Department of Defense

High Level Architecture

Interface Specification

Version 1.1

12 February 1997

TABLE OF CONTENTS

1 GENERAL....4

1.1 PURPOSE....4
1.2 HLA FEDERATION OBJECT MODEL FRAMEWORK....4
1.3 GENERAL NOMENCLATURE AND CONVENTIONS....5
1.4 ORGANIZATION OF THIS DOCUMENT....6

2 FEDERATION MANAGEMENT....8

2.1 CREATE FEDERATION EXECUTION....10
2.2 DESTROY FEDERATION EXECUTION....11
2.3 JOIN FEDERATION EXECUTION....12
2.4 RESIGN FEDERATION EXECUTION....13
2.5 REQUEST PAUSE....15
2.6 INITIATE PAUSE †....16
2.7 PAUSE ACHIEVED....17
2.8 REQUEST RESUME....18
2.9 INITIATE RESUME †....19
2.10 RESUME ACHIEVED....20
2.11 REQUEST FEDERATION SAVE....21
2.12 INITIATE FEDERATE SAVE †....23
2.13 FEDERATE SAVE BEGUN....24
2.14 FEDERATE SAVE ACHIEVED....25
2.15 REQUEST RESTORE....26
2.16 INITIATE RESTORE †....27
2.17 RESTORE ACHIEVED....28

3 DECLARATION MANAGEMENT....29

3.1 PUBLISH OBJECT CLASS....30
3.2 PUBLISH INTERACTION CLASS....32
3.3 SUBSCRIBE OBJECT CLASS ATTRIBUTE....33
3.4 SUBSCRIBE INTERACTION CLASS....35
3.5 CONTROL UPDATES †....36
3.6 CONTROL INTERACTIONS †....37

4 OBJECT MANAGEMENT....38

4.1 REQUEST ID....39
4.2 REGISTER OBJECT....40
4.3 UPDATE ATTRIBUTE VALUES....41
4.4 DISCOVER OBJECT †....42
4.5 REFLECT ATTRIBUTE VALUES †....44
4.6 SEND INTERACTION....45
4.7 RECEIVE INTERACTION †....47
4.8 DELETE OBJECT....48
4.9 REMOVE OBJECT †....49
4.10 CHANGE ATTRIBUTE TRANSPORTATION TYPE....50
4.11 CHANGE ATTRIBUTE ORDER TYPE....51
4.12 CHANGE INTERACTION TRANSPORTATION TYPE....52
4.13 CHANGE INTERACTION ORDER TYPE....53
4.14 REQUEST ATTRIBUTE VALUE UPDATE....54
4.15 PROVIDE ATTRIBUTE VALUE UPDATE†....55
4.16 RETRACT....56

4.17 REFLECT RETRACTION †....57

5 OWNERSHIP MANAGEMENT....58

5.1 REQUEST ATTRIBUTE OWNERSHIP DIVESTITURE....61
5.2 REQUEST ATTRIBUTE OWNERSHIP ASSUMPTION †....63
5.3 ATTRIBUTE OWNERSHIP DIVESTITURE NOTIFICATION †....64
5.4 ATTRIBUTE OWNERSHIP ACQUISITION NOTIFICATION †....65
5.5 REQUEST ATTRIBUTE OWNERSHIP ACQUISITION....66
5.6 REQUEST ATTRIBUTE OWNERSHIP RELEASE †....68
5.7 QUERY ATTRIBUTE OWNERSHIP....69
5.8 INFORM ATTRIBUTE OWNERSHIP †....70
5.9 IS ATTRIBUTE OWNED BY FEDERATE....71

6 TIME MANAGEMENT....71

6.1 REQUEST FEDERATION TIME....73
6.2 REQUEST LBTS....74
6.3 REQUEST FEDERATE TIME....75
6.4 REQUEST MINIMUM NEXT EVENT TIME....76
6.5 SET LOOKAHEAD....77
6.6 REQUEST LOOKAHEAD....78
6.7 TIME ADVANCE REQUEST....79
6.8 NEXT EVENT REQUEST....80
6.9 FLUSH QUEUE REQUEST....82
6.10 TIME ADVANCE GRANT †....83

7 DATA DISTRIBUTION MANAGEMENT....84

7.1 CREATE UPDATE REGION....86
7.2 CREATE SUBSCRIPTION REGION....87
7.3 ASSOCIATE UPDATE REGION....88
7.4 CHANGE THRESHOLDS †....90
7.5 MODIFY REGION....91
7.6 DELETE REGION....92

8 HLA IDL APPLICATION PROGRAMMER'S INTERFACE....93

9 HLA C++ APPLICATION PROGRAMMER'S INTERFACE....116

10 HLA ADA 95 APPLICATION PROGRAMMER'S INTERFACE....146

11 REFERENCES....180

TABLE OF FIGURES

FIGURE 1. BASIC STATES OF THE FEDERATION EXECUTION....8
FIGURE 2. OVERALL VIEW OF FEDERATE TO RTI RELATIONSHIP....9
FIGURE 3. FEDERATE DIVESTING ATTRIBUTE OWNERSHIP (NEGOTIATED)....59
FIGURE 4. FEDERATE ACQUIRING ATTRIBUTE OWNERSHIP....60
FIGURE 5. ROUTING SPACE EXAMPLE....85

1 General

The High Level Architecture (HLA) Interface Specification is one of the three HLA definition documents. This and the other two HLA definition documents, the HLA Object Model Template and the HLA Rules, along with supporting documents, are in the HLA Technical Library on the DMSO homepage (http://www.dmso.mil).

1.1 Purpose

This document provides a specification for the DoD High Level Architecture (HLA) functional interfaces between federates and the Runtime Infrastructure (RTI). The RTI provides services to federates in a way that is analogous to how a distributed operating system provides services to applications. These interfaces are arranged into the six basic RTI service groups given below:

- Federation Management
- Declaration Management
- Object Management
- Ownership Management
- Time Management
- Data Distribution Management

The six service groups describe the interface between the federates and the RTI, and the software services provided by the RTI for use by HLA federates. The initial set of these services was carefully chosen to be those functions most likely to be required across multiple federations. As a result, federate applications will require most of the services described in this document. The RTI requires a set of services from the federate that are referred to as "RTI Initiated" and are denoted with a †.

1.2 HLA Federation Object Model Framework

A concise and rigorous description of the object model framework is essential to the specification of the interface between federates and the RTI and of the RTI services. The rules and terminology used to describe a federation object model are described in the "High Level Architecture Object Model Template" [HLA OMT]. A Simulation Object Model (SOM) describes salient characteristics of a federate to aid in its reuse and other activities focused on the details of its internal operation and as such is not the concern of the RTI and its services. A Federation Object Model (FOM), on the other hand, deals with inter-federate issues and is relevant to the use of the RTI. The DoD HLA definition states that federation object models describe:

- The set of object classes chosen to represent the real world for a planned federation,

- The set of interaction classes chosen to represent the interplay among real world objects,
- The attribute and parameters of these classes,
- The level of detail at which these classes represent the real world, including all characteristics.

Every object is an instance of an object class found in the FOM. Object classes are chosen by the object model designer to facilitate a desired organizational scheme. Each object class has a set of attributes associated with it. An *attribute* is a distinct, identifiable portion of the object state. In this discussion, "attribute designator " refers to the attribute and "attribute value" refers to its contents. From the federation perspective, the set of all attribute values for a particular object completely defines the state of the object. Federates are free to associate additional state information with an object that is not communicated between federates, but this is outside the HLA federation object model purview.

Federates use the state of the objects as one of the primary means of communication. At any given time only one federate is responsible for simulating a given object attribute. That federate provides new values for that attribute to the other federates in the federation execution through the RTI services. The federate providing the new attribute values is said to be *updating* that attribute value. Federates receiving those values are said to be *reflecting* that attribute.

The privilege to update a value for an attribute is uniquely held by a single federate at any given time during a federation execution. A federate that has the privilege to update values for an attribute is said to *own* that attribute. The RTI provides services that allow federates to exchange ownership of object attributes. The federate that registers an object implicitly has the privilege to delete that object. The RTI provides services that allow federates to transfer the "privilegeToDeleteObject" attribute in the same way as other attributes.

All objects have an ID. The value of the ID is unique for each federation execution. Object IDs are dynamically generated by an RTI service or can be drawn from a pool of reserved values. These reserved values are set aside for special situations where federates must have knowledge of object IDs before a federation execution begins.

The FOM framework also allows for interaction classes for each object model. The types of interactions possible between different classes of objects, their affected attributes and the interaction parameters are specified. An interaction is an explicit action taken by an object, that can optionally be directed toward another object.

A *federation* is the combination of a particular FOM, a particular set of federates, and the RTI services. A federation is designed for a specific purpose using a commonly understood federation object model and a set of federates that can associate their individual semantics with that object model. A *federation execution* is an instance of executing the federation with a specific FOM, an RTI and using various execution details.

1.3 General Nomenclature and Conventions

There are various entities (classes, attributes, parameters, regions, federates, etc.) referenced in this document which can have the following different "views":

- name - human readable
- handle - computer manipulable

The parameters to the services described in this document will use different views of the entities depending on a particular RTI implementation. For clarity, this document refers only to a generic view, known as a “designator”, when referring to these entities.

The following sets of data are needed for the implementation of a running RTI and federation executions:

- Federation Execution Data (FED) - information derived from the FOM (class, attribute, parameter names, etc.) and used by the RTI at run-time. Each federation execution needs one. In the abstract, creation of a federation execution is simply the binding of a federation execution name to a FED. The organization of FEDs will become the subject of standardization so Object Model Development Tools can automatically generate them for any vendor's RTI,
- RTI Initialization Data (RID) - RTI vendor specific information needed to run an RTI. A RID is probably supplied when an RTI is initialized.

For all federate initiated services in this specification, except 2.1, 2.2, and 2.3, there is an implied supplied parameter which is a federation execution. For all RTI initiated services there is an implied supplied parameter which is a federate. The manner in which these parameters are actually provided to the services is RTI implementation dependent, and therefore not shown in the service descriptions.

1.4 Organization Of This Document

The six HLA service groups are specified in chapters 2 through 7. Each service is described using several components:

- **Name & Description**

  service name and narrative describing the functionality of the service
- **Service Initiator**

  indicator as to whether the service is RTI or federate initiated
- **Supplied Parameters**

  required and optional service initiator provided parameters
- **Returned Parameters**

  parameters returned by the service
- **Pre-conditions**

  conditions which must exist for the service to execute correctly
- **Post-conditions**

  conditions which will exist once the service has executed correctly

- **Exceptions**

  notifications of any irregularity which may occur during service execution

- **Related Services**

  other HLA services which are related to this service

The HLA Application Programmer's Interface (API) is given in chapter 8 in Common Object Request Broker Architecture (CORBA) Interface Definition Language (IDL) form. Chapter 9 contains the C++ API and Chapter 10 contains the Ada 95 API.

2 Federation Management

Federation Management refers to the creation, dynamic control, modification, and deletion of a federation execution.

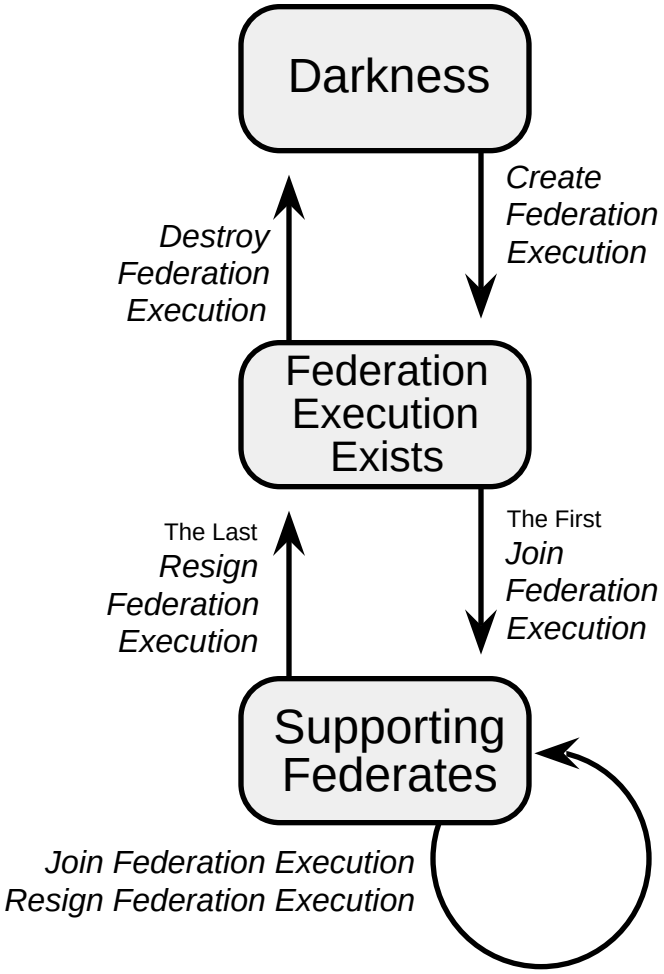

Figure 1. Basic States of the Federation Execution

Before a federate can join a federation execution, the federation execution must exist. Figure 1 above shows the overall state of a federation execution as certain basic federation management services are employed.

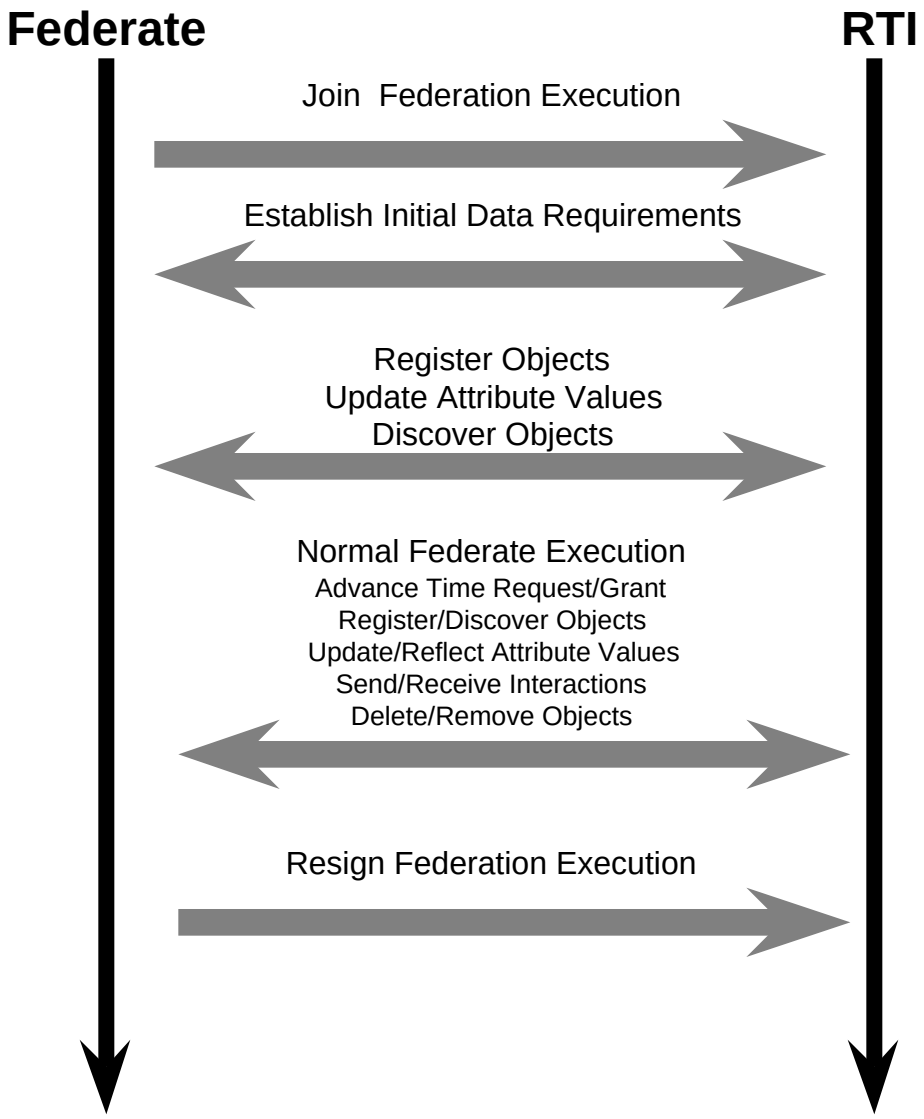

Figure 2. Overall View of Federate to RTI Relationship

Once a federation execution exists, federates can join and resign from it in any sequence that is meaningful to the federation user. Figure 2 above gives a generalized view of the basic relationship between a federate and the RTI during the federate participation in a federation execution. The arrows in Figure 2 represent the general invocation of RTI service groups and are not intended to demonstrate strict ordering requirements on the use of the services. The HLA concept does not preclude a single software system from participating in a given federation execution as multiple federates nor does it preclude a given system from participating in multiple (independent) federation executions.

2.1 Create Federation Execution

Federate initiated

The *Create Federation Execution* service creates a new federation execution, and adds it to the set of supported federation executions. The FED parameter supplies FOM and federation execution data to the RTI.

Supplied Parameters

A federation execution name

FED

- Set of all object class
- Set of attributes associated with each object class
- Set of interaction classes
- Set of interaction parameters associated with each interaction class
- The ordering and transportation service associated with each attribute of each object class
- The ordering and transportation service associated with each interaction class
- Optionally, routing spaces and the number of dimensions (see section 7)

Returned Parameters

None

Pre-conditions

The federation execution does not exist

Post-conditions

A federation execution exists with the given name that can be joined by federates

Exceptions

The federation execution already exists

Could not locate FED

Invalid FED

RTI internal error

Related Services

Destroy Federation Execution

2.2 Destroy Federation Execution

Federate initiated

This service removes a federation execution from the RTI set of supported federation executions. All federation activity should have stopped and all federates should have resigned before invoking this service.

Supplied Parameters

A federation execution name

Returned Parameters

None

Pre-conditions

There are no federates joined to this federation execution

Post-conditions

The federation execution does not exist

Exceptions

Federates are joined to the federation execution

The federation execution does not exist

RTI internal error

Related Services

Create Federation Execution

2.3 Join Federation Execution

Federate Initiated

The *Join Federation Execution* service affiliates the federate with a federation execution. Execution of the *Join Federation Execution* service indicates the intention to participate in the federation.

Disclaimer: The interaction between the *Join Federation Execution* service and Time Management is still under exploration. At the present time, there is not no temporal relationship between the federate and the federation at the completion of this service.

Supplied Parameters

A federate designator

A federation execution name

If required, connection parameters that allow the RTI and federate to communicate

Returned Parameters

A federate designator

Pre-conditions

The federation execution exists

The federate is not joined to that execution

Post-conditions

The federate is a member of the federation execution

Exceptions

Federate already joined to the federation execution

Specified federation execution does not exist

RTI internal error

Related Services

Resign Federation Execution

2.4 Resign Federation Execution

Federate initiated

The *Resign Federation Execution* service indicates the desired cessation of federation participation. Before resigning, ownership of attributes held by the federate should be resolved. The federate can transfer their ownership to other federates, release them for ownership acquisition at a later time, or delete the object to which they are attached. As a convenience to the federate, the *Resign Federation Execution* service accepts an action parameter that directs the RTI to perform zero, or more, of the following actions:

- delete all objects for which the federate holds that privilege
- release all other attributes for future ownership acquisition - this places the attributes into an unowned state (implying that their values are not being updated), which makes them eligible for ownership by another federate. See section 5 for a more detailed description.

Supplied Parameters

Directive to:

(1) release all attribute ownership

(2) delete all objects for which the federate holds delete privilege

(3) perform action (2) and then action (1)

(4) perform no actions

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

If optional directive is not supplied, the federate should not own any attributes

Post-conditions

The specified federate is not a member of the specified federation execution

There are no attributes in the federation execution owned by the specified federate

Exceptions

Federate owns attributes

Federate not a federation execution member

Specified federation execution does not exist

RTI internal error

Related Services

Join Federation Execution

2.5 Request Pause

Federate initiated

Indicates to the RTI the desire to stop the advance of the federation execution. The federation members will be instructed by the RTI to pause as soon after the invocation of the *Request Pause* service as possible. The label, supplied when the pause is requested, will be supplied to the other federates via the *Initiate Pause* service.

Supplied Parameters

A label

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation execution is advancing (not paused)

Post-conditions

A federation pause is pending

Exceptions

Federation already paused

Federate not a federation execution member

RTI internal error

Related Services

Initiate Pause

Pause Achieved

Request Resume

Initiate Resume

Resume Achieved

2.6 Initiate Pause †

RTI Initiated

Instructs the federate to stop changing state as soon as possible. The label provided to the RTI when the pause was requested, via the *Request Pause* service, will be supplied to the federate.

Supplied Parameters

The label supplied when the *Request Pause* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is not already paused

Post-conditions

The federate is informed of a pending pause

Exceptions

Federate already paused

Federate internal error

Related Services

Request Pause

Pause Achieved

2.7 Pause Achieved

Federate Initiated

Indicates that the federate has successfully stopped changing state.

Supplied Parameters

The label supplied when the *Initiate Pause* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has paused

Post-conditions

The RTI is informed that the federate is paused

Exceptions

Unknown label

No pause requested

Federate not a federation execution member

RTI internal error

Related Services

Request pause

Initiate Pause

Initiate Resume

2.8 Request Resume

Federate initiated

Indicates the desire to resume the advance of the federation execution. The federation members will be instructed by the RTI to resume the advance of their state as soon after the invocation of the *Request Resume* service as possible.

Supplied Parameters

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation execution is paused

Post-conditions

A federation resume is pending

Exceptions

Federation not paused

Federate not a federation execution member

RTI internal error

Related Services

Request Pause

Initiate Resume

Resume Achieved

2.9 Initiate Resume †

RTI Initiated

Informs a paused federate that it may return to the state evolution process in which it was engaged when it received the *Initiate Pause* service invocation. The federate should resume updating state as soon as possible.

Supplied Parameters

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is paused

Post-conditions

The federate is informed that it may resume evolving state

Exceptions

Federate not paused

Federate internal error

Related Services

Request Resume

Resume Achieved

2.10 Resume Achieved

Federate Initiated

Indicates that the federate is evolving state.

Supplied Parameters

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is evolving state

Post-conditions

The RTI has been informed that the federate is evolving state

Exceptions

No resume requested

Federate not a federation execution member

RTI internal error

Related Services

Request Resume

Initiate Resume

2.11 Request Federation Save

Federate initiated

Specifies that a federation save should take place. If the optional federation time parameter is present, the save takes place at that time. If there is no federation time parameter, the federation members will be instructed by the RTI to save as soon after the invocation of the *Request Federation Save* service as possible. The federation execution should be paused when a save occurs to help ensure consistency of the saved data among the federation participants. It is understood that the time required to perform a paused save may be unacceptable for some federations. In those cases, the federation save will be performed while the federates are executing. Only one requested/save will be outstanding at a time. A new save request replaces any outstanding save request.

