High Level Architecture
Module 2
Advanced Topics

Roy Crosbie
John Zenor

California State University, Chico

High Level Architecture
Module 2
Advanced Topics

Management Object Model

Sources

- This lesson is based in part on
  - The paper by Deborah Fullford and Darren Wetzel, “A Federation Management Tool: Using the Management Object Model (MOM) to Manage, Monitor, and Control and HLA Federation”, SISO, 1999 Spring Simulation Interoperability Workshop
  - The Federate Interface Specification of the HLA (Draft Standard for Modeling and Simulation)

The Task to Be Addressed

- Federation managers must be able to debug, monitor and control federations by obtaining answers to the following typical questions:
  - Who has joined the federation?
  - When was the last federation save?
  - What is the current federation time?
  - What objects have been discovered?

Management Object Model (MOM)

- The MOM was designed to provide management information and control of the RTI, federation, and federates through its objects and interactions
- Using the MOM, a federate can
  - Obtain management data directly from the RTI
  - Control the federation through interactions
  - Extend the MOM to provide federation-specific management functions

The MOM's Objects

- The MOM consists of two object classes that are used to provide persistent data about the federation, the RTI, and individual federates:
  - ***Manager.Federation***
  - ***Manager.Federate***

Object class *Manager.Federate* contains attributes that describe the state of the federate. The RTI publishes the class and provides updates for these attributes periodically. One object of this class is registered by the RTI for each federate in the federation.

Object class *Manager.Federation* contains attributes that describe the state of the federation execution. One object instance of this class is published by the RTI for the federation.

(See part 11 of the Modeling and Simulation HLA-Federate Interface Specification)

Manager.Federation Object Class

- The attributes of *Manager.Federation* provide federation information such as:
  - Federation name
  - List of federates
  - FED file ID
  - RTI Version
  - Save status

All MOM object classes, interaction classes, attributes, and parameters are predefined in the FED file. MOM definitions may be extended.

MOM object classes may be extended by adding subclasses or class attributes. All extensions are handled by federates, not by the RTI.

Manager.Federation Attributes

- FederationName
- FederatesInFederation
- RTIversion
- FEDid
- LastSaveName
- LastSaveTime
- NextSaveName
- NextSaveTime

Object class *Manager.Federation*

Attribute	Type	Description
FederationName	string	Name of the federation to which the federat4e belongs
FederatesInFederation	handle list	Comma-separated list of the designators of federates that have joined the federation execution (null string if none)
RTIversion	string	Version of the RTI software
FEDid	string	Identifier associated with the FED data used by the federation
LastSaveName	string	Name associated with the last federation state save (null if no saves have occurred)
LastSaveTime	time	Logical time at which the last federation state save occurred (zero if no saves have occurred)
NextSaveName	string	Name associated with the next federation state save (null if no saves are scheduled)
NextSaveTime	time	Logical time at which the next federation state save is scheduled (zero if no saves are scheduled)

Manager.Federate Object Class

- The attributes of *Manager.Federate* provide:
  - Federate type and ID
  - Host name of computer
  - Time management information
  - State of the federate
  - Object and interaction information:
    - Number of objects and interactions
    - Number of interactions sent and received
    - Number of objects updated and reflected
    - Number of objects owned

One instance of this object class is created by the RTI for each federate in the federation. This class may extended by adding subclasses or class attributes, however these extensions are handled by the federates, not he RTI.

Manager.Federate Attributes

FederateHandle	LBTS
RTIversion	MinNextEventTime
TimeRegulating	ROlength
TimeManagerState	TSOlength
FederateType	ReflectionsReceived
FEDid	UpdatesSent
AsynchronousDelivery	InteractionsReceived
FederateTime	InteractionsSent
FederateHost	ObjectsOwned
TimeConstrained	ObjectsUpdated
FederateState	ObjectsReflected
Lookahead	

One instance of the object class *Manager.Federate* is created for each federate in the federation. See Part 11 of the *HLA Interface Specification* for details on the attributes for this class.

