order to keep the specified regions valid until the next sampling time, ***we must extend the subscription region by  $\Delta v$  on both ends***, that is, define the extents of that region as  $[v_{lo} - \Delta v, v_{hi} + \Delta v]$



1) Of course, it might happen that  $\Delta v$  for the low end is different from  $\Delta v$  for high end of a filter specification but, for simplicity, we are assuming here that they are the same.

2) It is important to recognize that we could specify either a value for  $\Delta v$  which then dictates our choice for  $t$  or a value for  $t$  which then dictates our choice for  $\Delta v$ . What is the best approach? We will address that question in the next slide.



## Defining $\Delta v$ and $\Delta t$

- To make sure that federates are not required to redefine their subscription regions too frequently (which is impractical), we should set  $t_{min}$  as the minimum sensible time between filter updates
- Once  $t_{min}$  is defined, we can determine  $v_{min}$
- If it turns out that  $v_{min}$  is much smaller than what makes sense, we might have to increase it to some naturally determined value  $v$  and, if necessary, change  $t_{min}$  to another appropriate value  $t$



1) It is possible for the expanded subscription region to be much larger than the true region of interest. This is the tradeoff that one has to make concerning filter overhead and the benefits of filtering itself.

2) Concerning our remark that  $v_{min}$  might turn out to be much smaller than what makes sense, suppose, for example, that DDM is implemented using grid cells and their resolutions are much larger than  $v_{min}$ . Then without losing much efficiency (perhaps none at all!), we should be able to extend  $v_{min}$  further to  $v_{cell}$  where  $v_{cell}$  is naturally related to the cell size for that routing space. It must be remembered that the subscription regions include any cells that they might overlap in the routing space so it does not make sense to define subscription regions that are much smaller than cell sizes.

## Sampling Update Regions

---

- How often should we sample an update region?
  - If it is sampled too often, filtering becomes inefficient
  - If it isn't sampled often enough, filtering is ineffective
- It should be kept in mind that
  - It is possible for an object's routing space position to be just outside a federate's subscription region and, as its true physical state changes, it actually moves into the object's subscription region before the next sampling
  - Thus either the object's position has to be represented as an extended region or the subscription regions of other federates have to be further widened



## Defining $\delta v$ and $\delta t$

---

- Suppose the maximum change of the coordinate value in time  $t$  is  $v$ 
  - If update regions are represented as extended regions, the strategy for defining  $t$  and  $v$  is essentially the same as for  $t$  and  $v$  for subscription regions
  - If subscription regions are expanded to account for the sampling of unextended update regions, subscription regions must be further expanded by a known amount which limits the choices of how to define  $t$  and  $v$



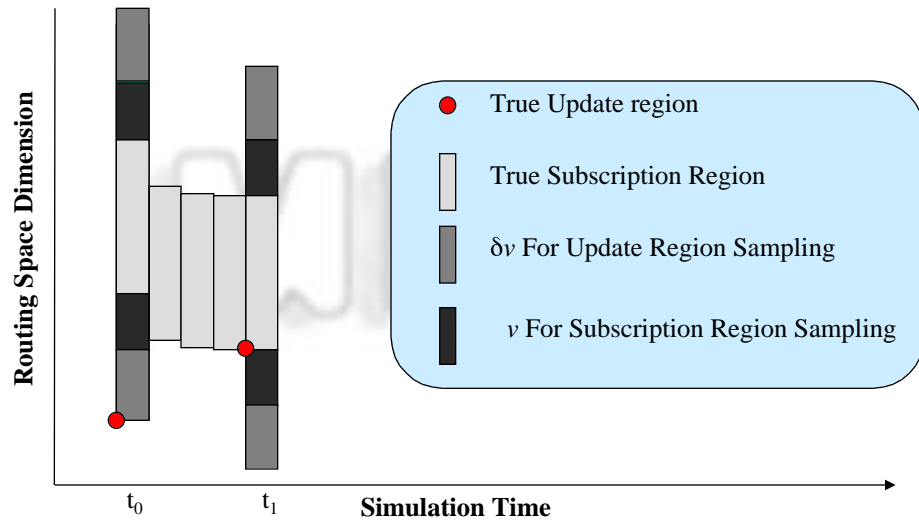
California State University, Chico

11/4/99

11

If the DDM implementation uses grid cells, what makes the most sense is to choose  $v$  to be related to the cell size and then let  $t$  be whatever it needs to be. If  $t$  is too small, then it might make sense to make it larger and to use an extended update region instead of representing the object as a point in order to make up the difference.

## An Example



California State University, Chico

11/4/99

12

This example illustrates an artificial expansion of a federate's subscription region based on its true subscription region and on update region sampling. Here we assume that at time  $t_0$ , a federate's subscription region is specified as its true region expanded by  $\delta v$  and  $v$ . At time  $t_1$ , the true region has changed by  $v$  and an update region has moved an amount  $\delta v$  towards the federate's subscription region. If the object's update region value was just inside the expanded subscription region at time  $t_0$ , then it will be discovered before it actually enters the federate's true subscription region just before  $t_1$ . Our scheme does not take into account latency affects.

## Accounting for Latencies

---

- As was pointed out earlier, the discovery process takes a certain amount of time
- During the time  $L_{tot}$  (= the total latency required for object discovery), dynamically changing routing space coordinates may vary enough to warrant object discovery and thus
  - It is necessary that update and subscription regions be artificially widened again to account for the dynamic changes possible between an object's subscription region and any other object's update region



## Multiple Dynamical Coordinates

---

- A routing space may have in its definition many dynamically changing coordinates which can be thought of as a vector,  $V$
- Updating routing space regions for the components of vector  $V$  should be done together instead of each dimension scheduling its updates independently, so
  - Subscription regions must be sampled in  $t$  time units
  - Update regions should be sampled every  $t$  time units
  - Due to the finite sampling rates and network latencies, regions must be increased accordingly



California State University, Chico

11/4/99

14

There is no reason for subscription region sampling rates and update region sampling rates to be the same. In fact, it is possible for each object to define its own sampling rates based on their characteristics. For example, slow moving objects may not have to sample as often as fast moving ones.