Supplied Parameters

A label

Optional federation time of the desired federation save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Save not in progress

Post-conditions

A federation save has been requested for the specified federation time

All previous requested saves are canceled

Exceptions

Save in progress

Federation time has already passed

Invalid federation time

Federate not a federation execution member

RTI internal error

Related Services

Initiate Federate Save

Federate Save Begun

Federate Save Achieved

Request Restore

Request Pause

2.12 Initiate Federate Save †

RTI Initiated

Instructs the federate to save state. If the optional federation time parameter is present, the save takes place at that time If there is no federation time parameter, the federate should save as soon after the invocation of the *Initiate Federate Save* service as possible. The label provided to the RTI when the save was requested, via the *Request Federation Save* service, will be supplied to the federate. A federate can expect a *Initiate Federate Save* invocation anytime it would expect a *Time Advance Grant* invocation. See the *Request Federation Save* service description for a discussion of the interaction of save and pause.

Supplied Parameters

The label supplied when the *Request Federation Save* service was invoked

Optional federation time to associate with the save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

A federation save has been scheduled for time t and the RTI has determined that it is safe for the federate to advance to t

Post-conditions

The federate has been notified to begin saving its state

Exceptions

Invalid federation time

Unable to perform save

Federate internal error

Related Services

Request Federation Save

Federate Save Begun

Federate Save Achieved

2.13 Federate Save Begun

Federate Initiated

Tells the RTI that the federate is beginning to save its state. This notification allows the RTI to perform any necessary state save actions. If the federate received a federation time parameter with the corresponding *Initiate Federate Save* service invocation, that federation time should be a parameter to the *Federate Save Begun* call so the service can confirm that the correct save is being taken.

Supplied Parameters

Optional federation time associated with the save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has received an *Initiate Federate Save* invocation

The federate is ready to start saving its state

Post-conditions

The RTI has been informed that the federate has begun saving its state

Exceptions

Save not initiated

Invalid federation time

Federate not a federation execution member

RTI internal error

Related Services

Request Federation Save

Initiate Federate Save

Federate Save Achieved

2.14 Federate Save Achieved

Federate Initiated

Tells the RTI that the federate has completed its save attempt. The save-success indicator informs the RTI that the federate save either succeeded or failed.

Supplied Parameters

A save-success indicator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has saved its state

Post-conditions

The RTI has been informed of the status of the state save attempt

Exceptions

Invalid save-success indicator

Save not initiated

Federate not a federation execution member

RTI internal error

Related Services

Request Federation Save

Initiate Federate Save

Federate Save Begun

2.15 Request Restore

Federate initiated

Directs the RTI to begin the federation execution restoration process.

Supplied Parameters

The label supplied when the *Request Federation Save* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation has a save with the specified label

Post conditions

Federation restore is pending

Exceptions

Specified save label does not exist

Federate not a federation execution member

RTI internal error

Related Services

Initiate Restore

Restore Achieved

Request Federation Save

2.16 Initiate Restore †

RTI Initiated

Instructs the federate to return to a previously saved state indicated by the supplied federation save label.

Supplied Parameters

The label supplied when the *Request Restore* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation has a save with the specified label

Post-conditions

The federate has been informed to begin restoring state

Exceptions

No federate save associated with the label

Could not initiate restore

Federate internal error

Related Services

Request Restore

Restore Achieved

2.17 Restore Achieved

Federate Initiated

Tells the RTI that the federate has completed its restore attempt. If restore was successful, the federate is in the state it was in when the federation save associated with the label occurred.

Supplied Parameters

The label supplied when the *Initiate Restore* service was invoked

Restore-success indicator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Federate was directed to restore through invocation of the *Initiate Restore* service

If restore was successful, the federate is in a state identical to the state it was in when the federation save associated with the supplied label occurred. If restore was unsuccessful the federate is in an undefined state.

Post-conditions

The RTI has been informed of the status of the restore attempt

Exceptions

Unknown label

Invalid restore-success indicator

Restore not requested

Federate not a federation execution member

RTI internal error

Related Services

Request Restore

Initiate Restore

3 Declaration Management

The HLA declaration management approach requires federates to declare to the RTI their desire to both generate and receive object state information. These declarations must be consistent with the Federation Object Model and made using services described in this section. In addition to object state information, the interactions generated and received by a federate must also be declared.

3.1 Publish Object Class

Federate Initiated

The information conveyed by the federate via the *Publish Object Class* service is used in multiple ways. First, it indicates which attributes of an object class the federate is capable of providing to the federation. The federate may do this by creating objects of the class and then updating the attribute values. The federate may also (or alternatively) use ownership management services to acquire attributes of objects registered by another federate (of the same object class) and then update the values of those acquired attributes. Note that every object has the "privilegeToDeleteObject" attribute that is used to manage the privilege to delete an object. This attribute is like any other attribute and its value may be updated by the owning federate if desired. A federate must publish the privilegeToDeleteObject attribute for the object class whose instances it intends to delete. If a federate intends to acquire the privilege to delete objects registered by other federates, it must both publish and subscribe to the privilegeToDeleteObject attribute for those objects' classes.

Each use of this service with the include indicator replaces all information specified to the RTI in previous "inclusive" service invocations for the same object class.

Supplied Parameters

An object class designator

Indication to include or exclude specified object class

Set of attribute designators (for include actions only)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The specified object class and attributes are part of the FED

The federate must not own object attributes of the specified object class (if the specified object class is to be excluded)

Post-conditions

The federate may now register objects of the specified class (for include only)

The federate may not register objects of the specified class (for exclude only)

Exceptions

Object class not defined in the FED

Attribute(s) not defined in the FED

Federate owns attribute(s)

Federate is not a federation execution member

RTI internal error

Related Services

Subscribe Object Class Attribute

Register Object

3.2 Publish Interaction Class

Federate Initiated

Informs the RTI which classes of interactions the federate will be sending to the federation.

Supplied Parameters

An interaction class designator

Indication to include or exclude specified interaction class

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The interaction class is specified in the FED

Post-conditions

The federate can now send interactions of the specified class (for include only)

The federate may not send interactions of the specified class (for exclude only)

Exceptions

Interaction class not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Subscribe Interaction Class

Send Interaction

3.3 Subscribe Object Class Attribute

Federate Initiated

Specifies (via subscribe indicator parameter) an object class for which the RTI is to begin notifying the federate of discovery of instantiated objects or specifies (via unsubscribe indicator parameter) an object class for which the RTI is to stop notifying the federate of object instance discovery.

An object instance is a member of the specified object class if it has been registered as a member of the object class or any subclass (descendent object class) of the object class as defined in the FED. Subscribing to a given object class implies a subscription to that object class and all descendent object classes. Objects registered as subclasses will be discovered by the federate as the specified object class.

When "subscribing" to an object class, the federate may also provide a set of attribute designators. The values of only the specified attributes, for all objects discovered as a result of this "subscribe" service invocation, will be provided to the federate from the RTI (via the Reflect Attribute Values service). The set of attribute designators may include attributes from the "subscribed" object class and all super-classes, as defined in the FED.

Each use of this service with the subscribe indicator replaces all information specified to the RTI in any previous "subscribe" service invocations for the same object class.

Supplied Parameters

An object class designator

Indication to subscribe to an object class (including any listed attributes) or to unsubscribe to an object class

Set of attribute designators (for subscribe action only)

An optional region designator (see section 7)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The object class and attributes are defined in the FED

The subscription region exists

Post-conditions

The RTI has been informed of the federate’s desired subscription or unsubscription

Exceptions

Object class not defined in the FED

Attribute(s) not defined in the FED

Subscription region not known

Federate is not a federation execution member

RTI internal error

Related Services

Publish Object Class

Discover Object

Reflect Attribute Values

3.4 Subscribe Interaction Class

Federate Initiated

Specifies the class of interactions which should or should not be sent to the federate.

Subscribe Interaction Class receives an interaction class designator and an indication of the federate's desire to subscribe or unsubscribe to the interaction class. Subscribing to a given interaction class implies a subscription to that interaction class and all descendent interaction classes. Each use of this service replaces all information specified to the RTI in previous service invocations for the same interaction class.

Supplied Parameters

An interaction class designator

Indication to include or exclude interaction class in subscription list

Optional subscription region designator (see section 7)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The interaction class is defined in the FED

The subscription region exists

Post-conditions

The RTI will, or will not, as indicated, deliver interactions of the specified interaction class

Exceptions

Interaction class not defined in the FED

Subscription region not known

Federate is not a federation execution member

RTI internal error

Related Services

Publish Interaction Class

Receive Interaction

3.5 Control Updates †

RTI Initiated

The RTI can issue the *Control Updates* service to the federate at any time. It tells the federate that the specified attributes for the specified object class are or are not required somewhere in the federation. If the supplied parameter indicates that the attribute values should not be updated, the federate should cease updating the values of the referenced attributes. However, if the supplied parameter indicates that the attribute values should be updated, the federate must update the values of the referenced attributes.

Supplied Parameters

An object class

A set of attribute designators

Flag indicating that the values of the specified attributes should or should not be updated

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is updating the attribute values for the object class

In the case of a request to stop updating, no federate is subscribing to the attributes of the object class

In the case of a request to update, some other federate is subscribing to the attributes of the object class

Post-conditions

The federate must or may not (as specified) update the attribute values of this object class

Exceptions

Object class not published

Attribute not published

Federate internal error

Related Services

Publish Object Class

Subscribe Object Class Attribute

Control Interactions

3.6 Control Interactions †

RTI Initiated

The RTI can issue the *Control Interactions* service to the federate at any time. It tells the federate that the specified class of interactions is or is not required somewhere in the federation. If the supplied parameter indicates that the interaction should not be sent, the federate should cease sending the referenced interaction. However, if the supplied parameter indicates that the interaction should be sent, the federate must send the referenced interaction.

Supplied Parameters

An interaction class designator

Flag indicating that the specified interactions of this class should or should not be sent

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is sending the interaction class

In the case of a request to stop sending, no federate is subscribing to the interaction class

In the case of a request to send, some federate is subscribing to the interaction class

Post-conditions

If possible, the federate will or will not (as specified) send interactions of this class

Exceptions

Interaction class not published

Federate internal error

Related Services

Publish Interaction Class

Subscribe Interaction Class

Control Updates

4 Object Management

This group of RTI services deals with the creation, modification, and deletion of objects and the interactions they produce.

4.1 Request ID

Federate Initiated

Request federation execution-unique object ID numbers. Each ID is valid for only one object registration. If the object is deleted from the federation, the ID must not be reused. The RTI will not reuse IDs once they are deleted. A set of IDs may be reserved for special purposes and will not be issued by the *Request ID* service.

Supplied Parameters

The desired number of new IDs

Returned Parameters

Set of IDs

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate can register objects with the returned IDs

Exceptions

Too many IDs requested

ID supply exhausted

Federate is not a federation execution member

RTI internal error

Related Services

Register Object

4.2 Register Object

Federate Initiated

Links an object ID with an instance of an object class. All attributes specified as publishable by the registering federate (see definition of *Publish Object Class*) are initially set as owned by the registering federate.

Supplied Parameters

An object class designator

An object ID

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is publishing the object class

The ID is a valid object ID

The ID has not already been linked with another object

Post-conditions

The specified object ID is associated with the specified object class

Exceptions

Invalid object ID

Object ID is already linked with another object

Object class not defined in FED

Federate not publishing the specified object class

Federate is not a federation execution member

RTI internal error

Related Services

Request ID

Publish Object Class

Update Attribute Values

4.3 Update Attribute Values

Federate Initiated

Provides the current attribute values to the federation for attributes owned by the federate. The federate should supply changed attribute values as specified in the FED. This service, coupled with the *Reflect Attribute Values* service, form the primary data exchange mechanism supported by the RTI.

Supplied Parameters

An object ID

A set of attribute designator and value pairs

Federation time

A user-supplied tag

Returned Parameters

An event retraction designator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attribute for which values are provided

Post-conditions

The RTI will distribute the new attribute values to subscribing federates

Exceptions

Object not known

Attribute(s) not defined in the FED

The federate does not own the specified attributes

Invalid federation time

Federate is not a federation execution member

RTI internal error

Related Services

Reflect Attribute Values†

Retract

Register Object

Discover Object†

Associate Update Region

4.4 Discover Object †

RTI Initiated

The *Discover Object* service informs the federate that the RTI has discovered an object. An object is discovered when the following occur:

- An attribute update is received from another federate,
- The federate has neither registered nor discovered the object,
- The update satisfies the federate's subscription, and
- The update is in the associated region (if regions are being used, see section 7).

Supplied Parameters

An object ID

An object class designator

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The object class is published by some federate

The federate has subscribed to the object class

An object of that class has been registered with that ID

The federate has not discovered this object

An update attribute value has been issued by some other federate

Post-conditions

The object is known to the federate

Exceptions

The federate could not discover the object

Object class not known

Invalid federation time

Federate internal error

Related Services

Register Object

Update Attribute Values

Subscribe Object Class

4.5 Reflect Attribute Values †

RTI Initiated

Provides the federate with new values for a discovered attribute. This service, coupled with the *Update Attribute Values* service, forms the primary data exchange mechanism supported by the RTI.

Supplied Parameters

An object ID

A set of attribute designator and value pairs

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is reflecting the attribute values

Post-conditions

The new attribute values have been supplied to the federate

Exceptions

Object not known

Attribute not known

Invalid federation time

Federate internal error

Related Services

Update Attribute Values

Time Advance Request

Next Event Request

Time Advance Grant

4.6 Send Interaction

Federate Initiated

Informs the federation of an action taken by one object, potentially towards another object. The service returns a federation-unique event retraction designator.

Supplied Parameters

An interaction class designator

A set of parameter designator and value pairs

Federation time

A user-supplied tag

Returned Parameters

A event retraction designator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is publishing the interaction class

Post-conditions

The RTI has received the interaction

Exceptions

Federate not publishing the specified interaction class

Interaction class not defined in FED

Interaction parameter not defined in FED

Invalid federation time

Federate is not a federation execution member

RTI internal error

Related Services

Time Advance Request

Next Event Request

Time Advance Grant

Receive Interaction

Publish Interaction Class

Retract

Associate Update Region

4.7 Receive Interaction †

RTI Initiated

Provides information about an action taken by one federation object potentially towards another object.

Supplied Parameters

An interaction class designator

Interaction parameter designator and value pairs

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is subscribing to the interaction class

Post-conditions

The federate has received the interaction

Exceptions

Interaction class not known

Interaction parameter not known

Invalid federation time

Federate internal error

Related Services

Retract

Send Interaction

Subscribe Interaction Class

4.8 Delete Object

Federate Initiated

Informs the federation that an object with that ID, owned by the federate, is to be removed from the federation execution. Once the object is removed from the federation execution, its ID cannot be reused. The RTI will use the *Remove Object* service to inform the reflecting federates that the object has been deleted.

Supplied Parameters

An object ID

Federation time

User-supplied tag

Returned Parameters

A event retraction designator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

An object with that ID exists

The federate has the privilege to delete the object (it owns the privilegeToDeleteObject attribute)

Post-conditions

The federate no longer updates values for the object

The object does not exist in the federation execution

Exceptions

Federate does not own the delete privilege

Object not known

Invalid federation time

Federate is not a federation execution member

RTI internal error

Related Services

Remove Object

4.9 Remove Object †

RTI Initiated

Informs the federate that an object either is no longer in the associated region (if data distribution management services are used, see section 7), or the object has been deleted from the federation execution.

Supplied Parameters

An object ID

Object removal reason (deleted or out-of-scope)

Federation time

User-supplied tag

Optional event retraction designator (for deleted objects)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate knows about the object

Post-conditions

The federate has been advised to remove the object

Exceptions

Object not known

Invalid federation time

Federate internal error

Related Services

Delete Object

4.10 Change Attribute Transportation Type

Federate Initiated

The transportation type for each attribute of an object is defaulted from the object class description in the FED. A federate may choose to change the transportation type during execution. Invoking the *Change Attribute Transportation Type* service will change the transportation type for all future *Update Attribute Values* service calls for the specified attributes on the specified object.

Supplied Parameters

An object ID

A set of attribute designators

A transportation type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attribute

Post-conditions

The transportation type is changed for the specified attributes

Exceptions

Object not known

Attribute not defined

The federate does not own the specified attributes

Invalid transportation type

Federate is not a federation execution member

RTI internal error

Related Service

Update Attribute Values

Change Attribute Order Type

4.11 Change Attribute Order Type

Federate Initiated

The data ordering type for each attribute of an object is defaulted from the object class description in the FED. A federate may choose to change the data ordering type during execution. Invoking the *Change Attribute Order Type* service will change the data ordering type for all future *Update Attribute Values* service calls for the specified attributes on the specified object.

Supplied Parameters

An object ID

A set of attribute designators

A data ordering type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attribute

Post-conditions

The data ordering type is changed for the specified attributes

Exceptions

Object not known

Attribute not defined

The federate does not own the specified attributes

Invalid data ordering type

Federate is not a federation execution member

RTI internal error

Related Service

Update Attribute Values

Change Attribute Transportation Type

4.12 Change Interaction Transportation Type

Federate Initiated

The transportation type for each interaction is defaulted from the interaction class description in the FED. A federate may choose to change the transportation type during execution. Invoking the *Change Interaction Transportation Type* service will change the transportation type for all future *Send Interaction* service calls for the specified interaction class.

Supplied Parameters

An interaction class

A transportation type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must be publishing the interaction class

Post-conditions

The transportation type is changed for the specified interaction class

Exceptions

Interaction class not defined

Federate not publishing the interaction class

Invalid transportation type

Federate is not a federation execution member

RTI internal error

Related Service

Send Interaction

Change Interaction Order Type

4.13 Change Interaction Order Type

Federate Initiated

The data ordering type for each interaction is defaulted from the interaction class description in the FED. A federate may choose to change the data ordering type during execution. Invoking the *Change Interaction Order Type* service will change the data ordering type for all future *Send Interaction* service calls for the specified interaction class.

Supplied Parameters

An interaction class

A data ordering type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must be publishing the interaction class

Post-conditions

The data ordering type is changed for the specified interaction class

Exceptions

Interaction class not defined

Federate not publishing the interaction class

Invalid data ordering type

Federate is not a federation execution member

RTI internal error

Related Service

Send Interaction

Change Interaction Transportation Type

4.14 Request Attribute Value Update

Federate Initiated

The *Request Attribute Value Update* service is used to stimulate the update of values of specified attributes. When this service is used, the RTI will solicit the current values of the specified attributes from their owners using the *Provide Attribute Value Update* service. When an object class is specified, the RTI will solicit the specified attributes for all the objects of that class. When an object ID is specified, the RTI will solicit the specified attributes for the particular object. The *Request Attribute Value Update* service is intended to be used in exceptional situations, and is not intended to be used to build a "pull" system.

Supplied Parameters

Object ID or object class

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The request for the updated attribute values has been received by the RTI

Exceptions

Object not known

Object class not defined in FED

Attribute(s) not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Provide Attribute Value Update

Update Attribute Values

4.15 Provide Attribute Value Update†

RTI Initiated

Requests the current values for attributes owned by the federate for a given object. The federate should respond to the *Provide Attribute Value Update* service with an invocation of the *Update Attribute Values* service to provide the requested attribute values to the federation.

Supplied Parameters

An object ID

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the specified attributes

Post-conditions

The federate has been notified to provide updates of the specified attribute values

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Value Update

Update Attribute Values

4.16 Retract

Federate Initiated

Event retraction refers to the ability of a federate to retract (sometimes called cancel or unschedule) a previously scheduled event. This is a common discrete-event simulation primitive often used to model interrupts and other preemptive behaviors. Event retraction is also utilized by optimistic federates to implement mechanisms such as "anti-messages." The *Update Attribute Values* and *Send Interaction* RTI services return a designator for the event that is used to specify the event that is to be retracted. See the "HLA Time Management: Design Document" [HLA TM].

Supplied Parameters

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has previously issued *Update Attribute Values* and *Send Interaction* service calls and obtained the event retraction designators.

Post-conditions

The RTI is informed that the federate desires to retract the specified event

Exceptions

Invalid event retraction designator

Federate is not a federation execution member

RTI internal error

Related Services

Reflect Retraction

4.17 Reflect Retraction †

RTI Initiated

Event retraction refers to the ability of a federate to retract (sometimes called cancel or unschedule) a previously scheduled event. If the RTI receives a *Retract* service call for an event that has already been delivered to a federate, the *Reflect Retraction* service is invoked on the federates that received that event. See the "HLA Time Management: Design Document" [HLA TM].

Supplied Parameters

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The retracted event has been delivered to the federate

Post-conditions

The federate has been directed to retract the specified event

Exceptions

Event not known

Federate internal error

Related Services

Retract

5 Ownership Management

The ownership management group of services allows federates to transfer ownership of object attributes. An attribute is defined as a distinct and identifiable portion of the state of an object and is defined in the FED. Owning an attribute gives a federate the privilege to provide new values to the federation execution for that attribute. Additionally, ownership of the predefined attribute "privilegeToDeleteObject," gives the owning federate the right to remove an object from the federation execution. The RTI will automatically define attribute privilegeToDeleteObject for all object instances. A federate must publish the privilegeToDeleteObject attribute for the object class whose instances it intends to delete. If a federate intends to acquire the privilege to delete objects registered by other federates, it must both publish and subscribe the privilegeToDeleteObject attribute for those objects' classes. The value of the privilegeToDeleteObject attribute does not affect ownership management services.

In Figure 3, the federate on the left owns an object attribute that it is attempting to transfer to some other federate. In step 1, the federate invokes the *Request Attribute Ownership Divestiture* service with the object ID and designator of the attribute to be transferred and the Negotiated divestiture condition. The RTI then invokes the *Request Attribute Ownership Assumption* service on all federates that have indicated the ability to publish this attribute with the *Publish Object Class* service or, if federates were given in the *Request Attribute Ownership Divestiture* invocation, just the specified federates. Upon receipt of an *Request Attribute Ownership Assumption* invocation, a federate returns the attribute designator if it wishes to to own that attribute. The RTI selects a recipient for the attribute ownership from the set of federates that make ownership requests and completes the transfer, in step 3, with the invocation of the *Attribute Ownership Divestiture Notification* service on the divesting federate and *Attribute Ownership Acquisition Notification* on the acquiring federate. If no federates "volunteer" to own the attribute, it is up to the federate invoking the *Request Attribute Ownership Divestiture* service to "time out" and perform any appropriate actions. An "open" Negotiated *Request Attribute Ownership Divestiture* service request may be canceled by making a *Request Attribute Ownership Acquisition* call with the attributes the federate was trying to divest.