The MOM's Interactions

- There are four classes of MOM interactions:
  - **Adjust** interactions control aspects of the federation, federate, and the RTI
  - **Request** interactions obtain RTI information from another federate
  - **Report** interactions report RTI data about a federate; the RTI issues them in reply to Request interactions
  - **Service** interactions are used to invoke RTI services on behalf of another federate

All MOM interaction classes and parameters shall be predefined in the FED file. These classes and parameters may be extended by adding subclasses or parameters. These extensions are handled by the federates, not by the RTI.

Adjust Interactions

- The type of control available through the Adjust interactions include:
  - Timing of attribute updates
  - Ownership of attributes
  - Setting service and reporting logging

Manager.Federate.Adjust

- **Subclasses**
  - SetTiming
  - ModifyAttributeState
  - SetServiceReporting
  - SetExceptionLogging

Interaction Subclass	**Parameters**
SetTiming	Federate, ReportPeriod
ModifyAttributeState	Federate, ObjectInstance, Attribute, AttributeState
SetServiceReporting	Federate, ReportingState
SetExceptionLogging	Federate, LoggingState

Report and Request Interactions

- The type of information available through the Report and Request interactions include:
  - Subscription and publication information
  - Ownership information
  - Update and Reflection information
  - Alert status

Manager.Federate.Request

- Subclasses
  - RequestPublications
  - RequestSubscriptions
  - RequestObjectsOwned
  - RequestObjectsUpdated
  - RequestObjectsReflected
  - RequestUpdatesSent
  - RequestInteractionsSent
  - RequestReflectionsReceived
  - RequestInteractionsReceived
  - RequestObjectInformation

Interaction Subclass	**Parameters**
RequestPublications	Federate
RequestSubscriptions	Federate
RequestObjectsOwned	Federate
RequestObjectsUpdated	Federate
RequestObjectsReflected	Federate
RequestUpdatesSent	Federate
RequestInteractionsSent	Federate
RequestReflectionsReceived	Federate
RequestInteractionsReceived	Federate
RequestObjectInformation	Federate

Interaction Class
Manager.Federate.Report

Subclasses

ReportObjectPublication	ReportUpdatesSent
ReportInteractionPublication	ReportReflectionsReceived
ReportObjectSubscription	ReportInteractionsSent
ReportInteractionSubscription	ReportInteractionsReceived
ReportObjectsOwned	ReportObjectInformation
ReportObjectsUpdated	Alert
ReportObjectsReflected	ReportServiceInvocation

Interaction Subclass	**Parameters**
ReportObjectPublication	Federate, NumberOfClasses, ObjectClass, AttributeList
ReportInteractionPublication	Federate, InteractionClassList
ReportObjectSubscription	Federate, NumberOfClasses, ObjectClass, Active, AttributeList
ReportInteractionSubscription	Federate, InteractionClassList
ReportObjectsOwned	Federate, ObjectCounts
ReportObjectsUpdated	Federate, ObjectCounts
ReportObjectsReflected	Federate, ObjectCounts
ReportUpdatesSent	Federate, TransportationType. UpdateCounts
ReportReflectionsReceived	Federate, TransportationType, ReflectCounts
ReportInteractionsSent	Federate, TransportationType, InteractionCounts
ReportInteractionsReceived	Federate, TransportationType, InteractionCounts
ReportObjectInformation	Federate, ObjectInstance, OwnedAttributeList, RegisteredClass, KnownClass
Alert	Federate, AlertSeverity, AlertDescription, AllertID
ReportServiceInvocation	Federate, Service, Initiator, SuccessIndicator, SuppliedArgument1-5, ReturnedArgument, ExceptionDescription, ExceptionID

Service Interactions

- The type of control available through the Service interactions include:
  - Resignation of federates
  - Saving and restoring of a federation
  - Publication and subscriptions of federates
  - Setting ownership and transportation of attributes
  - Setting federates time management parameters