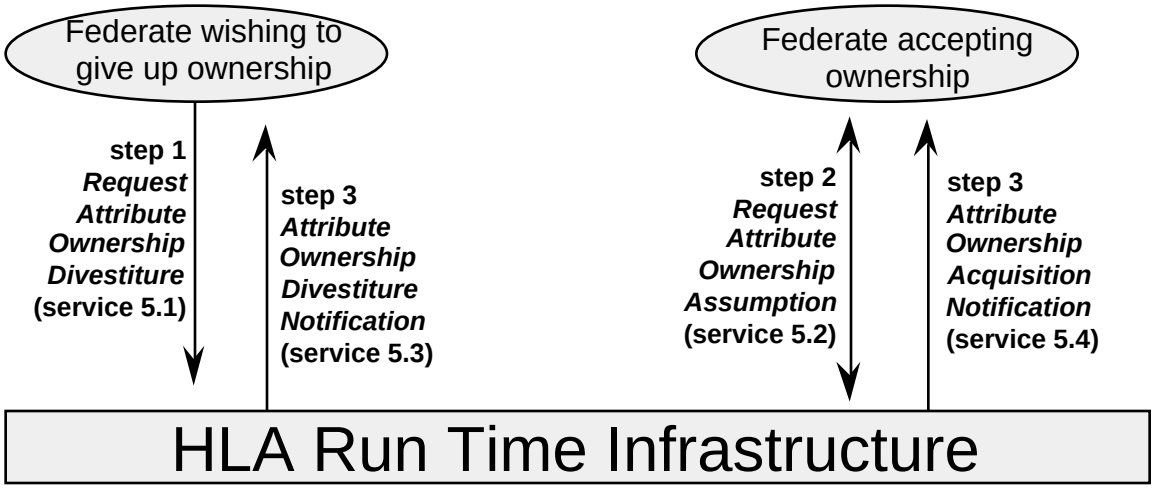

Figure 3. Federate Divesting Attribute Ownership (Negotiated)

If, on the other hand, a federate requests an Unconditional attribute ownership divestiture, the RTI immediately notifies the divesting federate of loss of attribute ownership via the *Attribute Ownership Divestiture Notification* service, the attribute is placed into an unowned state, and the RTI attempts to find an owner. The attribute remains in the unowned state until a) the RTI locates a federate willing to assume ownership or b) a federate requests ownership.

In Figure 4, the federate on the left is attempting to acquire ownership of an object attribute owned by the federate on the right. The federate informs the RTI of this desire, in step 1, using the *Request Attribute Ownership Acquisition* service. The RTI locates the current owner of the desired attribute, in step 2, and invokes the *Request Attribute Ownership Release* service providing the attribute owner with the designator of the desired attribute and the acquirer's user supplied tag (perhaps indicating why the transfer is desired). If the attribute owner decides to release ownership, it returns the designator of the attribute it is releasing. Lastly, the RTI informs the acquiring federate that it now owns the attribute, step 3.

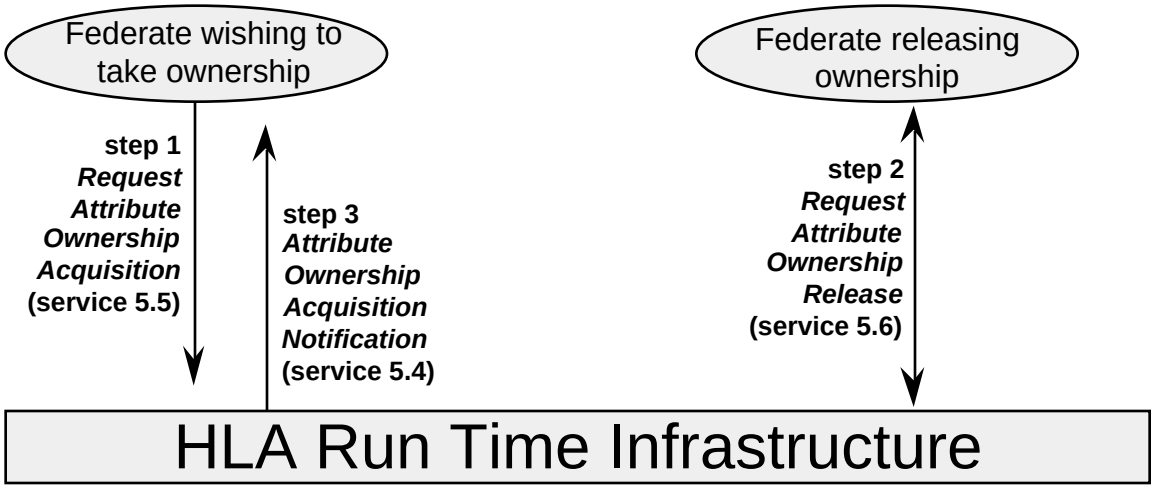

Figure 4. Federate Acquiring Attribute Ownership

5.1 Request Attribute Ownership Divestiture

Federate Initiated

Tells the RTI that the federate no longer wants to own the specified attributes of the specified object. The federate supplies an object ID and set of attribute designators.

Options:

1. The federate can specify which federate(s) can take ownership of the released attributes, otherwise any federate may own them.
2. The federate can indicate if the desired ownership divestiture is to be negotiated or unconditional. If the divestiture is negotiated, ownership will be transferred only if some federate(s) accepts. An unconditional transfer will relieve the divesting federate of the ownership, causing the attribute(s) to go into (possibly temporarily) the unowned state, without regard to the existence of an accepting federate.

The federate must continue its publication responsibility for the specified attributes until it receives permission to stop through a positive *Attribute Ownership Divestiture Notification* service. The use of this service is illustrated in figure 3.

Supplied Parameters

An object ID

A set of attribute designators

Ownership divestiture condition (negotiated or unconditional)

A user-supplied tag

Optional set of federates

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attributes specified in the service call

Post-conditions

No change in attribute ownership

The federate has informed the RTI of its intent to divest ownership of the specified attributes

Exceptions

Object not known

Attribute(s) not defined in the FED

Federate does not own the specified attributes

Invalid divestiture condition

Invalid candidate federate

Federate is not a federation execution member

RTI internal error

Related Services

Request Attribute Ownership Assumption

Attribute Ownership Divestiture Notification

Attribute Ownership Acquisition Notification

5.2 Request Attribute Ownership Assumption †

RTI Initiated

Informs the federate that the specified attributes for the specified object are available for transfer of ownership to the federate as a result of some other federation member invoking a *Request Attribute Ownership Divestiture* service. The RTI supplies an object ID and set of attribute designators. The federate returns the subset of the supplied attribute designators for which it is willing to assume ownership. The federate will be notified as to whether or not it received ownership of the attributes in a subsequent invocation of the *Attribute Ownership Acquisition Notification* service. The use of this service is illustrated in figure 3.

Supplied Parameters

An object ID

A set of attribute designators

A user-supplied tag

Returned Parameters

Set of attribute designators (possibly empty)

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is publishing and subscribing to the specified attributes

Post-conditions

No change in attribute ownership

The RTI has been informed of the set of attributes for which the federate is willing to assume ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Attribute Ownership Divestiture Notification

Attribute Ownership Acquisition Notification

5.3 Attribute Ownership Divestiture Notification †

RTI Initiated

Notifies the federate of the result of an invocation of the *Request Attribute Ownership Divestiture* service. The service provides the federate with the set of attributes that it no longer owns. Upon this notification, the federate should stop updating the specified attribute values. The federate may receive multiple notifications for a single invocation of the *Request Attribute Ownership Divestiture* service since different federates may wish to become the owner of different attributes. The use of this service is illustrated in figure 3.

Supplied Parameters

An object ID

The set of released attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has previously attempted to divest ownership of the specified attributes

Post-conditions

The federate does not own the specified attributes

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Request Attribute Ownership Assumption

Attribute Ownership Acquisition Notification

5.4 Attribute Ownership Acquisition Notification †

RTI Initiated

Notifies the federate of the result of an invocation of the *Request Attribute Ownership Acquisition* service or *Request Attribute Ownership Assumption* service. The service supplies the set of attributes that the federate has acquired. The federate may then begin updating those attribute values. The federate may receive multiple notifications for a single invocation of the *Request Attribute Ownership Acquisition* service since various federates may wish to become the owner of different attributes. The use of this service is illustrated in figures 3 and 4.

Supplied Parameters

An object ID

A set of acquired attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has previously attempted to acquire ownership of the specified attributes

Some other federate is willing to divest itself of ownership of the specified attributes

Post-conditions

The federate owns the specified attributes

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Request Attribute Ownership Assumption

Attribute Ownership Divestiture Notification

5.5 Request Attribute Ownership Acquisition

Federate Initiated

Requests the privilege to own the specified attributes of the specified object. The federate supplies an object ID and set of attribute designators. Before attempting to acquire ownership of an attribute, the federate should be reflecting the attribute value so it has the current attribute values. The federate may receive one or more *Attribute Ownership Acquisition Notification* invocations for each invocation of this service. The use of this service is illustrated in figure 4.

Supplied Parameters

An object ID

A set of attribute designators

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must be both publishing and subscribing to the specified attributes

The federate must not own the specified attributes

Post-conditions

The RTI has been informed of the federates desire to acquire ownership of the specified attributes

Exceptions

Object not known

Federate not publishing the object class

Federate not subscribing to the object class

Attribute(s) not defined in the FED

Federate not publishing the object attribute

Federate not subscribing to the object attribute

Federate already owns specified attributes

Federate is not a federation execution member

RTI internal error

Related Services

Request Attribute Ownership Release

Attribute Ownership Acquisition Notification

5.6 Request Attribute Ownership Release †

RTI Initiated

Requests that the federate release ownership of the specified attributes of the specified object. The *Request Attribute Ownership Release* service provides an object ID and set of attribute designators and is only invoked as the result of a *Request Attribute Ownership Acquisition* service invocation by some other federate. The federate returns the subset of the supplied attributes for which it is willing to release ownership. The use of this service is illustrated in figure 4.

Supplied Parameters

An object ID

A set of requested attribute designators for release

A user-supplied tag

Returned Parameters

Set of candidate attribute designators for release

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must own the specified attributes

Post-conditions

The RTI has been informed of the set of attributes of which the federate is willing to release ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Acquisition

Attribute Ownership Acquisition Notification

5.7 Query Attribute Ownership

Federate initiated

The *Query Attribute Ownership* service is used to determine the owner of the specified attribute of the specified object ID. The federate supplies an object ID and attribute designator. The RTI will provide the attribute owner information via the *Inform Attribute Ownership* service call.

Supplied Parameters

An object ID

An attribute designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The request for attribute ownership information has been received by the RTI.

Exceptions

Object not known

Attribute(s) not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Inform Attribute Ownership

5.8 Inform Attribute Ownership †

RTI initiated

The *Inform Attribute Ownership* service is used to provide ownership information for the specified attribute of the specified object. This service is invoked by the RTI in response to a *Query Attribute Ownership* service call by a federate. This service provides the federate designator of the attribute owner (if the attribute is owned) or an indication that the attribute is available for acquisition.

Supplied Parameters

An object ID

An attribute designator

A federate designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate has been informed of the attribute ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Query Attribute Ownership

5.9 Is Attribute Owned By Federate

Federate initiated

The *Is Attribute Owned By Federate* service is used to determine if the specified attribute of the specified object ID is owned by the invoking federate. The federate supplies an object ID and attribute designator. The service returns a Boolean value indicating ownership status.

Supplied Parameters

An object ID

An attribute designator

Returned Parameters

Attribute ownership indicator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate has the requested ownership information.

Exceptions

Object not known

Attribute not known

Federate is not a federation execution member

RTI internal error

Related Services

None

6 Time Management

Time management is concerned with the mechanisms for controlling the advancement of federates along the federation time axis during the execution. In general, time advances must be coordinated with object management services so that information is delivered to federates (e.g., state updates and interactions) in a timely and ordered fashion, thereby enabling federates to satisfy requirements for reproducing causal behavior in the system being modeled. For example, certain federates may require that notifications of events be delivered to the federate in time stamp order, and are not delivered in the federate's past

(time stamp less than the federate's current time). The time management services are intended to support federations that include federates with different ordering and delivery requirements. For further information, see the "HLA Time Management: Design Document" [HLA TM].

6.1 Request Federation Time

Federate Initiated

Requests a current estimate of the federation time. Federation time is defined to be the minimum of Lower Bound Time Stamp (LBTS) and the current value of the federate's Logical Time (LT).

Supplied Parameters

None

Returned Parameters

The current minimum of LBTS and LT

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate receives the current value of federation time

Exceptions

Federate is not a federation execution member

RTI internal error

Related Services

Request LBTS

Request Federate Time

Request Minimum Next Event Time

6.2 Request LBTS

Federate Initiated

Requests the current value of LBTS.

Supplied Parameters

None

Returned Parameters

The current value of LBTS

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate receives the current value of LBTS

Exceptions

Federate is not a federation execution member

RTI internal error

Related Services

Request Federation Time

Request Federate Time

Request Minimum Next Event Time

6.3 Request Federate Time

Federate Initiated

Requests the current value of the federate's Logical Time (LT).

Supplied Parameters

None

Returned Parameters

The current value of LT.

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate receives the current value of LT

Exceptions

Federate is not a federation execution member

RTI internal error

Related Services

Request Federation Time

Request LBTS

Request Minimum Next Event Time

6.4 Request Minimum Next Event Time

Federate Initiated

Requests the minimum of LBTS and the time stamp of the next Time Stamp Ordered (TSO) message that is held by the RTI for delivery to the requesting federate.

Supplied Parameters

None

Returned Parameters

The minimum of the following:

LBTS and

the time of the message at the head of the TSO queue (if it contains a message).

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate receives the minimum next event time.

Exceptions

Federate is not a federation execution member

RTI internal error

Related Services

Request Federation Time

Request LBTS

Request Federate Time

6.5 Set Lookahead

Federate Initiated

Sets the desired value of the federate's lookahead.

Supplied Parameters

A desired value of lookahead

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The RTI is informed of the federate's new value of lookahead

Exceptions

Invalid Lookahead Time

Federate is not a federation execution member

RTI internal error

Related Services

Request Lookahead

6.6 Request Lookahead

Federate Initiated

Queries the RTI for the current value of the federate's lookahead. The current value of lookahead may differ temporarily from the desired lookahead given in the *Set Lookahead* service if the value of lookahead has been reduced.

Supplied Parameters

None

Returned Parameters

The federate's current value of lookahead

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate receives the current value of it's lookahead

Exceptions

Federate is not a federation execution member

RTI internal error

Related Services

Set Lookahead

6.7 Time Advance Request

Federate Initiated

The *Time Advance Request* service requests an advance of the federate's logical time. Invocation of this service implies that the following messages are eligible for delivery to the federate: (i) all incoming receive ordered messages, and (ii) all messages using other ordering services with time stamp less than or equal to the specified time. After invoking *Time Advance Request*, the messages are passed to the federate by the RTI invoking the *Discover Object, Remove Object, Receive Interaction* and *Reflect Attribute Values* services. By invoking *Time Advance Request* with the specified time, the federate is guaranteeing that it will not generate a Time Stamp Ordered (TSO) message at any time in the future with time stamp less than the specified time, plus that federate's current lookahead. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the specified time.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The RTI is informed of the federate's desire to advance time

Exceptions

Invalid federation time

Time Advance Request or *Next Event Request*, or *Flush Queue Request* already pending

Federate is not a federation execution member

RTI internal error

Related Services

Next Event Request

Flush Queue Request

Time Advance Grant

6.8 Next Event Request

Federate Initiated

The *Next Event Request* service requests the next TSO message from the RTI, provided that message has a time stamp no greater than the logical time specified in the request. The invocation of this service implies that the following messages are eligible for delivery to the federate:

- All receive ordered messages,
- The smallest time stamped TSO message that will ever be delivered in the future with time stamp less than or equal to the specified time, and all other TSO messages containing this same time stamp value, and
- All messages using other ordering services with time stamp less than or equal to the specified time.

These messages are delivered to the federate by the RTI calling the *Discover Object*, *Remove Object*, *Receive Interaction* or *Reflect Attribute Values* services provided by the federate. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the latest time stamp of all the TSO messages that are delivered, if any, or to the specified time if no TSO messages were delivered.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The RTI is informed of the federate's desire to advance time

Exceptions

Invalid federation time

Time Advance Request or *Next Event Request*, or *Flush Queue Request* already pending

Federate is not a federation execution member

RTI internal error

Related Services

Time Advance Request

Flush Queue Request

Time Advance Grant

6.9 Flush Queue Request

Federate Initiated

The *Flush Queue Request* releases all messages stored in the RTI's internal queues and delivers them to the federate invoking this service. Time Stamp Ordered (TSO) messages are delivered, despite the fact that the RTI may not be able to guarantee future messages containing a smaller time stamp could arrive. If the federate does not receive any additional TSO messages with time stamp less than the specified time, then the federate's LT may be advanced up to the specified time. Received messages are delivered to the federate by the RTI calling the *Discover Object*, *Remove Object*, *Receive Interaction* or *Reflect Attribute Values* services provided by the federate. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the minimum of the specified time, LBTS, and the time stamp of the next TSO message to be received.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The RTI is informed of the federate's desire to advance time

Exceptions

Invalid federation time

Time Advance Request or *Next Event Request*, or *Flush Queue Request* already pending

Federate is not a federation execution member

RTI internal error

Related Services

Time Advance Request

Next Event Request

Time Advance Grant

6.10 Time Advance Grant †

RTI Initiated

Invocation of this service indicates a prior request to advance Logical Time (LT) has been honored. The parameter of this service indicates that LT for the federate has been advanced to this value and no additional Time Stamped Ordered (TSO) messages will be delivered in the future with a time stamp less than this value.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Any of the *Time Advance Request*, *Next Event Request* or *Flush Queue Request* services were previously invoked

Post-conditions

No additional TSO messages will be received with a time stamp less than the provided time

Exceptions

Invalid federation time

Time Advance Request, *Next Event Request* or *Flush Queue Request* was not pending

Federate internal error

Related Services

Time Advance Request

Next Event Request

Flush Queue Request

7 Data Distribution Management

To support efficient data distribution across a federation, the RTI provides a set of services which facilitate the explicit management of data distribution. The fundamental concept to support data distribution is called *routing spaces*. A routing *space* is a multidimensional coordinate system in which federates express an interest for either receiving data or sending data.

To use routing spaces, each federation defines the allowable routing spaces for the federation execution, including the dimensions (variables) of the routing space. Routing spaces are then initialized in the FED with a name and the number of dimensions.

Using declaration management services (Section 3), federates specify by class and attribute designator and by interaction class, the types of data they will send or receive. Routing spaces are used by federates to specify the distribution conditions for the specific data they are sending or expect to receive. Each federate decides which of the federation routing spaces are useful to them and defines the portions (subsets) of those routing spaces that specify (from the federate's perspective) *regions*, or logical areas of interest particular to the federate, by putting bounds (*extents*) on the dimensions of the selected routing space. The federate then uses these regions to both specify conditions (*Create Subscription Region*) under which they expect to receive the object state data and interactions they specified using declaration management services (*Subscribe Object Class Attribute* and *Subscribe Interaction Class*) and to specify conditions under which they are providing data *(Create Update Region and Associate Update Region).*

Specifying a subscription region, the federate tells the RTI to only deliver data which fall within the bounds (extents) of the region specified by that federate. Specifying an update region and associating that update region with a particular object instance, means that the federate will ensure that the characteristics of the object instance or interaction which map to the dimensions of the routing space fall within the bounds (extents) of the associated region at the time that the attribute update or send interaction call is issued. This implies that the federate is monitoring these added characteristics for each of the attributes owned by the federate. As the state of the objects change, the federate may need to either adjust the bounds on the associated regions (*Modify Region*) or change the association to another region (*Associate Update Region*).

The routing space, regions, and association data is used by the RTI to distribute data. When an update region and subscription regions of different federates overlap, the RTI ensures that the attribute updates and interactions associated with that update region are routed to federates with subscription regions which overlap the sender's update region.

It is important to note that the dimensions of routing spaces need not necessarily map to attributes in the FED. The data distribution management services provide a federation the capability to specify data distribution on characteristics of objects other than those exchanged as part of federation execution. By specifying routing spaces and regions using attributes specified in the FED, the data distribution management services provide a mechanism to control data distribution based on values of attributes.

Each federate can create multiple update and subscription regions. Update regions may be associated with individual objects that have been registered with the RTI. As an example, a federate might have a subscription region for each sensor system being simulated.

Figure 5 shows one update region (U1) and two subscription regions (S1, S2) within a two dimensional routing space. In this example, U1 and S1 overlap and therefore attributes and interactions associated with U1 will be routed by the RTI to the federate that created S1. In contrast U1 and S2 do not overlap and attributes and interactions will not be routed from U1 to the federate that created S2.

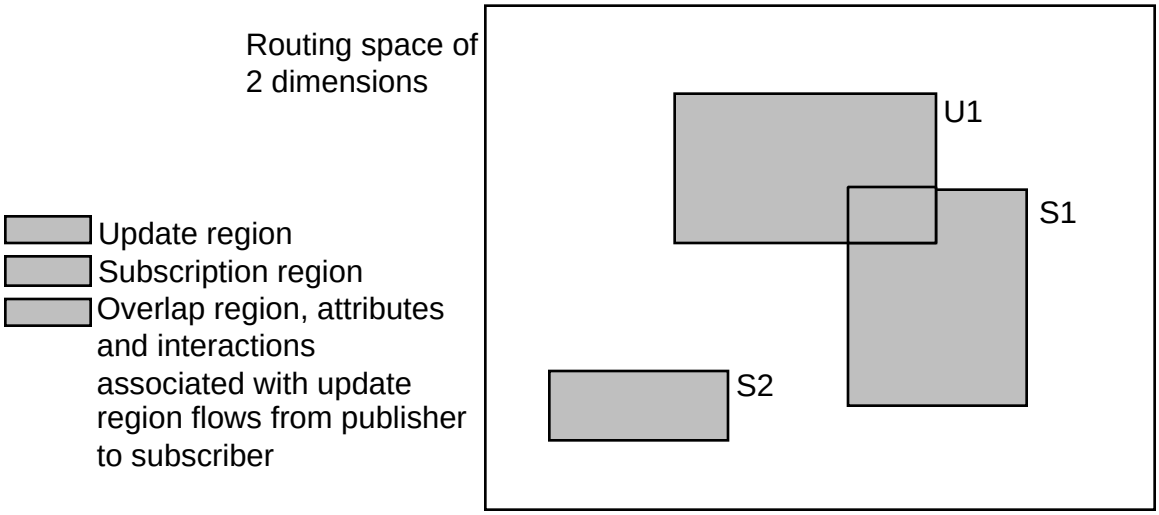

Figure 5. Routing Space Example

Disclaimer: There is ongoing exploration of the Data Distribution Management services, especially with respect to the interaction of Time Management and Data Distribution Management. As such, this specification contains no information concerning aspects of the interaction, such as at what point in federation time a certain Data Distribution Management service takes effect.

7.1 Create Update Region

Federate Initiated

Creates an update region that is a subset of the specified routing space. The extent set bounds the region within the routing space.

Supplied Parameters

A routing space designator

A set of extents

Returned Parameters

An update region designator

Pre-conditions

The routing space is defined in the FED

Post-conditions

An update region exists that is a subset of the specified routing space

Exceptions

Routing space not defined in the FED

Invalid extents

Federate is not an execution member

RTI internal error

Related Services

Create Subscription Region

Associate Update Region

Change Thresholds †

Modify Region

Delete Region

7.2 Create Subscription Region

Federate Initiated

Creates a subscription region that is a subset of the specified routing space. The extent set bounds the region within the routing space. In creating a subscription region, the federate tells the RTI to only deliver data which fall within the bounds (extents) of the region.

Supplied Parameters

A routing space designator

A set of extents

Returned Parameters

A subscription region designator

Pre-conditions

The routing space is defined in the FED

Post-conditions

A subscription region exists that is a subset of the specified routing space

Exceptions

Routing space not defined in the FED

Invalid extents

Federate is not a federation execution member

RTI internal error

Related Services

Create Update Region

Associate Update Region

Change Thresholds †

Modify Region

Delete Region

Subscribe Object Class Attribute

Subscribe Interaction

7.3 Associate Update Region

Federate Initiated

This service associates an update region with

- attributes of a specific object, or
- an interaction class.

Associating an update region with a particular object instance or interaction class, means that the federate will ensure that the characteristics of the object instance or interaction which map to the dimensions of the routing space fall within the bounds (extents) of the associated region at the time that the attribute update or send interaction call is issued. This implies that the federate is monitoring these added characteristics for each of the attributes owned by the federate. As the state of the objects change, as required the federate must either adjust the bounds on the associated regions (*Modify Region*) or change the association to another region (*Associate Update Region*).

The association is used by the *Update Attribute Values* and *Send Interaction* services to route data to subscribers whose subscription regions overlap the specified update region.

Supplied Parameters

A update region designator

For interaction classes:

An interaction class

or for objects:

An object ID

A set of attributes

An indication to form or break the association

Returned Parameters

None

Pre-conditions

The update region exists

The object ID is registered

The federate is publishing the specified attributes, or

The federate is publishing the specified interaction class

Post-conditions

The update region is associated with the object ID or interaction class for future *Update Attribute Values* or *Send Interactions* service invocations

Exceptions

Region not known

Object not known

Attributes not defined in the FED

Interaction class not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Create Update Region

Change Thresholds †

Modify Region

Delete Region

Update Attribute Values

Send Interaction

7.4 Change Thresholds †

RTI Initiated

Change Thresholds provides the federate with the new values of the thresholds for each of the dimensions of a routing space. The new threshold data should be used by the federate to decided when to invoke the *Modify Region* service.

Supplied Parameters

An update or subscription region designator

A set of thresholds

Returned Parameters

None

Pre-conditions

The region exists

Post-conditions

The federate in informed of the new threshold values for the region

Exceptions

Region not known

Federate internal error

Related Services

Create Subscription Region

Create Update Region

Modify Region

7.5 Modify Region

Federate Initiated

Modify Region changes the bounds of the update or subscription region to reflect the specified set of extents.

Supplied Parameters

An update or subscription region designator

A set of extents

Returned Parameters

None

Pre-conditions

The region exists

Post-conditions

The region has a new extent

Exceptions

Region not known

Invalid extents

Federate is not a federation execution member

RTI internal error

Related Services

Create Subscription Region

Create Update Region

Change Thresholds †

7.6 Delete Region

Federate Initiated

Deletes the specified update or subscription region.

Supplied Parameters

An update or subscription region designator

Returned Parameters

None

Pre-conditions

The region exists

Post-conditions

The region no longer exists

The region is no longer associated with any interaction class or object attributes

Exceptions

Region not known

Federate is not a federation execution member

RTI internal error

Related Services

Create Subscription Region

Create Update Region

8 HLA IDL Application Programmer's Interface

```
// File: baseTypes.idl

#ifndef __baseTypes
#define __baseTypes

interface baseTypes {

typedef unsigned short RTI_UShort;
typedef short RTI_Short;
typedef unsigned long RTI_ULong;
typedef long RTI_Long;
typedef double RTI_Double;
typedef float RTI_Float;
typedef boolean RTI_Boolean;
typedef any RTI_Datum;
typedef string RTI_String;
typedef unsigned long RTI_Id;

};

#endif
```

```
//File: rtiTypes.idl

#ifndef __rtiTypes
#define __rtiTypes

#include "baseTypes.idl"

interface rtiTypes: baseTypes {

enum OrderType {
    ORDER_TYPE_NONE,
    RECEIVE,
    TIMESTAMP
};

enum OwnershipDivestitureCondition {
    NEGOTIATED,
    UNCONDITIONAL
};

enum ObjectRemovalReason {
    OBJECT_OUT_OF_REGION,
    OBJECT_DELETED,
    NO_LONGER_SUBSCRIBED
};

enum TransportType {
    TRANSPORT_TYPE_NONE,
    RELIABLE,
    BEST_EFFORT
};

enum ResignAction {
    RELEASE_ATTRIBUTES,
    DELETE_OBJECTS,
    DELETE_OBJECTS_AND_RELEASE_ATTRIBUTES,
    NO_ACTION
};

typedef RTI_ULong UniqueId;

typedef RTI_Double FederationTimeDelta;
typedef RTI_Double FederationTime;
typedef RTI_Double WallClockTime;
typedef RTI_Double TickTime;

typedef RTI_UShort FederateHandle;
typedef RTI_String FederateName;
typedef sequence <FederateHandle> FederateHandleSet;

typedef RTI_UShort ObjectClassHandle;
typedef RTI_String ObjectClassName;

typedef RTI_UShort InteractionClassHandle;
typedef RTI_String InteractionClassName;

typedef RTI_UShort AttributeHandle;
typedef RTI_String AttributeName;
typedef RTI_Datum AttributeValue;
typedef sequence <AttributeHandle> AttributeHandleSet;
typedef struct AttributeHandleValuePair_s {
```

```
    AttributeHandle name;
    AttributeValue value;
} AttributeHandleValuePair;
typedef sequence <AttributeHandleValuePair> AttributeHandleValuePairSet;

typedef RTI_UShort ParameterHandle;
typedef RTI_String ParameterName;
typedef RTI_Datum ParameterValue;
typedef struct ParameterHandleValuePair_s {
    ParameterHandle name;
    ParameterValue value;
} ParameterHandleValuePair;
typedef sequence <ParameterHandleValuePair> ParameterHandleValuePairSet;

typedef RTI_ULong ObjectId;
typedef sequence <ObjectId> ObjectIdSet;
typedef RTI_UShort ObjectIdCount;

typedef struct EventRetractionHandle_s {
    FederationTime  theEventTime;
    UniqueId        theSerialNumber;
    FederateHandle  sendingFederate;
} EventRetractionHandle;

typedef RTI_Long SpaceHandle;
typedef RTI_String SpaceName;

typedef RTI_String FederationExecutionName;

typedef RTI_String FileName;

typedef RTI_String PauseLabel;

typedef RTI_String SaveLabel;

typedef RTI_Datum UserSuppliedTag;

typedef struct Range_s {
    RTI_ULong first;
    RTI_ULong last;
} Range;
typedef sequence <Range> Extents;
typedef sequence <Extents> ExtentSet;

typedef RTI_ULong Region;

typedef struct Threshold_s {
    RTI_ULong value;
} Threshold;
typedef sequence <Threshold> ThresholdSet;

exception AttributeAlreadyOwned {};
exception AttributeNotDefined {};
exception AttributeNotKnown {};
exception AttributeNotOwned {};
exception AttributeNotPublished {};
exception AttributeNotSubscribed {};
exception CouldNotDiscover {};
exception CouleNotLocateFED{};
exception CouldNotInitiateRestore{};
exception DeletePrivilegeNotHeld {};
```

```
exception InvalidFED{};
exception EventNotKnown {};
exception FederateAlreadyPaused {};
exception FederateAlreadyExecutionMember {};
exception FederateDoesNotExist {};
exception FederateInternalError {};
exception FederateNotExecutionMember {};
exception FederateNotPaused {};
exception FederateNotPublishingInteractionClass {};
exception FederateOwnsAttributes {};
exception FederatesCurrentlyJoined {};
exception FederationAlreadyPaused {};
exception FederationExecutionAlreadyExists {};
exception FederationExecutionDoesNotExist {};
exception FederationNotPaused {};
exception FederationTimeAlreadyPassed {};
exception IdSupplyExhausted {};
exception InteractionClassNotDefined {};
exception InteractionClassNotKnown {};
exception InteractionClassNotPublished {};
exception InteractionParameterNotDefined {};
exception InteractionParameterNotKnown {};
exception InvalidDivestitureCondition {};
exception InvalidExtents {};
exception InvalidFederationTime {};
exception InvalidLookAheadTime {};
exception InvalidObjectId {};
exception InvalidOrderType {};
exception InvalidRetractionHandle {};
exception InvalidResignAction {};
exception InvalidTransportType {};
exception NameNotFound {};
exception NoPauseRequested {};
exception NoResumeRequested {};
exception ObjectClassNotDefined {};
exception ObjectClassNotKnown {};
exception ObjectClassNotPublished {};
exception ObjectClassNotSubscribed {};
exception ObjectNotKnown {};
exception ObjectIdAlreadyLinked {};
exception RegionNotKnown {};
exception RestoreNotRequested {};
exception RTIinternalError {};
exception SpaceNotDefined {};
exception SaveInProgress {};
exception SaveNotInitiated {};
exception SpecifiedSaveLabelDoesNotExist {};
exception TimeAdvanceAlreadyInProgress {};
exception TimeAdvanceNotInProgress {};
exception TooManyIdsRequested {};
exception UnableToPerformSave {};
exception UnimplementedService {};
exception UnknownLabel {};

};

#endif
```

```
//File: rtiAmbServices.idl

#ifndef __rtiAmbServices
#define __rtiAmbServices

#include "rtiTypes.idl"

////////////////////////////////////
// Federation Management Services //
////////////////////////////////////

interface rtiAmbassador: rtiTypes {

// 2.1
void createFederationExecution (
  in FederationExecutionName    theName,
  in FileName                   FED)
raises (
  FederationExecutionAlreadyExists,
  CouleNotLocateFED,
  InvalidFED,
  RTIinternalError,
  UnimplementedService);

// 2.2
void destroyFederationExecution (
  in FederationExecutionName    executionName)
raises (
  FederatesCurrentlyJoined,
  FederationExecutionDoesNotExist,
  RTIinternalError,
  UnimplementedService);

// 2.3
void joinFederationExecution (
  in FederateName               theFederateName,
  in FederationExecutionName    executionName,
  out FederateHandle            theFederate)
raises (
  FederateAlreadyExecutionMember,
  FederationExecutionDoesNotExist,
  RTIinternalError,
  UnimplementedService);

// 2.4
void resignFederationExecution (
  in ResignAction               theAction)
raises (
  FederateOwnsAttributes,
  FederateNotExecutionMember,
  FederationExecutionDoesNotExist,
  InvalidResignAction,
  RTIinternalError,
  UnimplementedService);

// 2.5
void requestPause (
  in PauseLabel label)
raises (
  FederationAlreadyPaused,
  FederateNotExecutionMember,
```

```
  RTIinternalError,
  UnimplementedService);

// 2.7
void pauseAchieved (
  in PauseLabel label)
raises (
  UnknownLabel,
  NoPauseRequested,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.8
void requestResume ()
raises (
  FederationNotPaused,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.10
void resumeAchieved ()
raises (
  NoResumeRequested,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.11
void requestFederationSave (
  in SaveLabel          label,
  in FederationTime     theTime)
raises (
  FederationTimeAlreadyPassed,
  InvalidFederationTime,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void requestFederationSave_OVER0 (
  in SaveLabel  label)
raises (
  SaveInProgress,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.13
void federateSaveBegun ()
raises (
  SaveNotInitiated,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void federateSaveBegun_OVER0 (
  in FederationTime     theTime)
raises (
  SaveNotInitiated,
  InvalidFederationTime,
```

```
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.14
void federateSaveAchieved ()
raises (
  SaveNotInitiated,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void federateSaveNotAchieved ()
raises (
  SaveNotInitiated,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.15
void requestRestore (
  in SaveLabel          label)
raises (
  SpecifiedSaveLabelDoesNotExist,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 2.17
void restoreAchieved (
  in SaveLabel  label)
raises (
  UnknownLabel,
  RestoreNotRequested,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void restoreNotAchieved (
  in SaveLabel  label)
raises (
  UnknownLabel,
  RestoreNotRequested,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

////////////////////////////////////
// Declaration Management Services //
////////////////////////////////////

// 3.1
void publishObjectClass (
  in ObjectClassHandle  theClass,
  in AttributeHandleSet attributeList)
raises (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);
```

```
void unpublishObjectClass (
  in ObjectClassHandle  theClass)
raises (
  ObjectClassNotDefined,
  FederateOwnsAttributes,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 3.2
void publishInteractionClass (
  in InteractionClassHandle     theInteraction)
raises (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void unpublishInteractionClass (
  in InteractionClassHandle     theInteraction)
raises (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 3.3
void subscribeObjectClassAttribute (
  in ObjectClassHandle  theClass,
  in AttributeHandle    theHandle)
raises (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void subscribeObjectClassAttribute_OVER0 (
  in ObjectClassHandle  theClass,
  in AttributeHandle    theAttribute,
  in Region             theRegion)
raises (
  ObjectClassNotDefined,
  AttributeNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void unsubscribeObjectClassAttribute (
  in ObjectClassHandle  theClass)
raises (
  ObjectClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void unsubscribeObjectClassAttribute_OVER0 (
  in ObjectClassHandle  theClass,
  in Region             theSubscribeRegion)
```

```
raises (
  ObjectClassNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 3.4
void subscribeInteractionClass (
  in InteractionClassHandle     theClass)
raises (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void subscribeInteractionClass_OVER0 (
  in InteractionClassHandle     theClass,
  in Region                     theRegion)
raises (
  InteractionClassNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void unsubscribeInteractionClass (
  in InteractionClassHandle     theClass)
raises (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void unsubscribeInteractionClass_OVER0 (
  in InteractionClassHandle     theClass,
  in Region                     theRegion)
raises (
  InteractionClassNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

//////////////////////////////
// Object Management Services //
//////////////////////////////

// 4.1
void requestId (
  in ObjectIdCount      theCount,
  out ObjectIdSet       theList)
raises (
  TooManyIdsRequested,
  IdSupplyExhausted,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.2
void registerObject (
```

```
  in ObjectClassHandle  theClass,
  in ObjectId           theObject)
raises (
  InvalidObjectId,
  ObjectIdAlreadyLinked,
  ObjectClassNotDefined,
  ObjectClassNotPublished,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.3
void updateAttributeValues (
  in ObjectId                          theObject,
  in AttributeHandleValuePairSet       attributeList,
  in FederationTime                    theTime,
  in UserSuppliedTag                   theTag,
  out EventRetractionHandle            theHandle)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidFederationTime,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.6
void sendInteraction (
  in InteractionClassHandle            theInteraction,
  in ParameterHandleValuePairSet       parameters,
  in FederationTime                    theTime,
  in UserSuppliedTag                   theTag,
  out EventRetractionHandle            theHandle)
raises (
  InteractionClassNotPublished,
  InteractionClassNotDefined,
  InteractionParameterNotDefined,
  InvalidFederationTime,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.8
void deleteObject (
  in ObjectId                  theObject,
  in FederationTime            theTime,
  in UserSuppliedTag           theTag,
  out EventRetractionHandle    theHandle)
raises (
  DeletePrivilegeNotHeld,
  ObjectNotKnown,
  InvalidFederationTime,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.10
void changeAttributeTransportType (
  in ObjectId           theObject,
  in AttributeHandleSet attributeList,
```

```
  in TransportType      theType)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidTransportType,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.11
void changeAttributeOrderType (
  in ObjectId           theObject,
  in AttributeHandleSet attributeList,
  in OrderType          theType)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidOrderType,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.12
void changeInteractionTransportType (
  in InteractionClassHandle     theClass,
  in TransportType              theType)
raises (
  InteractionClassNotDefined,
  FederateNotPublishingInteractionClass,
  InvalidTransportType,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.13
void changeInteractionOrderType (
  in InteractionClassHandle     theClass,
  in OrderType                  theType)
raises (
  InteractionClassNotDefined,
  FederateNotPublishingInteractionClass,
  InvalidOrderType,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.14
void requestObjectAttributeValuesUpdate (
  in ObjectId           theObject,
  in AttributeHandleSet attributeList)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void requestClassAttributeValuesUpdate (
  in ObjectClassHandle  theClass,
```

```
  in AttributeHandleSet attributeList)
raises (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 4.16
void retract (
  in eventRetractionHandle      theHandle)
raises (
  InvalidRetractionHandle,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

///////////////////////////////////
// Ownership Management Services //
///////////////////////////////////

// 5.1
void requestAttributeOwnershipDivestiture (
  in ObjectId                           theObject,
  in AttributeHandleSet                 attrs,
  in OwnershipDivestitureCondition      theCondition,
  in UserSuppliedTag                    theTag)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidDivestitureCondition,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void requestAttributeOwnershipDivestiture_OVER0 (
  in ObjectId                           theObject,
  in AttributeHandleSet                 attrs,
  in OwnershipDivestitureCondition      theCondition,
  in UserSuppliedTag                    theTag,
  in FederateHandleSet                  candidates)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidDivestitureCondition,
  FederateDoesNotExist,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 5.5
void requestAttributeOwnershipAcquisition (
  in ObjectId           theObject,
  in AttributeHandleSet attrs,
  in UserSuppliedTag    theTag)
raises (
  ObjectNotKnown,
  ObjectClassNotPublished,
  ObjectClassNotSubscribed,
```

```
  AttributeNotDefined,
  AttributeNotPublished,
  AttributeNotSubscribed,
  FederateOwnsAttributes,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 5.7
void queryAttributeOwnership (
  in ObjectId           theObject,
  in AttributeHandle    theAttr)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 5.9
void isAttributeOwnedByFederate (
  in ObjectId           theObject,
  in AttributeHandle    theAttr,
  out RTI_Boolean       ownershipIndicator)
raises (
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

//////////////////////////////
// Time Management Services //
//////////////////////////////

// 6.1
void requestFederationTime (
  out FederationTime    theTime)
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.2
void requestLBTS (
  out FederationTime    theTime)
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.3
void requestFederateTime (
  out FederationTime    theTime)
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.4
void requestMinNextEventTime (
```

```
  out FederationTime    theTime)
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.5
void setLookahead (
  in FederationTimeDelta        lookahead)
raises (
  InvalidLookaheadTime,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.6
void requestLookahead (
 out FederationTimeDelta        theLookahead)
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.7
void timeAdvanceRequest (
  in FederationTime     theTime)
raises (
  InvalidFederationTime,
  TimeAdvanceAlreadyInProgress,
  FederationTimeAlreadyPassed,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.8
void nextEventRequest (
  in FederationTime     theTime)
raises (
  InvalidFederationTime,
  TimeAdvanceAlreadyInProgress,
  FederationTimeAlreadyPassed,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 6.9

void flushQueueRequest (
  in FederationTime     EtheTime)
raises (
  InvalidFederationTime,
  TimeAdvanceAlreadyInProgress,
  FederationTimeAlreadyPassed,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

//////////////////////////////////////////
// Data Distribution Management Services //
//////////////////////////////////////////
```

```
// 7.1
void CreateUpdateRegion (
  in SpaceHandle        theSpace,
  in ExtentSet          theExtents,
  out Region            theRegion)
raises (
  SpaceNotDefined,
  InvalidExtents,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 7.2
void CreateSubscriptionRegion (
  in SpaceHandle        theSpace,
  in ExtentSet          theExtents,
  out Region            theRegion)
raises (
  SpaceNotDefined,
  InvalidExtents,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 7.3
void associateUpdateRegion (
  in Region             theRegion,
  in ObjectId           theObject,
  in AttributeHandleSet theAttributes)
raises (
  RegionNotKnown,
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void associateUpdateRegion_OVER0 (
  in Region                     theRegion,
  in InteractionClassHandle     theClass)
raises (
  RegionNotKnown,
  InteractionClassNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void disassociateUpdateRegion (
  in Region             theRegion,
  in ObjectId           theObject)
raises (
  RegionNotKnown,
  ObjectNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void disassociateUpdateRegion_OVER0 (
  in Region                     theRegion,
  in InteractionClassHandle     theClass)
raises (
```

```
  RegionNotKnown,
  InteractionClassNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 7.5
void modifyRegion (
  in Region     theRegion,
  in ExtentSet  theExtents)
raises (
  RegionNotKnown,
  InvalidExtents,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// 7.6
void deleteRegion (
  in Region     theRegion)
raises (
  RegionNotKnown,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);


////////////////////////////////

                // the following services are not in the I/F spec
                // they are here for informational purposes only

////////////////////////////////////////
// RTI Support & Operational Services //
////////////////////////////////////////

// RSOS.1
ObjectClassHandle getObjectClassHandle (
  in ObjectClassName    theClass)
raises (
  NameNotFound,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.2
ObjectClassName getObjectClassName (
  in ObjectClassHandle  theClass)
raises (
  ObjectClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.3
AttributeHandle getAttributeHandle (
  in AttributeName      theAttribute,
  in ObjectClassHandle  theClass)
raises (
  ObjectClassNotDefined,
  NameNotFound,
```

```
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.4
AttributeName getAttributeName (
  in AttributeHandle    theAttribute,
  in ObjectClassHandle  theClass)
raises (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.5
InteractionClassHandle getInteractionClassHandle (
  in InteractionClassName       theInteraction)
raises (
  NameNotFound,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.6
InteractionClassName getInteractionClassName (
  in InteractionClassHandle     theInteraction)
raises (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.7
ParameterHandle getParameterHandle (
  in ParameterName              theParameter,
  in InteractionClassHandle     theClass)
raises (
  InteractionClassNotDefined,
  NameNotFound,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.8
ParameterName getParameterName (
  in ParameterHandle            theParameter,
  in InteractionClassHandle     theClass)
raises (
  InteractionClassNotDefined,
  InteractionParameterNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.9
SpaceHandle getSpaceHandle (
  in SpaceName  theSpace)
raises (
  NameNotFound,
  FederateNotExecutionMember,
```

```
  RTIinternalError,
  UnimplementedService);

// RSOS.10
SpaceName getSpaceName (
  in SpaceHandle        theSpace)
raises (
  SpaceNotDefined,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.11
void turnRegulationOn ()
raises (
  FederationTimeAlreadyPassed,
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

FederationTime turnRegulationOnNow ()
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

void turnRegulationOff ()
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.12
void setTimeConstrained (
  in RTI_Boolean      state)
raises (
  FederateNotExecutionMember,
  RTIinternalError,
  UnimplementedService);

// RSOS.13
RTI_boolean tick ()
raises (
  RTIinternalError,
  UnimplementedService);

RTI_boolean tick_OVER0 (
  in TickTime   minimum,
  in TickTime   maximum)
raises (
  RTIinternalError,
  UnimplementedService);

};

#endif
```

```
//File: federateAmbServices.idl

#ifndef __federateAmbServices
#define __federateAmbServices

#include "rtiTypes.idl"

////////////////////////////////////
// Federation Management Services //
////////////////////////////////////

interface federateAmbassador: rtiTypes {

// 2.6
void initiatePause (
  in PauseLabel label)
raises (
  FederateAlreadyPaused,
  FederateInternalError);

// 2.9
void initiateResume ( )
raises (
  FederateNotPaused,
  FederateInternalError);

// 2.12
void initiateFederateSave (
  in SaveLabel label)
raises (
  UnableToPerformSave,
  FederateInternalError);

void initiateFederateSave_OVER0 (
  in SaveLabel          label,
  in FederationTime     theTime)
raises (
  UnableToPerformSave,
  InvalidFederationTime,
  FederateInternalError);

// 2.16
void initiateRestore (
  in SaveLabel label)
raises (
  SpecifiedSaveLabelDoesNotExist,
  CouldNotInitiateRestore,
  FederateInternalError);


//////////////////////////////////////
// Declaration Management Services //
//////////////////////////////////////

// 3.5
void startUpdates (
  in ObjectClassHandle  theClass,
  in AttributeHandleSet attributeList)
raises (
  ObjectClassNotPublished,
  AttributeNotPublished,
```

```
  FederateInternalError);

void stopUpdates (
  in ObjectClassHandle  theClass,
  in AttributeHandleSet attributeList)
raises (
  ObjectClassNotPublished,
  AttributeNotPublished,
  FederateInternalError);

// 3.6
void startInteractionGeneration (
  in InteractionClassHandle     theClass)
raises (
  InteractionClassNotPublished,
  FederateInternalError);

void stopInteractionGeneration (
  in InteractionClassHandle     theClass)
raises (
  InteractionClassNotPublished,
  FederateInternalError);

//////////////////////////////
// Object Management Services //
//////////////////////////////

// 4.4
void discoverObject (
  in ObjectId                   theObject,
  in ObjectClassHandle          theClass,
  in FederationTime             theTime,
  in UserSuppliedTag            theTag,
  in EventRetractionHandle      theHandle)
raises (
  CouldNotDiscover,
  ObjectClassNotKnown,
  InvalidFederationTime,
  FederateInternalError);

// 4.5
void reflectAttributeValues (
  in ObjectId                           objectId,
  in AttributeHandleValuePairSet        attributeValueList,
  in FederationTime                     theTime,
  in UserSuppliedTag                    theTag,
  in EventRetractionHandle              theHandle)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  InvalidFederationTime,
  FederateInternalError);

// 4.7
void receiveInteraction (
  in InteractionClassHandle             theInteraction,
  in ParameterHandleValuePairSet        parameters,
  in FederationTime                     theTime,
  in UserSuppliedTag                    theTag,
  in EventRetractionHandle              theHandle)
raises (
```

```
  InteractionClassNotKnown,
  InteractionParameterNotKnown,
  InvalidFederationTime,
  FederateInternalError);

// 4.9
void removeObject (
  in ObjectId                   theObject,
  in ObjectRemovalReason        theReason,
  in FederationTime             theTime,
  in UserSuppliedTag         theTag,
  in EventRetractionHandle      theHandle)
raises (
  ObjectNotKnown,
  InvalidFederationTime,
  FederateInternalError);

void removeObject_OVER (
  in ObjectId                   theObject,
  in ObjectRemovalReason        theReason,
  in FederationTime             theTime,
  in UserSuppliedTag         theTag)
raises (
  ObjectNotKnown,
  InvalidFederationTime,
  FederateInternalError);

// 4.15
void provideAttributeValueUpdate (
  in ObjectId           theObject,
  in AttributeHandleSet attributeList)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError);

// 4.17
void reflectRetraction (
  in EventRetractionHandle      theHandle)
raises (
  EventNotKnown,
  FederateInternalError);

//////////////////////////////////
// Ownership Management Services //
//////////////////////////////////

// 5.2
void requestAttributeOwnershipAssumption (
  in ObjectId                   theObject,
  in AttributeHandleSet         offeredAttributes,
  in UserSuppliedTag            theTag,
  out AttributeHandleSet        requestedAttributes)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  AttributeAlreadyOwned,
  FederateInternalError);

// 5.3
void attributeOwnershipDivestitureNotification (
```

```
  in ObjectId           theObject,
  in AttributeHandleSet releasedAttributes)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError);

// 5.4
void attributeOwnershipAcquisitionNotification (
  in ObjectId           theObject,
  in AttributeHandleSet securedAttributes)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError);

// 5.6
void requestAttributeOwnershipRelease (
  in ObjectId                   theObject,
  in AttributeHandleSet         candidateAttributes,
  in UserSuppliedTag            theTag,
  out AttributeHandleSet        releasedAttributes)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError);

// 5.8
void informAttributeOwnership (
  in ObjectId                   theObject,
  in AttributeHandle            theAttribute,
  in FederateHandle             theOwner)
raises (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError);


//////////////////////////////
// Time Management Services //
//////////////////////////////

// 6.10
void timeAdvanceGrant (
  in FederationTime theTime)
raises (
  InvalidFederationTime,
  TimeAdvanceNotInProgress,
  FederationTimeAlreadyPassed,
  FederateInternalError);

//////////////////////////////////////////
// Data Distribution Management Services //
//////////////////////////////////////////

// 7.4
void changeThresholds (
  in Region             theRegion,
  in ThresholdSet       theThresholds)
raises (
  RegionNotKnown,
```

```
  FederateInternalError);

};

#endif
```