Interaction Subclass	Parameters
ResignFederationExecution	Federate
SynchronizationPointAchieved	Federate, Label
FederateSaveBegun	Federate
FederateSaveComplete	Federate, SuccessIndicator
FederateRestoreComplete	Federate, Label, SuccessIndicator
PublishObjectClass	Federate, AttributeList
UnpublishObjectClass	Federate, ObjectClass
PublishInteractionClass	Federate, InteractionClass
UnpublishInteractionClass	Federate, InteractionClass
SubscribeObjectClassAttributes	Federate, ObjectClass, AttributeList, Active
UnsubscribeObjectClass	Federate, ObjectClass
SubscribeInteractionClass	Federate, InteractionClass, Active
UnsubscribeInteractionClass	Federate, InteractionClass
DeleteObjectInstance	Federate, ObjectInstance, Tag, FederationTime

This table is continued on the next slide's note page

Interaction Class
Manager.Federate.Service

Subclasses

LocalDeleteObjectInstance	DisableTimeConstrained
ChangeAttributeTransportationType	EnableAsynchronousDelivery
ChangeAttributeOrderType	DisableAsynchronousDelivery
ChangeInteractionTransportationType	ModifyLookahead
ChangeInteractionOrderType	TimeAdvanceRequest
UnconditionalAttributeOwnership Divestiture	TimeAdvanceRequestAvailable
EnableTimeRegulation	NextEventRequest
DisableTimeRegulation	NextEventRequestAvailable
EnableTimeConstrained	FlushQueueRequest

See Previous Page

Interaction Subclass	Parameters
LocalDeleteObjectInstance	Federate, ObjectInstance
ChangeAttributeTransportation Type	Federate, ObjectInstance, AttributeList, TransportationType
ChangeAttributeOrderType	Federate, ObjectInstance, AttributeList, OrderingType
ChangeInteractionTransportation Type	Federate, InteractionClass, TransportationType
ChangeInteractionOrderType	Federate, InteractionClass, OrderingType
UnconditionalAttributeOwnership Divestiture	Federate, ObjectInstance, AttributeList
EnableTimeRegulation	Federate, FederationTime, Lookahead
DisableTimeRegulation	Federate
EnableTimeConstrained	Federate
DisableTimeConstrained	Federate
EnableAsynchronousDelivery	Federate
DisableAsynchronousDelivery	Federate
ModifyLookahead	Federate, Lookahead
TimeAdvanceRequest	Federate, FederationTime
TimeAdvanceRequestAvailable	Federate, FederationTime
NextEventRequest	Federate, FederationTime
NextEventRequestAvailable	Federate, FederationTime
FlushQueueRequest	Federate, FederationTime

Federation Management Tool

- Federation Management Tool (FMT)
  - Java based tool providing intuitive access to MOM data
  - Implemented by Mäk Technologies
    - Based on original tool developed by MITRE
    - Available “free” from the DMSO Web site
  - Four modes for viewing federation or individual federate
    - Time Mode
    - Object Mode
    - Communication Mode
    - File Mode

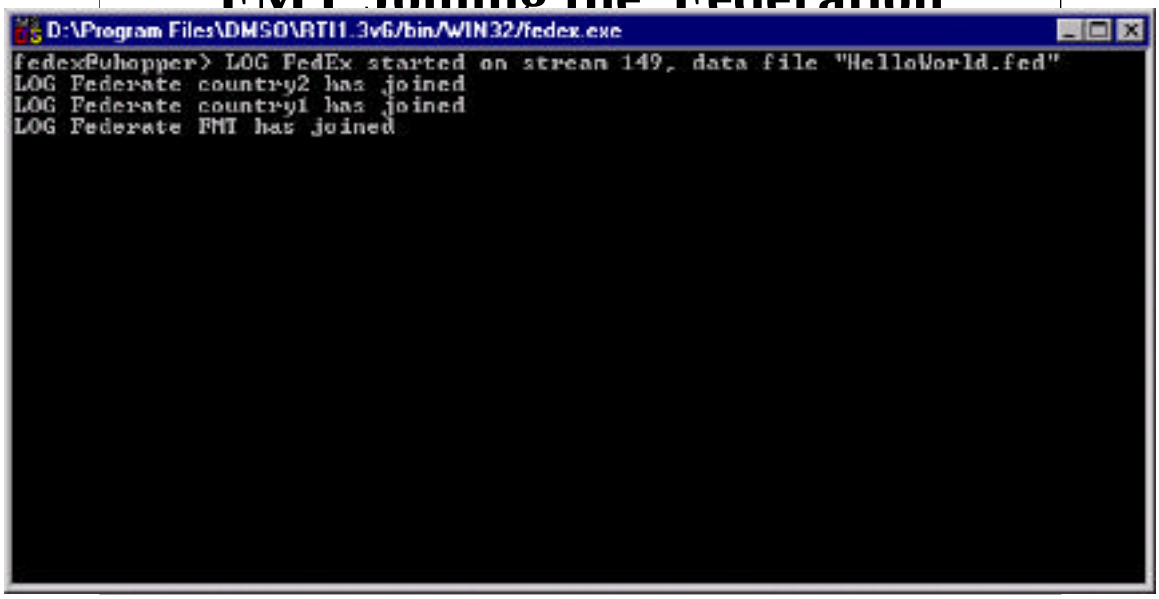
Installation of the FMT tool requires:

RTI 1.3V6

JDK 1.2

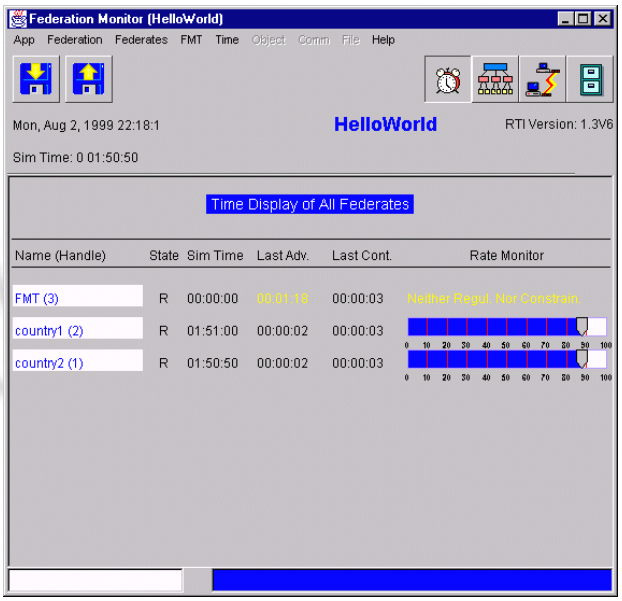
Several views from this tool follow. For this example, the helloWorld federation supplied with the RTI 1.3V6 release was started with two countries, then the FMT was started.

FMT Joining the Federation



This is a screen snapshot of the command window for the federation executive, *fedex*. It shows the Federation Management Tool joining the *helloWorld* federation after *country1* and *country2* have been running. Time is advanced as quickly as possible in this simulation and is not regulated in any way.

FMT Time Display

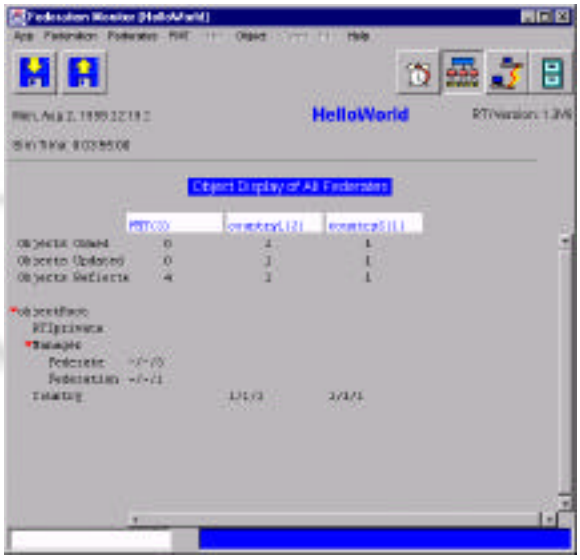


This shows the FMT time mode display for the *helloWorld* federation. It shows the simulation time and rate for the 2 *country* federates.