9 HLA C++ Application Programmer's Interface

```
//File baseTypes.hh


#ifndef NULL
#define NULL (0)
#endif

typedef unsigned short UShort;
typedef short          Short;
typedef unsigned long  ULong;
typedef long           Long;
typedef double         Double;
typedef float          Float;

enum Boolean {
    RTI_FALSE = 0,
    RTI_TRUE};

class Exception {
public:
  RTI::ULong _serial;
  char *_reason;
  const char *_name;
  Exception (const char *reason);
  Exception (RTI::ULong serial, const char *reason=NULL);
  Exception (const Exception &toCopy);
  virtual ~Exception ();
  Exception & operator = (const Exception &);
  friend ostream& operator<< (ostream &, Exception *);
};

#define RTI_EXCEPT(A)   \
class A : public RTI::Exception {       \
public: \
  static const char *_ex;  \
  A (const char *reason) : Exception (reason)  { _name = _ex;
}   \
  A (RTI::ULong serial, const char *reason=NULL)        \
    : Exception (serial, reason)  { _name = _ex; }        \
  A (const RTI::Exception &toCopy) : Exception(toCopy) {
_name = _ex; } \
};
```

```
//File: RTItypes.hh


//                RTI Parameter Passing Memory Conventions
//
// C1  In parameter by value.
// C2  Out parameter by reference.
// C3  Function return by value.
// C4  In parameter by const reference.  Caller provides memory.
//     Caller may free memory or overwrite it upon completion of
//     the call.  Callee must copy during the call anything it
//     wishes to save beyond completion of the call.  Parameter
//     type must define const accessor methods.
// C5  Out parameter by reference.  Caller provides reference to object.
//     Callee constructs an instance on the heap (new) and returns.
//     The caller destroys the instance (delete) at its leisure.
// C6  Function return by reference.  Callee constructs an instance on
//     the heap (new) and returns a reference.  The caller destroys the
//     instance (delete) at its leisure.

#define MAX_FEDERATION                20
#define MAX_FEDERATE                  20
#define MAX_FEDERATION_NAME_LENGTH    64
#define MAX_FEDERATE_NAME_LENGTH      64
#define MAX_OBJECT_CLASSES            100
#define MAX_INTERACTION_CLASSES       100
#define MAX_ATTRIBUTES_PER_CLASS      50
#define MAX_PARAMETERS_PER_CLASS      50
#define MAX_BYTES_PER_VALUE           512
#define PRIVILEGE_TO_DELETE_HANDLE    0
#define ROOT_OBJECT_CLASS_HANDLE      0
#define ROOT_INTERACTION_CLASS_HANDLE 0
#define ROOT_OBJECT_CLASS_NAME        "root"
#define ROOT_INTERACTION_CLASS_NAME   "root"
#define PRIVILEGE_TO_DELETE_NAME      "privilegeToDelete"
#define MAX_USER_TAG_LENGTH           16
#define RTI_VERSION                   "1.0R2"

#define EPSILON (1.0e-9)
//as defined for IEEE 754
//see <math.h>
#define POSITIVE_INFINITY (HUGE_VAL)

RTI_EXCEPT(ArrayIndexOutOfBounds)
RTI_EXCEPT(AttributeAlreadyOwned)
RTI_EXCEPT(AttributeNotDefined)
RTI_EXCEPT(AttributeNotKnown)
RTI_EXCEPT(AttributeNotOwned)
RTI_EXCEPT(AttributeNotPublished)
RTI_EXCEPT(AttributeNotSubscribed)
RTI_EXCEPT(ConcurrentAccessAttempted)
RTI_EXCEPT(CouldNotDiscover)
RTI_EXCEPT(CouldNotOpenFED)
RTI_EXCEPT(CouldNotRestore)
RTI_EXCEPT(DeletePrivilegeNotHeld)
RTI_EXCEPT(ErrorReadingFED)
RTI_EXCEPT(EventNotKnown)
RTI_EXCEPT(FederateAlreadyPaused)
RTI_EXCEPT(FederateAlreadyExecutionMember)
RTI_EXCEPT(FederateDoesNotExist)
RTI_EXCEPT(FederateInternalError)
```

```
RTI_EXCEPT(FederateNameAlreadyInUse)
RTI_EXCEPT(FederateNotExecutionMember)
RTI_EXCEPT(FederateNotPaused)
RTI_EXCEPT(FederateOwnsAttributes)
RTI_EXCEPT(FederatesCurrentlyJoined)
RTI_EXCEPT(FederationAlreadyPaused)
RTI_EXCEPT(FederationExecutionAlreadyExists)
RTI_EXCEPT(FederationExecutionDoesNotExist)
RTI_EXCEPT(FederationNotPaused)
RTI_EXCEPT(FederationTimeAlreadyPassed)
RTI_EXCEPT(HandleValuePairMaximumExceeded)
RTI_EXCEPT(IDsupplyExhausted)
RTI_EXCEPT(InteractionClassNotDefined)
RTI_EXCEPT(InteractionClassNotKnown)
RTI_EXCEPT(InteractionClassNotPublished)
RTI_EXCEPT(InteractionParameterNotDefined)
RTI_EXCEPT(InteractionParameterNotKnown)
RTI_EXCEPT(InvalidDivestitureCondition)
RTI_EXCEPT(InvalidExtents)
RTI_EXCEPT(InvalidFederationTime)
RTI_EXCEPT(InvalidLookahead)
RTI_EXCEPT(InvalidObjectID)
RTI_EXCEPT(InvalidOrderType)
RTI_EXCEPT(InvalidResignAction)
RTI_EXCEPT(InvalidRetractionHandle)
RTI_EXCEPT(InvalidTransportType)
RTI_EXCEPT(MemoryExhausted)
RTI_EXCEPT(NameNotFound)
RTI_EXCEPT(NoPauseRequested)
RTI_EXCEPT(NoResumeRequested)
RTI_EXCEPT(ObjectClassNotDefined)
RTI_EXCEPT(ObjectClassNotKnown)
RTI_EXCEPT(ObjectClassNotPublished)
RTI_EXCEPT(ObjectClassNotSubscribed)
RTI_EXCEPT(ObjectNotKnown)
RTI_EXCEPT(ObjectAlreadyRegistered)
RTI_EXCEPT(RegionNotKnown)
RTI_EXCEPT(RestoreNotRequested)
RTI_EXCEPT(RTIinternalError)
RTI_EXCEPT(SpaceNotDefined)
RTI_EXCEPT(SaveInProgress)
RTI_EXCEPT(SaveNotInitiated)
RTI_EXCEPT(SpecifiedSaveLabelDoesNotExist)
RTI_EXCEPT(TimeAdvanceAlreadyInProgress)
RTI_EXCEPT(TimeAdvanceWasNotInProgress)
RTI_EXCEPT(TooManyIDsRequested)
RTI_EXCEPT(UnableToPerformSave)
RTI_EXCEPT(UnimplementedService)
RTI_EXCEPT(UnknownLabel)
RTI_EXCEPT(ValueCountExceeded)
RTI_EXCEPT(ValueLengthExceeded)

enum OrderType {
    RECEIVE = 1,
    TIMESTAMP
};

enum OwnershipDivestitureCondition {
    NEGOTIATED = 1,
    UNCONDITIONAL
};
```

```
enum ObjectRemovalReason {
    OUT_OF_REGION = 1,
    OBJECT_DELETED,
    NO_LONGER_SUBSCRIBED
};

enum TransportType {
    RELIABLE = 1,
    BEST_EFFORT
};

enum ResignAction {
    RELEASE_ATTRIBUTES = 1,
    DELETE_OBJECTS,
    DELETE_OBJECTS_AND_RELEASE_ATTRIBUTES,
    NO_ACTION
};

class FederateAmbassador;

typedef FederateAmbassador *FederateAmbassadorPtr;

typedef RTI::UShort ObjectClassHandle;

typedef RTI::UShort InteractionClassHandle;

typedef RTI::UShort Handle;

typedef Handle AttributeHandle;

typedef Handle ParameterHandle;

typedef RTI::UShort FederateHandle;

typedef RTI::ULong FederateID;

typedef RTI::ULong ObjectID;

typedef RTI::ULong UniqueID;

class HandleValuePairSet {
public:
  virtual ~HandleValuePairSet() { ; }

  virtual ULong     // returned C3
  size() const = 0; // Cardinality of the set.

  virtual Handle // returned C3
  getHandle(RTI::ULong i) const
  throw (
    ArrayIndexOutOfBounds) = 0;

  virtual void getValue(RTI::ULong i,
    char *buff,               // supplied C4
    ULong &valueLength) const // returned C2
  throw (
    ArrayIndexOutOfBounds) = 0;

  virtual void add(    // Add new value.
    Handle h,          // supplied C1
```

```
    const char *buff,  // supplied C4
    ULong valueLength) // supplied C1
  throw (
    ValueLengthExceeded,
    ValueCountExceeded) = 0;

  virtual void moveFrom(const HandleValuePairSet& ahvp,
                        RTI::ULong &i) // supplied C1
  throw (
    ValueCountExceeded,
    ArrayIndexOutOfBounds) = 0;

  virtual void empty() = 0; // Empty the Set without deallocating space.

  virtual inline RTI::ULong start() const = 0;
  virtual inline RTI::ULong valid(RTI::ULong i) const = 0;
  virtual inline RTI::ULong next(RTI::ULong i) const = 0;
};

typedef HandleValuePairSet AttributeHandleValuePairSet;

class AttributeSetFactory {
public:
  static AttributeHandleValuePairSet  // returned C6
  *create(                            // Create set with space for
                                      // specified number of attributes.
    ULong count)                      // supplied C1
  throw (
    MemoryExhausted,
    ValueCountExceeded);
};

class AttributeHandleSet {
public:
  AttributeHandleSet();
  ULong     // returned C3
  size() const; // Cardinality of the set.

  AttributeHandle // returned C3
  getHandle(              // Get ith handle.
    ULong i) const        // supplied C1
  throw (
    ArrayIndexOutOfBounds);

  void add(    // Add new value.
    AttributeHandle h) // supplied C1
  throw (
    AttributeNotDefined);

  void empty(); // Empty the Set

  Boolean isEmpty();  //is set empty?
  Boolean isMember(AttributeHandle h) const;
  AttributeHandleSet setUnion(AttributeHandleSet &);
  AttributeHandleSet setIntersection(AttributeHandleSet &);
  AttributeHandleSet removeSetIntersection(AttributeHandleSet &);

  ULong encodedLength() const;
  void encode(char *buff) const;
  static AttributeHandleSet *decode(const char *buff)
    throw(ValueCountExceeded);
```

```
  friend ostream& operator << (ostream&, AttributeHandleSet &);

  private:
  unsigned short _size;
  unsigned long _words[(MAX_ATTRIBUTES_PER_CLASS-1)/32+1];
};


class AttributeHandleSetFactory {
public:
  static AttributeHandleSet // returned C6
  *create(                  // Create set with space for specified
                            // number of AttributeHandles.
    ULong count)            // supplied C1
  throw(
    MemoryExhausted,
    ValueCountExceeded);
};


typedef RTI::UShort ObjectIDcount;

typedef RTI::Double FederationTimeDelta;

typedef RTI::Double FederationTime;

typedef RTI::Double TickTime;

class FederateHandleSet {
public:
  virtual ULong     // returned C3
  size() const = 0; // Cardinality of the set.

  virtual FederateHandle // returned C3
  getHandle(             // Get ith handle.
    ULong i) const       // supplied C1
  throw (
    ArrayIndexOutOfBounds) = 0;

  virtual void add(   // Add new value.
    FederateHandle h) // supplied C1
  throw (
    ValueCountExceeded) = 0;

  virtual void empty() = 0; // Empty the set without deallocating space.
};

class FederateHandleSetFactory {
public:
  static FederateHandleSet // returned C6
  *create(                 // Create set with space for specified
                           // number of FederateHandles.
    ULong count)           // supplied C1
  throw (
    MemoryExhausted,
    ValueCountExceeded);
};

struct EventRetractionHandle_s {
    FederationTime  theEventTime;
```

```
    UniqueID        theSerialNumber;
    FederateHandle  sendingFederate;
};
typedef struct EventRetractionHandle_s EventRetractionHandle;

// All char * declarations should use null terminated strings

typedef char * FederationExecutionName;

typedef char * FederateName;

typedef char * FileName;

typedef char * PauseLabel;

typedef char * SaveLabel;

typedef char * UserSuppliedTag;

typedef char * ObjectClassName;

typedef char * AttributeName;

typedef char * InteractionClassName;

typedef char * ParameterName;

typedef HandleValuePairSet ParameterHandleValuePairSet;

class ParameterSetFactory {
public:
  static ParameterHandleValuePairSet // returned C6
  *create(                           // Create set with space for
                                     // specified number of attributes.
    ULong count)                     // supplied C1
  throw (
    MemoryExhausted,
    ValueCountExceeded);
};

struct range_struct_s {
    RTI::ULong first;
    RTI::ULong last;
};
typedef struct range_struct_s Range;


struct Extents_s {
    RTI::ULong _length;
    Range* _buffer;
};
typedef struct Extents_s Extents;


struct ExtentSet_s {
    RTI::ULong _length;
    Extents* _buffer;
};
typedef struct ExtentSet_s ExtentSet;
```

```
typedef RTI::ULong Region;


struct Threshold_s {
    RTI::ULong value;
};
typedef Threshold_s Threshold;


struct ThresholdSet_s {
    RTI::ULong _length;
    Threshold* _buffer;
};
typedef struct ThresholdSet_s ThresholdSet;

typedef RTI::Long SpaceHandle;

typedef char * SpaceName;
```

```
//File: RTIambServices.hh


//                RTI Parameter Passing Memory Conventions
//
// C1  In parameter by value.
// C2  Out parameter by reference.
// C3  Function return by value.
// C4  In parameter by const reference.  Caller provides memory.
//     Caller may free memory or overwrite it upon completion of
//     the call.  Callee must copy during the call anything it
//     wishes to save beyond completion of the call.  Parameter
//     type must define const accessor methods.
// C5  Out parameter by reference.  Caller provides reference to object.
//     Callee constructs an instance on the heap (new) and returns.
//     The caller destroys the instance (delete) at its leisure.
// C6  Function return by reference.  Callee constructs an instance on
//     the heap (new) and returns a reference.  The caller destroys the
//     instance (delete) at its leisure.

typedef FederateAmbassador *FederateAmbassadorPtr;

///////////////////////////////////
// Federation Management Services //
///////////////////////////////////

// 2.1
void createFederationExecution (
  const FederationExecutionName executionName) // supplied C4
throw (
  FederationExecutionAlreadyExists,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 2.2
void destroyFederationExecution (
  const FederationExecutionName executionName) // supplied C4
throw (
  FederatesCurrentlyJoined,
  FederationExecutionDoesNotExist,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 2.3
FederateHandle                                                // returned C3
joinFederationExecution (
  const FederateName            yourName,                     // supplied C4
  const FederationExecutionName executionName,                // supplied C4
        FederateAmbassadorPtr   federateAmbassadorReference) // supplied C1
throw (
  FederateAlreadyExecutionMember,
  FederationExecutionDoesNotExist,
  CouldNotOpenFED,
  ErrorReadingFED,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 2.4
void resignFederationExecution (
  ResignAction theAction) // supplied C1
throw (
```

```
  FederateOwnsAttributes,
  FederateNotExecutionMember,
  FederationExecutionDoesNotExist,
  InvalidResignAction,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 2.5
void requestPause (       // not implemented in F.0
  const PauseLabel label) // supplied C4
throw (
  FederationAlreadyPaused,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.7
void pauseAchieved (      // not implemented in F.0
  const PauseLabel label) // supplied C4
throw (
  UnknownLabel,
  NoPauseRequested,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.8
void requestResume () // not implemented in F.0
throw (
  FederationNotPaused,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.10
void resumeAchieved () // not implemented in F.0
throw (
  NoResumeRequested,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.11
void requestFederationSave (    // not implemented in F.0
  const SaveLabel      label,   // supplied C4
        FederationTime theTime) // supplied C1
throw (
  FederationTimeAlreadyPassed,
  InvalidFederationTime,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void requestFederationSave ( // not implemented in F.0
  const SaveLabel label)     // supplied C4
throw (
```

```
  SaveInProgress,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.13
void federateSaveBegun () // not implemented in F.0
throw (
  SaveNotInitiated,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void federateSaveBegun (  // not implemented in F.0
  FederationTime theTime) // supplied C1
throw (
  SaveNotInitiated,
  InvalidFederationTime,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.14
void federateSaveAchieved () // not implemented in F.0
throw (
  SaveNotInitiated,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void federateSaveNotAchieved () // not implemented in F.0
throw (
  SaveNotInitiated,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.15
void requestRestore (    // not implemented in F.0
  const SaveLabel label) // supplied C4
throw (
  SpecifiedSaveLabelDoesNotExist,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 2.17
void restoreAchieved (   // not implemented in F.0
  const SaveLabel label) // supplied C4
throw (
  UnknownLabel,
  RestoreNotRequested,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
```

```
  UnimplementedService);

void restoreNotAchieved ( // not implemented in F.0
  const SaveLabel label)  // supplied C4
throw (
  UnknownLabel,
  RestoreNotRequested,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

////////////////////////////////////
// Declaration Management Services //
////////////////////////////////////

// 3.1
void publishObjectClass (
        ObjectClassHandle   theClass,      // supplied C1
  const AttributeHandleSet& attributeList) // supplied C4
throw (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void unpublishObjectClass (
  ObjectClassHandle theClass) // supplied C1
throw (
  ObjectClassNotDefined,
  FederateOwnsAttributes,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 3.2
void publishInteractionClass (
  InteractionClassHandle theInteraction) // supplied C1
throw (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void unpublishInteractionClass (
  InteractionClassHandle theInteraction) // supplied C1
throw (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 3.3
void subscribeObjectClassAttribute (
        ObjectClassHandle   theClass,      // supplied C1
  const AttributeHandleSet& attributeList) // supplied C4
throw (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
```

```
  ConcurrentAccessAttempted,
  RTIinternalError);

void subscribeObjectClassAttribute ( // not implemented in F.0
  ObjectClassHandle theClass,        // supplied
  AttributeHandle   theAttribute,    // supplied
  Region            theRegion)       // supplied
throw (
  ObjectClassNotDefined,
  AttributeNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void unsubscribeObjectClassAttribute (
  ObjectClassHandle theClass) // supplied C1
throw (
  ObjectClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void unsubscribeObjectClassAttribute ( // not implemented in F.0
  ObjectClassHandle theClass,          // supplied
  Region            theRegion)         // supplied
throw (
  ObjectClassNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 3.4
void subscribeInteractionClass (
  InteractionClassHandle theClass) // supplied C1
throw (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void subscribeInteractionClass (    // not implemented in F.0
  InteractionClassHandle theClass,  // supplied
  Region                 theRegion) // supplied
throw (
  InteractionClassNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void unsubscribeInteractionClass (
  InteractionClassHandle theClass) // supplied C1
throw (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
```

```
  RTIinternalError);

void unsubscribeInteractionClass (  // not implemented in F.0
  InteractionClassHandle theClass,  // supplied
  Region                 theRegion) // supplied
throw (
  InteractionClassNotDefined,
  RegionNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

//////////////////////////////
// Object Management Services //
//////////////////////////////

// 4.1
void requestID (
  ObjectIDcount idCount, // supplied C1
  ObjectID&     firstID, // returned C2
  ObjectID&     lastID)  // returned C2
throw (
  TooManyIDsRequested,
  IDsupplyExhausted,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.2
void registerObject (
  ObjectClassHandle theClass,  // supplied C1
  ObjectID          theObject) // supplied C1
throw (
  InvalidObjectID,
  ObjectAlreadyRegistered,
  ObjectClassNotDefined,
  ObjectClassNotPublished,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.3
EventRetractionHandle                                // returned C3
updateAttributeValues (
        ObjectID                     theObject,     // supplied C1
  const AttributeHandleValuePairSet& theAttributes, // supplied C4
        FederationTime               theTime,       // supplied C1
  const UserSuppliedTag              theTag)        // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidFederationTime,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.6
EventRetractionHandle                                 // returned C3
sendInteraction (
```

```
        InteractionClassHandle       theInteraction, // supplied C1
  const ParameterHandleValuePairSet& theParameters,  // supplied C4
        FederationTime               theTime,        // supplied C1
  const UserSuppliedTag              theTag)         // supplied C4
throw (
  InteractionClassNotPublished,
  InteractionClassNotDefined,
  InteractionParameterNotDefined,
  InvalidFederationTime,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.8
EventRetractionHandle             // returned C3
deleteObject (
        ObjectID        objectID, // supplied C1
        FederationTime  theTime,  // supplied C1
  const UserSuppliedTag theTag)   // supplied C4
throw (
  DeletePrivilegeNotHeld,
  ObjectNotKnown,
  InvalidFederationTime,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.10
void changeAttributeTransportType (
        ObjectID            theObject,     // supplied C1
  const AttributeHandleSet& theAttributes, // supplied C4
        TransportType       theType)       // supplied C1
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidTransportType,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.11
void changeAttributeOrderType (
        ObjectID            theObject,     // supplied C1
  const AttributeHandleSet& theAttributes, // supplied C4
        OrderType           theType)       // supplied C1
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidOrderType,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.12
void changeInteractionTransportType (
  InteractionClassHandle theClass, // supplied C1
  TransportType          theType)  // supplied C1
throw (
  InteractionClassNotDefined,
```

```
  InteractionClassNotPublished,
  InvalidTransportType,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.13
void changeInteractionOrderType (
  InteractionClassHandle theClass, // supplied C1
  OrderType              theType)  // supplied C1
throw (
  InteractionClassNotDefined,
  InteractionClassNotPublished,
  InvalidOrderType,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.14
void requestObjectAttributeValueUpdate (
        ObjectID            theObject,     // supplied C1
  const AttributeHandleSet& theAttributes) // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void requestObjectAttributeValueUpdate (
  ObjectID theObject) // supplied C1
throw (
  ObjectNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void requestClassAttributeValueUpdate (
        ObjectClassHandle   theClass,      // supplied C1
  const AttributeHandleSet& theAttributes) // supplied C4
throw (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void requestClassAttributeValueUpdate (
  ObjectClassHandle theClass) // supplied C1
throw (
  ObjectClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 4.16
void retract (
  EventRetractionHandle theHandle) // supplied C1
throw (
  InvalidRetractionHandle,
  FederateNotExecutionMember,
```

```
  ConcurrentAccessAttempted,
  RTIinternalError);

//////////////////////////////////
// Ownership Management Services //
//////////////////////////////////

// 5.1
void requestAttributeOwnershipDivestiture (
        ObjectID                      theObject,     // supplied C1
  const AttributeHandleSet&           theAttributes, // supplied C4
        OwnershipDivestitureCondition theCondition,  // supplied C1
  const UserSuppliedTag               theTag)        // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidDivestitureCondition,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

void requestAttributeOwnershipDivestiture (
        ObjectID                      theObject,     // supplied C1
  const AttributeHandleSet&           theAttributes, // supplied C4
        OwnershipDivestitureCondition theCondition,  // supplied C1
  const UserSuppliedTag               theTag,        // supplied C4
  const FederateHandleSet&            theCandidates) // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotOwned,
  InvalidDivestitureCondition,
  FederateDoesNotExist,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 5.5
void requestAttributeOwnershipAcquisition (
        ObjectID            theObject,         // supplied C1
  const AttributeHandleSet& desiredAttributes, // supplied C4
  const UserSuppliedTag     theTag)            // supplied C4
throw (
  ObjectNotKnown,
  ObjectClassNotPublished,
  ObjectClassNotSubscribed,
  AttributeNotDefined,
  AttributeNotPublished,
  AttributeNotSubscribed,
  FederateOwnsAttributes,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 5.7
void queryAttributeOwnership (
  ObjectID        theObject,    // supplied C1
  AttributeHandle theAttribute) // supplied C1
throw (
  ObjectNotKnown,
```

```
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 5.9
RTI::Boolean                      // returned C3
attributeIsOwnedByFederate (
  ObjectID        theObject,      // supplied C1
  AttributeHandle theAttribute)   // supplied C1
throw (
  ObjectNotKnown,
  AttributeNotDefined,
  AttributeNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

//////////////////////////////
// Time Management Services //
//////////////////////////////

// 6.1
FederationTime // returned C3
requestFederationTime ()
throw (
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.2
FederationTime // returned C3
requestLBTS ()
throw (
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.3
FederationTime // returned C3
requestFederateTime ()
throw (
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.4
FederationTime // returned C3
requestMinNextEventTime ()
throw (
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.5
void setLookahead (
  FederationTimeDelta theLookahead) // supplied C1
throw (
  InvalidLookahead,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
```

```
  RTIinternalError);

// 6.6
FederationTimeDelta // returned C3
requestLookahead ()
throw (
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.7
void timeAdvanceRequest (
  FederationTime theTime) // supplied C1
throw (
  InvalidFederationTime,
  TimeAdvanceAlreadyInProgress,
  FederationTimeAlreadyPassed,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.8
void nextEventRequest (
  FederationTime theTime) // supplied C1
throw (
  TimeAdvanceAlreadyInProgress,
  FederationTimeAlreadyPassed,
  InvalidFederationTime,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 6.9
void flushQueueRequest (
  FederationTime theTime) // supplied C1
throw (
  InvalidFederationTime,
  TimeAdvanceAlreadyInProgress,
  FederationTimeAlreadyPassed,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

//////////////////////////////////
// Data Distribution Management //
//////////////////////////////////

// 7.1
void createUpdateRegion ( // not implemented in F.0
  SpaceHandle theSpace,   // supplied
  ExtentSet&  theExtents, // supplied
  Region&     theRegion)  // returned
throw (
  SpaceNotDefined,
  InvalidExtents,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 7.2
```

```
void createSubscriptionRegion ( // not implemented in F.0
  SpaceName  theSpace,          // supplied
  ExtentSet& theExtents,        // supplied
  Region&    theRegion)         // returned
throw (
  SpaceNotDefined,
  InvalidExtents,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 7.3
void associateUpdateRegion (         // not implemented in F.0
  Region              theRegion,     // supplied
  ObjectID            theObject,     // supplied
  AttributeHandleSet& theAttributes) // supplied
throw (
  RegionNotKnown,
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void disassociateUpdateRegion (      // not implemented in F.0
  Region              theRegion,     // supplied
  ObjectID            theObject,     // supplied
  AttributeHandleSet& theAttributes) // supplied
throw (
  RegionNotKnown,
  ObjectNotKnown,
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void associateUpdateRegion (        // not implemented in F.0
  Region                 theRegion, // supplied
  InteractionClassHandle theClass)  // supplied
throw (
  RegionNotKnown,
  InteractionClassNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

void disassociateUpdateRegion (     // not implemented in F.0
  Region                 theRegion, // supplied
  InteractionClassHandle theClass)  // supplied
throw (
  RegionNotKnown,
  InteractionClassNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);
```

```
// 7.5
void modifyRegion (      // not implemented in F.0
  Region     theRegion,  // supplied
  ExtentSet& theExtents) // supplied
throw (
  RegionNotKnown,
  InvalidExtents,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 7.6
void deleteRegion ( // not implemented in F.0
  Region theRegion) // supplied
throw (
  RegionNotKnown,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

//////////////////////////
// RTI Support Services //
//////////////////////////

// 8.1
ObjectClassHandle                // returned C3
getObjectClassHandle (
  const ObjectClassName theName) // supplied C4
throw (
  NameNotFound,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.2
ObjectClassName                // returned C6
getObjectClassName (
  ObjectClassHandle theHandle) // supplied C1
throw (
  ObjectClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.3
AttributeHandle                       // returned C3
getAttributeHandle (
  const AttributeName     theName,    // supplied C4
        ObjectClassHandle whichClass) // supplied C1
throw (
  ObjectClassNotDefined,
  NameNotFound,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.4
AttributeName                   // returned C6
getAttributeName (
```

```
  AttributeHandle   theHandle,  // supplied C1
  ObjectClassHandle whichClass) // supplied C1
throw (
  ObjectClassNotDefined,
  AttributeNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.5
InteractionClassHandle                // returned C1
getInteractionClassHandle (
  const InteractionClassName theName) // supplied C4
throw (
  NameNotFound,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.6
InteractionClassName                // returned C6
getInteractionClassName (
  InteractionClassHandle theHandle) // supplied C1
throw (
  InteractionClassNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.7
ParameterHandle                           // returned C1
getParameterHandle (
  const ParameterName          theName,    // supplied C4
        InteractionClassHandle whichClass) // supplied C1
throw (
  InteractionClassNotDefined,
  NameNotFound,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.8
ParameterName                        // returned C6
getParameterName (
  ParameterHandle        theHandle,  // supplied C1
  InteractionClassHandle whichClass) // supplied C1
throw (
  InteractionClassNotDefined,
  InteractionParameterNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.9
SpaceHandle                // returned C1
getSpaceHandle (           // not implemented in F.0
  const SpaceName theName) // supplied C4
throw (
  NameNotFound,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
```

```
  RTIinternalError,
  UnimplementedService);

// 8.10
SpaceName                      // returned C6
getSpaceName (                 // not implemented in F.0
  const SpaceHandle theHandle) // supplied C4
throw (
  SpaceNotDefined,
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError,
  UnimplementedService);

// 8.11
void RTI::RTIambassador::turnRegulationOn ()
throw (
  RTI::FederationTimeAlreadyPassed,
  RTI::FederateNotExecutionMember,
  RTI::ConcurrentAccessAttempted,
  RTI::RTIinternalError);

RTI::FederationTime                        // Returned C3
RTI::RTIambassador::turnRegulationOnNow ()
throw (
  RTI::FederateNotExecutionMember,
  RTI::ConcurrentAccessAttempted,
  RTI::RTIinternalError);

void RTI::RTIambassador::turnRegulationOff ()
throw (
  RTI::FederateNotExecutionMember,
  RTI::ConcurrentAccessAttempted,
  RTI::RTIinternalError);

// 8.12
void setTimeConstrained (
  RTI::Boolean state) // supplied C1
throw (
  FederateNotExecutionMember,
  ConcurrentAccessAttempted,
  RTIinternalError);

// 8.13
RTI::Boolean // returned C3
tick ()
throw (
  ConcurrentAccessAttempted,
  RTIinternalError);

RTI::Boolean        // returned C3
tick (
  TickTime minimum, // supplied C1
  TickTime maximum) // supplied C1
throw (
  ConcurrentAccessAttempted,
  RTIinternalError);

RTIambassador()
throw (
  MemoryExhausted,
```

```
  RTIinternalError);

~RTIambassador()
throw (RTIinternalError);
```

```
//File: federateAmbServices.hh


//                RTI Parameter Passing Memory Conventions
//
// C1  In parameter by value.
// C2  Out parameter by reference.
// C3  Function return by value.
// C4  In parameter by const reference.  Caller provides memory.
//     Caller may free memory or overwrite it upon completion of
//     the call.  Callee must copy during the call anything it
//     wishes to save beyond completion of the call.  Parameter
//     type must define const accessor methods.
// C5  Out parameter by reference.  Caller provides reference to object.
//     Callee constructs an instance on the heap (new) and returns.
//     The caller destroys the instance (delete) at its leisure.
// C6  Function return by reference.  Callee constructs an instance on
//     the heap (new) and returns a reference.  The caller destroys the
//     instance (delete) at its leisure.

////////////////////////////////////
// Federation Management Services //
////////////////////////////////////

// 2.6
virtual void initiatePause (
  const PauseLabel label) // supplied C4
throw (
  FederateAlreadyPaused,
  FederateInternalError) = 0;

// 2.9
virtual void initiateResume ()
throw (
  FederateNotPaused,
  FederateInternalError) = 0;

// 2.12
virtual void initiateFederateSave (
  const SaveLabel label) // supplied C4
throw (
  UnableToPerformSave,
  FederateInternalError) = 0;

virtual void initiateFederateSave (
  const SaveLabel      label,   // supplied C4
        FederationTime theTime) // supplied C1
throw (
  InvalidFederationTime,
  UnableToPerformSave,
  FederateInternalError) = 0;

// 2.16
virtual void initiateRestore (
  const SaveLabel label) // supplied C4
throw (
  SpecifiedSaveLabelDoesNotExist,
  CouldNotRestore,
  FederateInternalError) = 0;

//////////////////////////////////////
```

```
// Declaration Management Services //
////////////////////////////////////

// 3.5
virtual void startUpdates (
        ObjectClassHandle   theClass,      // supplied C1
  const AttributeHandleSet& theAttributes) // supplied C4
throw (
  ObjectClassNotPublished,
  AttributeNotPublished,
  FederateInternalError) = 0;

virtual void stopUpdates (
        ObjectClassHandle   theClass,      // supplied C1
  const AttributeHandleSet& theAttributes) // supplied C4
throw (
  ObjectClassNotPublished,
  AttributeNotPublished,
  FederateInternalError) = 0;

// 3.6
virtual void startInteractionGeneration (
  InteractionClassHandle theHandle) // supplied C1
throw (
  InteractionClassNotPublished,
  FederateInternalError) = 0;

virtual void stopInteractionGeneration (
  InteractionClassHandle theHandle) // supplied C1
throw (
  InteractionClassNotPublished,
  FederateInternalError) = 0;

//////////////////////////////
// Object Management Services //
//////////////////////////////

// 4.4
virtual void discoverObject (
        ObjectID              theObject,      // supplied C1
        ObjectClassHandle     theObjectClass, // supplied C1
        FederationTime        theTime,        // supplied C1
  const UserSuppliedTag       theTag,         // supplied C4
        EventRetractionHandle theHandle)      // supplied C1
throw (
  CouldNotDiscover,
  ObjectClassNotKnown,
  InvalidFederationTime,
  FederateInternalError) = 0;

// 4.5
virtual void reflectAttributeValues (
        ObjectID                     theObject,     // supplied C1
  const AttributeHandleValuePairSet& theAttributes, // supplied C4
        FederationTime               theTime,       // supplied C1
  const UserSuppliedTag              theTag,        // supplied C4
        EventRetractionHandle        theHandle)     // supplied C1
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  InvalidFederationTime,
```

```
  FederateInternalError) = 0;

// 4.7
virtual void receiveInteraction (
        InteractionClassHandle       theInteraction, // supplied C1
  const ParameterHandleValuePairSet& theParameters,  // supplied C4
        FederationTime               theTime,        // supplied C1
  const UserSuppliedTag              theTag,         // supplied C4
        EventRetractionHandle        theHandle)      // supplied C1
throw (
  InteractionClassNotKnown,
  InteractionParameterNotKnown,
  InvalidFederationTime,
  FederateInternalError) = 0;

// 4.9
virtual void removeObject (
        ObjectID              theObject, // supplied C1
        ObjectRemovalReason   theReason, // supplied C1
        FederationTime        theTime,   // supplied C1
  const UserSuppliedTag       theTag,    // supplied C4
        EventRetractionHandle theHandle) // supplied C1
throw (
  ObjectNotKnown,
  InvalidFederationTime,
  FederateInternalError) = 0;

virtual void removeObject (
       ObjectID            theObject, // supplied C1
       ObjectRemovalReason theReason) // supplied C1
       FederationTime      theTime,   // supplied C1
 const UserSuppliedTag     theTag)    // supplied C4
throw (
  ObjectNotKnown,
  InvalidFederationTime,
  FederateInternalError) = 0;

// 4.15
virtual void provideAttributeValueUpdate (
        ObjectID            theObject,     // supplied C1
  const AttributeHandleSet& theAttributes) // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError) = 0;

// 4.17
virtual void reflectRetraction (
  EventRetractionHandle theHandle) // supplied C1
throw (
  EventNotKnown,
  FederateInternalError) = 0;

//////////////////////////////////
// Ownership Management Services //
//////////////////////////////////

// 5.2
virtual AttributeHandleSet&                            // returned C6
requestAttributeOwnershipAssumption (
        ObjectID            theObject,         // supplied C1
```

```
  const AttributeHandleSet& offeredAttributes, // supplied C4
  const UserSuppliedTag     theTag)            // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  AttributeAlreadyOwned,
  FederateInternalError) = 0;

// 5.3
virtual void attributeOwnershipDivestitureNotification (
        ObjectID            theObject,          // supplied C1
  const AttributeHandleSet& releasedAttributes) // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError) = 0;

// 5.4
virtual void attributeOwnershipAcquisitionNotification (
        ObjectID            theObject,         // supplied C1
  const AttributeHandleSet& securedAttributes) // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError) = 0;

// 5.6
virtual AttributeHandleSet&                              // returned C6
requestAttributeOwnershipRelease (
        ObjectID            theObject,           // supplied C1
  const AttributeHandleSet& candidateAttributes, // supplied C4
  const UserSuppliedTag     theTag)              // supplied C4
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError) = 0;

// 5.8
virtual void informAttributeOwnership (
  ObjectID        theObject,    // supplied C1
  AttributeHandle theAttribute, // supplied C1
  FederateHandle  theOwner)     // supplied C1
throw (
  ObjectNotKnown,
  AttributeNotKnown,
  FederateInternalError) = 0;

//////////////////////////////
// Time Management Services //
//////////////////////////////

// 6.10
virtual void timeAdvanceGrant (
  FederationTime theTime) // supplied C1
throw (
  InvalidFederationTime,
  TimeAdvanceWasNotInProgress,
  FederationTimeAlreadyPassed,
  FederateInternalError) = 0;

//////////////////////////////////
```

```
// Data Distribution Management //
/////////////////////////////////

// 7.4
virtual void changeThresholds (
  Region        theRegion,     // supplied
  ThresholdSet& theThresholds) // returned
throw (
  RegionNotKnown,
  FederateInternalError) = 0;
```

```
//File: RTI.hh

#ifndef RTI_hh
#define RTI_hh

#include <fstream.h>
#include <math.h>

struct RTIambPrivate;

class RTI {
public:

  #include "baseTypes.hh"
  #include "RTItypes.hh"

  class RTIambassador {
  public:
    #include "RTIambServices.hh"
  private:
    RTIambPrivate* privateData;
  };

  class FederateAmbassador {
  public:
    #include "federateAmbServices.hh"
  };
};

#endif
```