FMT Federation Simulation Rate

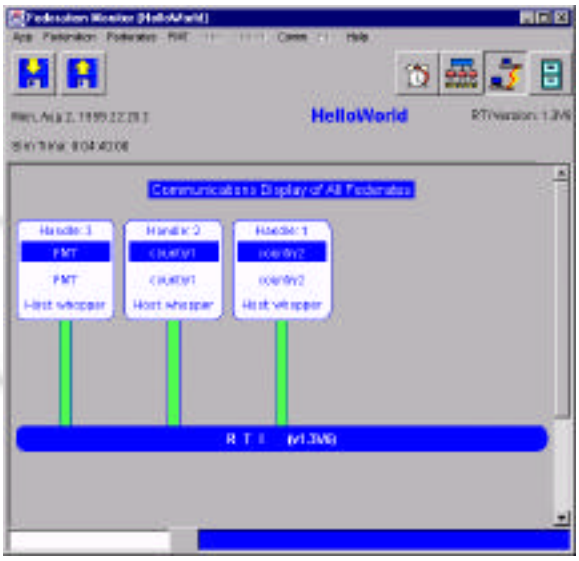
This snapshot of the timing display shows the unregulated rate of time advancement in the helloWorld simulation as it is running with two country federates on a 200 MHz Pentium MMX.

FMT Federation Object View

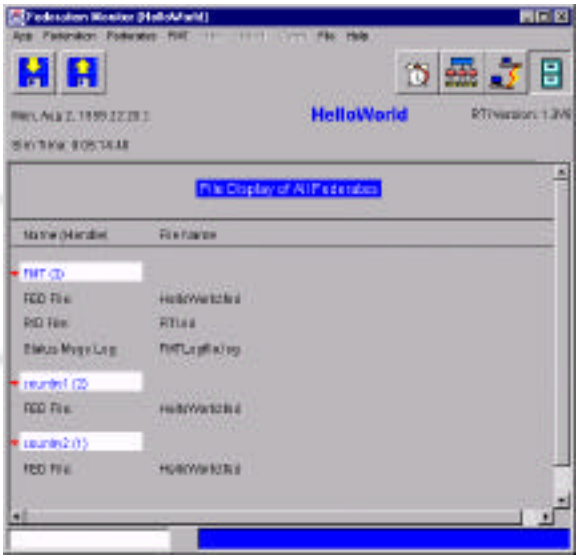


The **FMT Federation Object View** is shown for the *helloWorld* federation. It shows the number of objects owned, updated, and reflected by each federate. Each running *country* federate in the *helloWorld* federation owns (and simulates) a single *country* object, reflecting the data from other *country* federates. The FMT owns (and updates) no objects of its own, but reflects data from all running federates.

FMT Federation Network Information



FMT Federation File Display



The **FMT Federation File Display** shows the files associatedwith each federate in the federation.

FMT Federation Simulation Rate

This timing display snapshot was taken at a later time in the simulation, showing that the unregulated simulation rate is not constant.

MOM Summary

- The Management Object Model (MOM) provides a mechanism for monitoring and control of the federation
- Interfaces with the federation as an object model, **not** as a set of RTI services
  - Interacts with federation using normal updates and interactions like any other model
- The Federation Management Tool (FMT) uses MOM for federation monitoring and control
  - Intuitive, visual user interface (Java GUI & C++ Code)
  - Implemented as a special-purpose federate

An interesting description of the FMT and the design tradeoffs encountered in its implementation may be found in:

Development of a Federation Management Tool: Implications for HLA, Dave Prochnow, Ernest H. Page, Bryan Youmans, MITRE Corporation,