10 HLA Ada 95 Application Programmer's Interface

```
--
-- File: rti.1.ada
--

with System;
with RTI_Base_Types;
with Unchecked_Conversion;

-- with RTI_Basic_Lists;
package RTI is

--
-- system parameters
--
--     These are from the C++.  Probably all are not needed, but they are
--     all reproduced here.  Easier to delete than to add.
--
    Max_Federation : constant Positive := 20;
    Max_Federate : constant Positive := 20;
    Max_Object_Classes : constant Positive := 100;
    Max_Interaction_Classes : constant Positive := 100;
    Max_Attributes_Per_Class : constant Positive := 50;
    Max_Parameters_Per_Class : constant Positive := 50;
    Max_Bytes_Per_Value : constant Positive := 256;
    Max_Federation_Name_Length : constant Positive := 64;
    Max_Federate_Name_Length : constant Positive := 64;
    Privilege_To_Delete_Name : constant String := "privilegeToDelete";
    Privilege_To_Delete_Handle : constant := 0;
    Root_Object_Class_Handle : constant := 0;
    Root_Interaction_Class_Handle : constant := 0;


--
-- types used across the RTI
--

    type Simple_Handle is new RTI_Base_Types.RTI_UShort;

    type Value_Ptr is access System.Address;
    type Value_Bucket is record
       The_Length: positive;
       The_Value: Value_Ptr;
    end record;

    type Federation_Execution_Name is new RTI_Base_Types.RTI_String;
    type Federation_Handle is new Simple_Handle;

    type Federation_Time is new RTI_Base_Types.RTI_Duration;
    type Tick_Time is new RTI_Base_Types.RTI_Duration;

    type Federate_Name is new RTI_Base_Types.RTI_String;
    type Federate_Handle is new Simple_Handle;

    type Object_Id is new RTI_Base_Types.RTI_ULong;
```

```
type Object_Class_Name is new RTI_Base_Types.RTI_String;
type Object_Class_Handle is new Simple_Handle;

type Interaction_Class_Name is new RTI_Base_Types.RTI_String;
type Interaction_Class_Handle is new Simple_Handle;

type Attribute_Name is new RTI_Base_Types.RTI_String;

type Attribute_Handle is new Simple_Handle;

--
-- List Types And Declarations
--
type Node_Item;
type Node_Ptr is access all Node_Item'Class;
type Node_Item is tagged
  record
     Next : Node_Ptr := null;
  end Record;

type Attribute_Handle_Set_Element is new Node_Item
with record
    The_Attribute_Handle : Attribute_Handle;
end record;
type Attribute_Handle_Set is new Node_Ptr;


type Attribute_Handle_Value_Pair_Set_Element is
    new Node_Item
with record
    The_Attribute_Handle: Attribute_Handle;
    The_Attribute_Value: Value_Bucket;
end record;
type Attribute_Handle_Value_Pair_Set is new Node_Ptr;


type Parameter_Handle_Value_Pair_Set_Element is
    new Node_Item
with record
    The_Parameter_Handle: Attribute_Handle;
    The_Parameter_Value: Value_Bucket;
end record;
type Parameter_Handle_Value_Pair_Set is new Node_Ptr;

--
-- End List Types
--

type Parameter_Name is new RTI_Base_Types.RTI_String;
type Parameter_Handle is new Simple_Handle;

type Pause_Label is new RTI_Base_Types.RTI_String;

type Save_Label is new RTI_Base_Types.RTI_String;

type Event_Retraction_Handle is private;

-- type User_Supplied_Tag is new Value_Bucket; -- WHY?????
type User_Supplied_Tag is new RTI_Base_Types.RTI_String;
```

```
    type RTI_Boolean is (RTI_False, RTI_True);
    for RTI_Boolean use
       (RTI_False => 0,
       RTI_True => 1);

    procedure Set_Retraction_Handle -- New
      (Handle : in out Event_Retraction_Handle;
       Value : in Simple_Handle);

    function Convert_To_Integer is new Unchecked_Conversion
      (Source => RTI_Boolean,
       Target => Integer);

    function Convert_To_Rti_Boolean is new Unchecked_Conversion
      (Source => Integer,
       Target => RTI_Boolean);



private

    type Event_Retraction_Handle is new Simple_Handle;


end RTI;
```

```
--
-- File: rti_base_types.1.ada
--

with Interfaces.C;


package RTI_Base_Types is

   package C renames Interfaces.C;

--
-- base types
--
    subtype RTI_String is String;

       -- RTI_Float isn't used, is it needed?
--    subtype RTI_Float is Float;

    subtype RTI_Duration is Duration;

    subtype RTI_ULong is integer range 0 .. Integer'Last;
    subtype RTI_Long is integer;


    subtype RTI_Ushort is C.Unsigned_Short;
    subtype RTI_Short is C.Short;
    subtype RTI_Double is C.Double;
    subtype RTI_Float is C.C_Float;
    subtype RTI_Id is C.Unsigned_Long;
    -- subtype RTI_Boolean is C.Unsigned_Char; -- Huh????


end RTI_Base_Types;
```

```
--
-- File: rti_basic_lists.1.ada
--

-------------------------------------------------------------------------------
--|NAME rti_basic_lists.1.ada
--|PURPOSE
--|
--|ENVIRONMENT_DEPENDENCIES None
--|MACHINE_DEPENDENCIES designed for use on IBM PowerPC running AIX
--|DEVICE_IO_DEPENDENCIES None
--|EXECUTION_CONSTRAINTS None
--|DESIGN_CONSTRAINTS designed to function with version F0 of the RTI
--|
-------------------------------------------------------------------------------

with RTI;

package RTI_Basic_Lists is



--     type Node_Item is tagged limited private;
--     type Node_Ptr is access all Node_Item'Class;

    -- abstraction assumes head of a list is a Node_Ptr

    function Size_Of (The_List : in RTI.Node_Ptr) return Natural;

    procedure Empty (The_List : in out RTI.Node_Ptr);

    procedure Add (The_List : in out RTI.Node_Ptr;
                   The_Item : in RTI.Node_Item'Class);

    procedure Remove (The_List : in out RTI.Node_Ptr;
                      The_Item : in RTI.Node_Item'Class);

    function Is_Empty (The_List : in RTI.Node_Ptr) return Boolean;

    function Get (The_List : in RTI.Node_Ptr;
                  Ith_Item: Positive) return RTI.Node_Item'Class;


generic
        with function Equal (Item_A: in RTI.Node_Item'Class;
                             Item_B: in RTI.Node_Item'Class) return Boolean;
    function Is_Member (The_List : in RTI.Node_Item'Class;
                        The_Item : in RTI.Node_Item'Class) return Boolean;



end RTI_Basic_Lists;
```

```
--
-- File: rti_ambassador_class.1.ada
--

with Federate_Ambassador_Class;

--
-- RTI_Ambassador_Class represents the class of objects (and services
-- associated with them) which provide the interface to the RTI.  A new RTI
-- Ambassador Object becomes associated with each new "join" by a simulation
-- to the Federation Execution
--

package RTI_Ambassador_Class is

  -- The type from which RTI Ambassadors will be instantiated
  -- This tagged type contains a reference to the associated Federate
  -- Ambassador object.  This is needed due to the way Ada95 associates
  -- routines with objects.  This differs from the C++ paradigm in which
  -- routines are tightly bound to the object definition

  type RTI_Ambassador is tagged
    record
      This_Federate_Ambassador :
        Federate_Ambassador_Class.Federate_Ambassador_Ptr;
    end record;

  type RTI_Ambassador_Ptr is access RTI_Ambassador;

end RTI_Ambassador_Class;
```

```
--
-- File: federate_ambassador_class.1.ada
--

-------------------------------------------------------------------------------
--|NAME federate_ambassador_class.1.ada
--|PURPOSE
--|   Provide an abstract class definition for the Federate_Ambassador Object.
--|   Federates need to "inherit" from the template and define the member
--|   functions listed here.  Instantiations of this class will be called
--|   by the RTI
--|
--|ENVIRONMENT_DEPENDENCIES None
--|MACHINE_DEPENDENCIES designed for use on IBM PowerPC running AIX
--|DEVICE_IO_DEPENDENCIES None
--|EXECUTION_CONSTRAINTS None
--|DESIGN_CONSTRAINTS designed to function with version F0 of the RTI
--|
------------------------------------------------------------------------------

--
-- Federate_Ambassador_Class provides a "template" for federate supplied
-- routines.  Each of the federate supplied routines are initiated by the
-- RTI via callbacks
--

with RTI;
with DDM;

package Federate_Ambassador_Class is

   -- Federate_Ambassador is an abstract tagged type because it is up to the
   -- the individual Federates to instantiate thisobject along with its
   -- corresponding routines.
   type Federate_Ambassador is abstract tagged
      record
         Identifier : Integer;
      end record;

   --
   -- Federate_Ambassador_Ptr provides access to the class wide set of
   -- Federate Ambassadors.  This allows dispatching to the appropriate
   -- routine as additional Federate Ambassadors are created.
   type Federate_Ambassador_Ptr is access Federate_Ambassador'Class;


--
--
-- Provides the Federation Management services (2.x services) initiated
-- by the RTI.
--

-- 2.6
    procedure Initiate_Pause
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Label : in RTI.Pause_Label) is abstract;
    -- should raise only the following exceptions:
    --    Federate_Already_Paused
    --    Federate_Internal_Error

-- 2.9
```

```
    procedure Initiate_Resume
      (The_Federate_Ambassador : in Federate_Ambassador) is abstract;
    -- should raise only the following exceptions:
    --    Federate_Not_Paused
    --    Federate_Internal_Error

-- 2.12
    procedure Initiate_Federate_Save
         (The_Federate_Ambassador : in Federate_Ambassador;
           The_Label : in RTI.Save_Label) is abstract;
    -- should raise only the following exceptions:
    --    Unable_To_Perform_Save
    --    Federate_Internal_Error

    procedure Initiate_Federate_Save
         (The_Federate_Ambassador : in Federate_Ambassador;
           The_Label : in RTI.Save_Label;
           The_Time : in RTI.Federation_Time) is abstract;
    -- should raise only the following exceptions:
    --    Invalid_Federation_Time
    --    Unable_To_Perform_Save
    --    Federate_Internal_Error

-- 2.16
    procedure Initiate_Restore
         (The_Federate_Ambassador : in Federate_Ambassador;
           The_Label : in RTI.Save_Label) is abstract;
    -- should raise only the following exceptions:
    --    Specified_Save_Label_Does_Not_Exist
    --    Could_Not_Restore
    --    Federate_Internal_Error


--
--
-- Provides the Declaration Management services (3.x services) initiated
-- by the RTI.
--

-- 3.5
    procedure Start_Updates
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Class : in RTI.Object_Class_Handle;
       The_Attributes : access RTI.Attribute_Handle_Set) is abstract;
    -- should raise only the following exceptions:
    --    Object_Class_Not_Published
    --    Attribute_Not_Published
    --    Federate_Internal_Error

    procedure Stop_Updates
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Class : in RTI.Object_Class_Handle;
       The_Attributes : access RTI.Attribute_Handle_Set) is abstract;
    -- should raise only the following exceptions:
    --    Object_Class_Not_Published
    --    Attribute_Not_Published
    --    Federate_Internal_Error

-- 3.6
    procedure Start_Interaction_Generation
      (The_Federate_Ambassador : in Federate_Ambassador;
```

```
    The_Handle : RTI.Interaction_Class_Handle) is abstract;
 -- should raise only the following exceptions:
 --    Interaction_Class_Not_Published
 --    Federate_Internal_Error

 procedure Stop_Interaction_Generation
   (The_Federate_Ambassador : in Federate_Ambassador;
    The_Handle : in RTI.Interaction_Class_Handle) is abstract;
 -- should raise only the following exceptions:
 --    Interaction_Class_Not_Published
 --    Federate_Internal_Error


--
--
-- Provides the Object Management services (4.x services) initiated
-- by the RTI.
--

-- types unique to Internal Object Management services

type Object_Removal_Reason is (Out_Of_Region,
                               Object_Deleted,
                               No_Longer_Subscribed);

for Object_Removal_Reason use
   (Out_Of_Region => 0,
    Object_Deleted => 1,
    No_Longer_Subscribed => 2);


-- 4.4
procedure Discover_Object
  (The_Federate_Ambassador : in Federate_Ambassador;
   The_Object : in RTI.Object_Id;
   The_Object_Class : in RTI.Object_Class_Handle;
   The_Time :in  RTI.Federation_Time;
   The_Tag : in RTI.User_Supplied_Tag;
   The_Event : in RTI.Event_Retraction_Handle) is abstract;
-- should raise only the following exceptions:
--    Could_Not_Discover_Object
--    Object_Class_Not_Known
--    Invalid_Federation_Time
--    Federate_Internal_Error

-- 4.5
procedure Reflect_Attribute_Values
  (The_Federate_Ambassador : in Federate_Ambassador;
   The_Object : in RTI.Object_Id;
   The_Attributes : access RTI.Attribute_Handle_Value_Pair_Set;
   The_Time : in RTI.Federation_Time;
   The_Tag : in RTI.User_Supplied_Tag;
   The_Event : in RTI.Event_Retraction_Handle) is abstract;
-- should raise only the following exceptions:
--    Object_Not_Known
--    Attribute_Not_Known
--    Invalid_Federation_Time
--    Federate_Internal_Error

-- 4.7
procedure Receive_Interaction
```

```
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Interaction_Class : in RTI.Interaction_Class_Handle;
       The_Parameters : access RTI.Parameter_Handle_Value_Pair_Set;
       The_Time : in RTI.Federation_Time;
       The_Tag : in RTI.User_Supplied_Tag;
       The_Event : in RTI.Event_Retraction_Handle) is abstract;
   -- should raise only the following exceptions:
   --    Interaction_Class_Not_Known
   --    Interaction_Parameter_Not_Known
   --    Invalid_Federation_Time
   --    Federate_Internal_Error



   -- 4.9
   procedure Remove_Object
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Object : in RTI.Object_Id;
       The_Reason : in Object_Removal_Reason;
       The_Time : in RTI.Federation_Time;
       The_Tag : in RTI.User_Supplied_Tag;
       The_Event : in RTI.Event_Retraction_Handle) is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Invalid_Federation_Time
   --    Federate_Internal_Error

   procedure Remove_Object
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Object : in RTI.Object_Id;
       The_Reason : in Object_Removal_Reason;
       The_Time : in RTI.Federation_Time;
       The_Tag : in RTI.User_Supplied_Tag) is abstract;
  -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Invalid_Federation_Time
   --    Federate_Internal_Error


   -- 4.15
   procedure Provide_Attribute_Value_Update
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Object : in RTI.Object_Id;
       The_Attributes : access RTI.Attribute_Handle_Set) is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Known
   --    Federate_Internal_Error

   -- 4.17
   procedure Reflect_Retraction
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Event : in RTI.Event_Retraction_Handle) is abstract;
   -- should raise only the following exceptions:
   --    Event_Not_Known
   --    Federate_Internal_Error


--
--
-- Provides the Ownership Management services (5.x services) initiated
```

```
-- by the RTI.
--


-- 5.2
   function Request_Attribute_Ownership_Assumption
     (The_Federate_Ambassador : in Federate_Ambassador;
      The_Object : in RTI.Object_Id;
      Offered_Attributes : access RTI.Attribute_Handle_Set;
      The_Tag : in RTI.User_Supplied_Tag)
    return RTI.Attribute_Handle_Set is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Known
   --    Attribute_Already_Owned
   --    Federate_Internal_Error

   function Request_Attribute_Ownership_Assumption
     (The_Federate_Ambassador : in Federate_Ambassador;
      The_Object : in RTI.Object_Id;
      Offered_Attributes : access RTI.Attribute_Handle_Set)
    return RTI.Attribute_Handle_Set is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Known
   --    Federate_Internal_Error

   -- 5.3
   procedure Attribute_Ownership_Divestiture_Notification
     (The_Federate_Ambassador : in Federate_Ambassador;
      The_Object : in RTI.Object_Id;
      Released_Attributes : access RTI.Attribute_Handle_Set) is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Known
   --    Federate_Internal_Error

   -- 5.4
   procedure Attribute_Ownership_Acquisition_Notification
     (The_Federate_Ambassador : in Federate_Ambassador;
      The_Object : in RTI.Object_Id;
      Acquired_Attributes : access RTI.Attribute_Handle_Set) is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Known
   --    Federate_Internal_Error

   -- 5.6
   function Request_Attribute_Ownership_Release
     (The_Federate_Ambassador : in Federate_Ambassador;
      The_Object : in RTI.Object_Id;
      Requested_Attributes : access RTI.Attribute_Handle_Set;
      The_Tag : in RTI.User_Supplied_Tag)
    return RTI.Attribute_Handle_Set is abstract;
   -- should raise only the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Known
   --    Federate_Internal_Error


   -- 5.7
```

```
   procedure Inform_Attribute_Ownership
     (The_Federate_Ambassador : in Federate_Ambassador;
      The_Object: in RTI.Object_Id;
      The_Attribute: in RTI.Attribute_Handle;
      The_Owner: in RTI.Federate_Handle) is abstract;
   -- should raise only the following exceptions:
   --    Federate_Internal_Error

--
--
-- Provides the Time Management services (6.x services) initiated
-- by the RTI.
--


-- 6.10
    procedure Time_Advance_Grant
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Time : in RTI.Federation_Time) is abstract;
    -- should raise only the following exceptions:
    --    Invalid_Federation_Time
    --    Time_Advance_Was_Not_In_Progress
    --    Federation_Time_Already_Passed
    --    Federate_Internal_Error


--
--
-- Provides the Data Distribution Management services (7.x services) initiated
-- by the RTI.
--

-- 7.4
    procedure Change_Thresholds
      (The_Federate_Ambassador : in Federate_Ambassador;
       The_Region : in DDM.Region;
       The_Thresholds : access DDM.Threshold_Set) is abstract;
    -- should raise only the following exceptions:
    --    Region_Not_Known
    --    Federate_Internal_Error



end Federate_Ambassador_Class;
```

```
--
-- File rac_federation_management.1.ada
--

with RTI;
with RTI_Base_Types;

package RTI_Ambassador_Class.Federation_Management is

--
-- Provides the Federation Management services (2.x services) initiated
-- by federates.
--

--
-- types unique to Federation Management services
--

   type File_Name is new RTI_Base_Types.RTI_String;

   type Resign_Action is (Release_Attributes,
                          Delete_Objects,
                          Delete_Objects_And_Release_Attributes);

   for Resign_Action use
          (Release_Attributes => 1,
          Delete_Objects => 2,
          Delete_Objects_And_Release_Attributes => 3);

   type Save_Indicator is (Save_Achieved,
                           Save_Not_Achieved);

   type Restore_Indicator is (Restore_Federate_Achieved,
                              Restore_Federate_Not_Achieved);


   --
   -- 2.x services
   --

   -- 2.1
   procedure Create_Federation_Execution
     (The_Name : in RTI.Federation_Execution_Name;
      R_I_D : in File_Name);
   -- should raise ONLY the following exceptions:
   --    Federation_Execution_Already_Exists
   --    Could_Not_Open_RID
   --    Error_Reading_RID
   --    RTI_Internal_Error


   -- 2.2
   procedure Destroy_Federation_Execution
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
      Execution_Name : in RTI.Federation_Execution_Name);
   -- should raise ONLY the following exceptions:
   --    Federates_Currently_Joined
   --    Federation_Execution_Does_Not_Exist
   --    RTI_Internal_Error
```

```
-- 2.3
procedure Join_Federation_Execution
  (Your_Name : in RTI.Federate_Name;
   Execution_Name : in RTI.Federation_Execution_Name;
   The_RTI_Ambassador : in RTI_Ambassador_Ptr);
-- should raise ONLY the following exceptions:
--    Federate_Already_Execution_Member
--    Federation_Execution_Does_Not_Exist
--    Could_Not_Open_RID
--    Error_Reading_RID
--    RTI_Internal_Error


-- 2.4
procedure Resign_Federation_Execution
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Action : in Resign_Action);
-- should raise ONLY the following exceptions:
--    Federate_Owns_Attributes
--    Federate_Not_Execution_Member
--    Federation_Execution_Does_Not_Exist
--    Invalid_Resign_Action
--    RTI_Internal_Error

                   --  not implemented in F.0
procedure Resign_Federation_Execution
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   A_Federate : in RTI.Federate_Handle;
   The_Action : in Resign_Action);
-- should raise ONLY the following exceptions:
--    Federate_Owns_Attributes
--    Federation_Execution_Does_Not_Exist
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 2.5
                   --  not implemented in F.0
procedure Request_Pause
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Label : in RTI.Pause_Label);
-- should raise ONLY the following exceptions:
--    Federation_Already_Paused
--    Federate_Not_Execution_Member
--    RTI_Internal_Error



-- 2.7
                   --  not implemented in F.0
procedure Pause_Achieved
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Label : in RTI.Pause_Label);
-- should raise ONLY the following exceptions:
--    Unknown_Label
--    No_Pause_Requested
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 2.8
```

```
                    --  not implemented in F.0
procedure Request_Resume
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr);
-- should raise ONLY the following exceptions:
--    Federation_Not_Paused
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 2.10
                    --  not implemented in F.0
procedure Resume_Achieved
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr);
-- should raise ONLY the following exceptions:
--    No_Resume_Requested
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 2.11
                    --  not implemented in F.0
procedure Request_Federation_Save
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Label : in RTI.Save_Label);
-- should raise ONLY the following exceptions:
--    Save_In_Progress
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


                    --  not implemented in F.0
procedure Request_Federation_Save
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Label : in RTI.Save_Label;
   The_Time : in RTI.Federation_Time);
-- should raise ONLY the following exceptions:
--    Save_In_Progress
--    Federation_Time_Already_Passed
--    Invalid_Federation_Time
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 2.13
                    --  not implemented in F.0
procedure Federate_Save_Begun
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr);
-- should raise ONLY the following exceptions:
--    Save_Not_Initiated
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


                    --  not implemented in F.0
procedure Federate_Save_Begun
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Time : in RTI.Federation_Time);
-- should raise ONLY the following exceptions:
--    Save_Not_Initiated
--    Invalid_Federate_Time
--    Federate_Not_Execution_Member
```

```
   --    RTI_Internal_Error


   -- 2.14
                      --  not implemented in F.0
   procedure Federate_Save_Achieved
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr);
   -- should raise ONLY the following exceptions:
   --    Save_Not_Initiated
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error

                      --  not implemented in F.0
   procedure Federate_Save_Not_Achieved
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr);
   -- should raise ONLY the following exceptions:
   --    Save_Not_Initiated
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 2.15
                      --  not implemented in F.0
   procedure Request_Restore
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
      The_Label : in RTI.Save_Label);
   -- should raise ONLY the following exceptions:
   --    Specified_Save_Label_Does_Not_Exist
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error

   -- 2.17
                      --  not implemented in F.0
   procedure Restore_Achieved
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
      The_Label : in RTI.Save_Label);
   -- should raise ONLY the following exceptions:
   --    Unknown_Label
   --    Restore_Not_Requested
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error

                      --  not implemented in F.0
   procedure Restore_Not_Achieved
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
      The_Label : in RTI.Save_Label);
   -- should raise ONLY the following exceptions:
   --    Unknown_Label
   --    Restore_Not_Requested
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


end RTI_Ambassador_Class.Federation_Management;
```

```
--
-- File rac_declaration_management.1.ada
--

with RTI;
with DDM;

package RTI_Ambassador_Class.Declaration_Management is


--
-- Provides the Declaration Management services (3.x services) initiated
-- by federates.
--


-- 3.1
    procedure Publish_Object_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Object_Class_Handle;
                Attribute_List : access RTI.Attribute_Handle_Set);
    -- should raise ONLY the following exceptions:
    --    Object_Class_Not_Defined
    --    Attribute_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


    procedure Unpublish_Object_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Object_Class_Handle);
    -- should raise ONLY the following exceptions:
    --    Object_Class_Not_Defined
    --    Federate_Owns_Attributes
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


-- 3.2
    procedure Publish_Interaction_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Interaction : in RTI.Interaction_Class_Handle);
    -- should raise ONLY the following exceptions:
    --    Interaction_Class_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


    procedure Unpublish_Interaction_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Interaction : in RTI.Interaction_Class_Handle);
    -- should raise ONLY the following exceptions:
    --    Interaction_Class_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


-- 3.3
    procedure Subscribe_Object_Class_Attribute
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Object_Class_Handle;
```

```
                Attribute_List : access RTI.Attribute_Handle_Set);
    -- should raise ONLY the following exceptions:
    --    Object_Class_Not_Defined,
    --    Attribute_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


                      --  not implemented in F.0
    procedure Subscribe_Object_Class_Attribute
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Object_Class_Handle;
                The_Attribute : in RTI.Attribute_Handle;
                The_Region : in DDM.Region);
    -- should raise ONLY the following exceptions:
    --    Object_Class_Not_Defined
    --    Attribute_Not_Defined
    --    Region_Not_Known
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


    procedure Unsubscribe_Object_Class_Attribute
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Object_Class_Handle);
    -- should raise ONLY the following exceptions:
    --    Object_Class_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


                      --  not implemented in F.0
    procedure Unsubscribe_Object_Class_Attribute
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Object_Class_Handle;
                The_Region : in DDM.Region);
    -- should raise ONLY the following exceptions:
    --    Object_Class_Not_Defined
    --    Region_Not_Known
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


-- 3.4
    procedure Subscribe_Interaction_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Interaction_Class_Handle);
    -- should raise ONLY the following exceptions:
    --    Interaction_Class_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


                      --  not implemented in F.0
    procedure Subscribe_Interaction_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Interaction_Class_Handle;
                The_Region : in DDM.Region);
    -- should raise ONLY the following exceptions:
    --    Interaction_Class_Not_Defined
    --    Region_Not_Known
```

```
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


    procedure Unsubscribe_Interaction_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Interaction_Class_Handle);
    -- should raise ONLY the following exceptions:
    --    Interaction_Class_Not_Defined
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


                      --  not implemented in F.0
    procedure Unsubscribe_Interaction_Class
                 (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
                The_Class : in RTI.Interaction_Class_Handle;
                The_Region : in DDM.Region);
    -- should raise ONLY the following exceptions:
    --    Interaction_Class_Not_Defined
    --    Region_Not_Known
    --    Federate_Not_Execution_Member
    --    RTI_Internal_Error


end RTI_Ambassador_Class.Declaration_Management;
```

```
--
-- File rac_object_management.1.ada
--

with Rti;
with Rti_Basic_Lists;

package Rti_Ambassador_Class.Object_Management is

   --
   -- provides the object management services (2.x services) initiated
   -- by federates.
   --

   --
   -- types unique to object management services
   --

   subtype Object_Id_Count is Positive;

   type Data_Order_Type is (Receive,
                            Timestamp);
   for Data_Order_Type use
      (Receive => 1,
       Timestamp => 2);

   type Transportation_Type is (Reliable,
                                Best_Effort);
   for Transportation_Type use
      (Reliable => 1,
       Best_Effort => 2);


   -- 4.1
   procedure Request_Id
     (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
      Id_Count : in Object_Id_Count;
      First_Id: Rti.Object_Id;
      Last_Id: Rti.Object_Id);
   -- should raise only the following exceptions:
   --    too_many_ids_requested
   --    id_supply_exhausted
   --    federate_not_execution_member
   --    rti_internal_error


   -- 4.2
   procedure Register_Object
     (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
      The_Object_Class : in Rti.Object_Class_Handle;
      The_Object : in Rti.Object_Id);
   -- should raise only the following exceptions:
   --    invalid_object_id
   --    object_id_already_registered
   --    object_class_not_defined
   --    object_class_not_published
   --    federate_not_execution_member
   --    rti_internal_error

   -- 4.3
   function Update_Attribute_Values
```

```
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Object : in Rti.Object_Id;
   The_Attributes_And_Values : access Rti.Attribute_Handle_Value_Pair_Set;
   The_Time : in Rti.Federation_Time;
   The_Tag : in Rti.User_Supplied_Tag)
 return Rti.Event_Retraction_Handle;
-- should raise only the following exceptions:
--    object_not_know
--    attribute_not_defined
--    attribute_not_owned
--    invalid_federation_time
--    federate_not_execution_member
--    rti_internal_error


-- 4.6
function Send_Interaction
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Interaction : in Rti.Interaction_Class_Handle;
   The_Parameters_And_Values : access Rti.Parameter_Handle_Value_Pair_Set;
   The_Time : in Rti.Federation_Time;
   The_Tag : in Rti.User_Supplied_Tag)
return Rti.Event_Retraction_Handle;
-- should raise only the following exceptions:
--    interaction_class_not_defined
--    interaction_class_not_published
--    interaction_parameter_not_defined
--    invalid_federation_time
--    federate_not_execution_member
--    rti_internal_error

-- 4.8
function Delete_Object
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Object : in Rti.Object_Id;
   The_Time : in Rti.Federation_Time;
   The_Tag : in Rti.User_Supplied_Tag)
return Rti.Event_Retraction_Handle;
-- should raise only the following exceptions:
--    object_not_known
--    delete_privilege_not_held
--    invalid_federation_time
--    federate_not_execution_member
--    rti_internal_error


-- 4.10
procedure Change_Attribute_Transportation_Type
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Object : in Rti.Object_Id;
   The_Attributes : access Rti.Attribute_Handle_Set;
   The_Type : in Transportation_Type);
-- should raise only the following exceptions:
--    object_not_known
--    attribute_not_defined
--    attribute_not_owned
--    invalid_transportation_type
--    federate_not_execution_member
--    rti_internal_error
```

```
-- 4.11
procedure Change_Attribute_Order_Type
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Object : in Rti.Object_Id;
   The_Attributes : access Rti.Attribute_Handle_Set;
   The_Data_Order : in Data_Order_Type);
-- should raise only the following exceptions:
--    object_not_known
--    attribute_not_defined
--    attribute_not_owned
--    invalid_order_type
--    federate_not_execution_member
--    rti_internal_error

-- 4.12
procedure Change_Interaction_Transportation_Type
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Interaction_Class : in Rti.Interaction_Class_Handle;
   The_Type : in Transportation_Type);
-- should raise only the following exceptions:
--    interaction_class_not_defined
--    federate_not_publishing_interaction_class
--    invalid_transportation_type
--    federate_not_execution_member
--    rti_internal_error


-- 4.13
procedure Change_Interaction_Order_Type
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Interaction_Class : in Rti.Interaction_Class_Handle;
   The_Data_Order : in Data_Order_Type);
-- should raise only the following exceptions:
--    interaction_class_not_defined
--    federate_not_publishing_interaction_class
--    invalid_order_type
--    federate_not_execution_member
--    rti_internal_error

-- 4.14
procedure Request_Object_Attribute_Value_Update
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Object : in Rti.Object_Id;
   The_Attributes : access Rti.Attribute_Handle_Set);
-- should raise only the following exceptions:
--    object_not_known
--    attribute_not_defined
--    federate_not_execution_member
--    rti_internal_error

procedure Request_Object_Attribute_Value_Update
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Object : in Rti.Object_Id);
-- should raise only the following exceptions:
--    object_not_known
--    federate_not_execution_member
--    rti_internal_error

procedure Request_Class_Attribute_Value_Update
  (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
   The_Class : in Rti.Object_Class_Handle;
```

```
      The_Attributes : access Rti.Attribute_Handle_Set);
   -- should raise only the following exceptions:
   --    object_class_not_defined
   --    attribute_not_defined
   --    federate_not_execution_member
   --    rti_internal_error

   procedure Request_Class_Attribute_Value_Update
     (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
      The_Class : in Rti.Object_Class_Handle);
   -- should raise only the following exceptions:
   --    object_class_not_defined
   --    federate_not_execution_member
   --    rti_internal_error


   -- 4.16
   procedure Retract
     (The_Rti_Ambassador : in Rti_Ambassador_Ptr;
      The_Event : in Rti.Event_Retraction_Handle);
   -- should raise only the following exceptions:
   --    invalid_retraction_handle
   --    federate_not_execution_member
   --    rti_internal_error


end Rti_Ambassador_Class.Object_Management;
```

```
--
-- File rac_ownership_management.1.ada
--

with RTI_Base_Types;
-- with RTI_Basic_Lists;
with RTI;

package RTI_Ambassador_Class.Ownership_Management is

--
-- Provides the Ownership Management services (5.x services) initiated
-- by federates.
--

--
-- types unique to Ownership Management services
--

   type Ownership_Divestiture_Condition is (Negotiated,
                                            Unconditional);

   for Ownership_Divestiture_Condition use
      (Negotiated => 1,
       Unconditional => 2);

   type Federate_Set_Element is new RTI.Node_Item with record
       The_Federate: RTI.Federate_Handle;
   end record;
   type Federate_Handle_Set is new RTI.Node_Ptr;


   -- 5.1
   procedure Request_Attribute_Ownership_Divestiture
     (The_RTI_Ambassador :in RTI_Ambassador_Ptr;
      The_Object : in RTI.Object_Id;
      The_Attributes : access RTI.Attribute_Handle_Set;
      The_Condition : in Ownership_Divestiture_Condition;
      The_Tag : in RTI.User_Supplied_Tag);
   -- should raise ONLY the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Defined
   --    Attribute_Not_Owned
   --    Invalid_Divestiture_Condition
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   procedure Request_Attribute_Ownership_Divestiture
     (The_RTI_Ambassador :in RTI_Ambassador_Ptr;
      The_Object : in RTI.Object_Id;
      The_Attributes : access RTI.Attribute_Handle_Set;
      The_Condition : in Ownership_Divestiture_Condition;
      The_Tag : in RTI.User_Supplied_Tag;
      The_Candidates : access Federate_Handle_Set);
   -- should raise ONLY the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Defined
   --    Attribute_Not_Owned
   --    Invalid_Divestiture_Condition
   --    Federate_Does_Not_Exist
```

```
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 5.5
   procedure Request_Attribute_Ownership_Acquisition
     (The_RTI_Ambassador :in RTI_Ambassador_Ptr;
      The_Object : in RTI.Object_Id;
      Desired_Attributes : access RTI.Attribute_Handle_Set;
      The_Tag : in RTI.User_Supplied_Tag);
   -- should raise ONLY the following exceptions:
   --    Object_Not_Known
   --    Object_Class_Not_Published
   --    Object_Class_Not_Subscribed
   --    Attribute_Not_Defined
   --    Attribute_Not_Published
   --    Attribute_Not_Subscribed
   --    Federate_Owns_Attributes
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 5.7
   procedure Query_Attribute_Ownership
     (The_RTI_Ambassador :in RTI_Ambassador_Ptr;
      The_Object : in RTI.Object_Id;
      The_Attribute : in RTI.Attribute_Handle);
   -- should raise ONLY the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Defined
   --    Attribute_Not_Known
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   function Attribute_Is_Owned_By_Federate
     (The_RTI_Ambassador :in RTI_Ambassador_Ptr;
      The_Object : in RTI.Object_Id;
      The_Attribute : in RTI.Attribute_Handle)
   return RTI.RTI_Boolean;
   -- should raise ONLY the following exceptions:
   --    Object_Not_Known
   --    Attribute_Not_Defined
   --    Attribute_Not_Known
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error

end RTI_Ambassador_Class.Ownership_Management;
```

```
--
-- File rac_time_management.1.ada
--

with RTI;

package RTI_Ambassador_Class.Time_Management is

--
-- Provides the Time Management services (6.x services) initiated
-- by federates.
--

--
-- types unique to Time Management services
--

   subtype Federation_Time_Delta is RTI.Federation_Time;


   -- 6.1
   function Request_Federation_Time
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr)
   return RTI.Federation_Time;
   -- should raise ONLY the following exceptions:
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 6.2
   function Request_Lower_Bound_Timestamp
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr)
   return RTI.Federation_Time;
   -- should raise ONLY the following exceptions:
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 6.3
   function Request_Federate_Time
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr)
   return RTI.Federation_Time;
   -- should raise ONLY the following exceptions:
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 6.4
   function Request_Minimum_Next_Event_Time
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr)
   return RTI.Federation_Time;
   -- should raise ONLY the following exceptions:
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error;


   -- 6.5
   procedure Set_Lookahead
     (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
      The_Lookahead : in Federation_Time_Delta);
   -- should raise ONLY the following exceptions:
```

```
--    Invalid_Federation_Time_Delta
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 6.6
function Request_Lookahead
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr)
return Federation_Time_Delta;
-- should raise ONLY the following exceptions:
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 6.7
procedure Time_Advance_Request
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Time : in RTI.Federation_Time);
-- should raise ONLY the following exceptions:
--    Time_Advance_Already_In_Progress
--    Federation_Time_Already_Passed
--    Invalid_Federation_Time
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 6.8
procedure Next_Event_Request
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Time : in RTI.Federation_Time);
-- should raise ONLY the following exceptions:
--    Time_Advance_Already_In_Progress
--    Federation_Time_Already_Passed
--    Invalid_Federation_Time
--    Federate_Not_Execution_Member
--    RTI_Internal_Error

                    -- not implemented in F.0
procedure Next_Event_Request
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Time : in RTI.Federation_Time;
   Delta_Time : in RTI.Federation_Time);
-- should raise ONLY the following exceptions:
--    Time_Advance_Already_In_Progress
--    Federation_Time_Already_Passed
--    Invalid_Federation_Time
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 6.9
procedure Flush_Queue_Request
  (The_RTI_Ambassador : in RTI_Ambassador_Ptr;
   The_Time : in RTI.Federation_Time);
-- should raise ONLY the following exceptions:
--    Time_Advance_Already_In_Progress
--    Federation_Time_Already_Passed
--    Invalid_Federation_Time
--    Federate_Not_Execution_Member
--    RTI_Internal_Error
```

```
end RTI_Ambassador_Class.Time_Management;
```

```
--
-- File: ddm.1.ada
--

with RTI_Base_Types;
-- with RTI_Basic_Lists;
with RTI;

package DDM is

   type Space_Name is new RTI_BASE_TYPES.RTI_String;
   type Space_Handle is private;

   type Region is private;

   type RTI_Range is record
       first, last: RTI_Base_Types.RTI_ULong;
   end record;

   type Extents_Element is new RTI.Node_Item with record
       The_Range: RTI_Range;
   end record;
   type Extents is new RTI.Node_Ptr;

   type Extent_Set_Element is new RTI.Node_Item with record
       The_Extent: Extents;
   end record;
   type Extent_Set is new RTI.Node_Ptr;

   type Threshold is new RTI_Base_Types.RTI_ULong;

   type Threshold_Set_Element is new RTI.Node_Item with record
       The_Threshold: Threshold;
   end record;
   type Threshold_Set is new RTI.Node_Ptr;

   type Association_Action is (Form_Association,
                               Break_Association);

   procedure Set_Region
         (My_Region : in out Region;
         With_Value : in RTI_Base_Types.RTI_Ulong);

private

   type Region is new RTI_Base_Types.RTI_ULong;

   type Space_Handle is new RTI.Simple_Handle;

end DDM;
```

```
--
--  File rac_data_distribution_management.1.ada
--

with RTI_BASE_TYPES;
with RTI;
with DDM;

package RTI_Ambassador_Class.Data_Distribution_Management is

   --
   -- Provides the Data Distribution Management services
   -- (7.x services) initiated by federates.
   --


   -- 7.1
   procedure Create_Update_Region
     (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
      The_Routing_Space : in DDM.Space_Handle;
      The_Extents : access DDM.Extent_Set;
      The_Region : out DDM.Region);

   -- should raise ONLY the following exceptions:
   --    Space_Not_Defined
   --    Invalid_Extents
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error

   -- 7.2
   procedure Create_Subscription_Region
     (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
      The_Routing_Space : in DDM.Space_Name;
      The_Extents : access DDM.Extent_Set;
      The_Region : out DDM.Region);
   -- should raise ONLY the following exceptions:
   --    Space_Not_Defined
   --    Invalid_Extents
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 7.3
   procedure Associate_Update_Region
     (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
      The_Region : in DDM.Region;
      The_Object : in RTI.Object_Id;
      The_Attributes : access RTI.Attribute_Handle_Set;
      The_Action : in DDM.Association_Action);
   -- should raise ONLY the following exceptions:
   --    Region_Not_Known
   --    Object_Not_Known
   --    Attribute_Not_Defined
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   procedure Associate_Update_Region
     (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
      The_Region : in DDM.Region;
      The_Interaction_Class : in RTI.Interaction_Class_Handle;
```

```
      The_Action : in DDM.Association_Action);
   -- should raise ONLY the following exceptions:
   --    Region_Not_Known
   --    Interaction_Class_Not_Known
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 7.5
   procedure Modify_Region
     (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
      The_Region : in DDM.Region;
      The_Extents : access DDM.Extent_Set);
   -- should raise ONLY the following exceptions:
   --    Region_Not_Known
   --    Invalid_Extents
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


   -- 7.6
   procedure Delete_Region
     (The_RTI_Ambassador: in RTI_Ambassador_Ptr;
      The_Region : in DDM.Region);
   -- should raise ONLY the following exceptions:
   --    Region_Not_Known
   --    Federate_Not_Execution_Member
   --    RTI_Internal_Error


--
-- 8.9
    function Get_Space_Handle
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Name:
           in DDM.Space_Name)
   return DDM.Space_Handle;
-- should raise only the following exceptions:
--    Name_Not_Found
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error
--    Unimplemented_Service


-- 8.10
    function Get_Space_Name
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Handle: DDM.Space_Handle)
    return DDM.Space_Name;
-- should raise only the following exceptions:
--    Space_Not_Defined
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error
--    Unimplemented_Service

end RTI_Ambassador_Class.Data_Distribution_Management;
```

```
--
-- File: rti_support.1.ada
--

with RTI;
with RTI_Base_Types;
with RTI_Ambassador_Class.Data_Distribution_Management;

package RTI_Support is

-- 8.1
    function Get_Object_Class_Handle
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Name: in RTI.Object_Class_Name)
    return RTI.Object_Class_Handle;
-- should raise only the following exceptions:
--    Name_Not_Found
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error


-- 8.2
    function Get_Object_Class_Name
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Handle: in RTI.Object_Class_Handle)
    return RTI.Object_Class_Name;
-- should raise only the following exceptions:
--    Object_Class_Not_Defined
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error


-- 8.3
    function Get_Attribute_Handle
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Name: in RTI.Attribute_Name;
       Which_Class: in RTI.Object_Class_Handle)
    return RTI.Attribute_Handle;
-- should raise only the following exceptions:
--    Object_Class_Not_Defined
--    Name_Not_Found
--    Federate_Not_ExecutionMember
--    Concurrent_Access_Attempted
--    RTI_Internal_Error


-- 8.4
    function Get_Attribute_Name
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Handle: in RTI.Attribute_Handle;
       Which_Class: in RTI.Object_Class_Handle)
    return RTI.Attribute_Name;
-- should raise only the following exceptions:
--    Object_Class_Not_Defined
--    Attribute_Not_Defined
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error;
```

```
-- 8.5
    function Get_Interaction_Class_Handle
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Name: in RTI.Interaction_Class_Name)
    return RTI.Interaction_Class_Handle;
-- should raise only the following exceptions:
--    Name_Not_Found
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error


-- 8.6
    function Get_Interaction_Class_Name
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Handle: in RTI.Interaction_Class_Handle)
    return RTI.Interaction_Class_Name;
-- should raise only the following exceptions:
--    Interaction_Class_Not_Defined
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error


-- 8.7
    function Get_Parameter_Handle
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Name: in RTI.Parameter_Name;
       Which_Class: in RTI.Interaction_Class_Handle)
    return RTI.Parameter_Handle;
-- should raise only the following exceptions:
--    Interaction_Class_Not_Defined
--    Name_Not_Found
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error


-- 8.8
    function Get_Parameter_Name
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_Handle: in RTI.Parameter_Handle;
       Which_Class: in RTI.Interaction_Class_Handle)
    return RTI.Parameter_Name;
-- should raise only the following exceptions:
--    Interaction_Class_Not_Defined
--    Interaction_Parameter_Not_Defined
--    Federate_Not_Execution_Member
--    Concurrent_Access_Attempted
--    RTI_Internal_Error



-- 8.13
    procedure Turn_Regulation_On
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr);
-- should raise only the following exceptions:
--    Federation_Time_Already_Passed
--    Federate_Not_Execution_Member
--    RTI_Internal_Error
```

```
    function Turn_Regulation_On_Now
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr)
    return RTI.Federation_Time;
-- should raise only the following exceptions:
--    Federate_Not_Execution_Member
--    RTI_Internal_Error

    procedure Turn_Regulation_Off
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr);
-- should raise only the following exceptions:
--    Federate_Not_Execution_Member
--    RTI_Internal_Error


-- 8.14
    procedure Set_Time_Constrained
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       The_State: RTI.RTI_Boolean);
-- should raise only the following exceptions:
--    Federate_Not_Execution_Member
--    RTI_Internal_Error;


-- 8.15
    function Tick
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr)
    return RTI.RTI_Boolean;
-- should raise only the following exceptions:
--    RTI_Internal_Error;

    function Tick
      (The_RTI_Ambassador: in RTI_Ambassador_Class.RTI_Ambassador_Ptr;
       Minimum : in RTI.Tick_Time;
       Maximum : in RTI.Tick_Time)
    return RTI.RTI_Boolean;
-- should raise only the following exceptions:
--    RTI_Internal_Error;


end RTI_Support;
```

11 References

[HLA DEF] "Preliminary Definition, High Level Architecture," briefing, Defense Modeling and Simulation Office, latest version on WEB.

[HLA OMT] "High Level Architecture Object Model Template", Defense Modeling and Simulation Office, latest version on WEB.

[HLA DDM] "HLA Data Distribution Management Design Document", Defense Modeling and Simulation Office, version release TBD.

[HLA TM] "HLA Time Management: Design Document", Defense Modeling and Simulation Office, latest version on WEB.

COMMENTS

Comments on this document should be sent by electronic mail to the Defense Modeling and Simulation Office (DMSO) HLA Specifications e-mail address (hla_specs@msis.dmso.mil). The subject line of the message should include the section and line numbers referenced in the comment. The body of each submittal should include: (1) the name and E-mail address of the person making the comment (separate from the mail header), (2) reference to the RTI service or portion of the I/F Specification that the comment addresses (by section number, page, and line number), (3) a one sentence summary of the comment/issue, (4) a brief description of the comment/issue, and (5) any suggested resolution or action to be taken.