



HLA Federation Development and Execution Process (FEDEP) & Supporting Tools

Bob Lutz and Chris Turrell

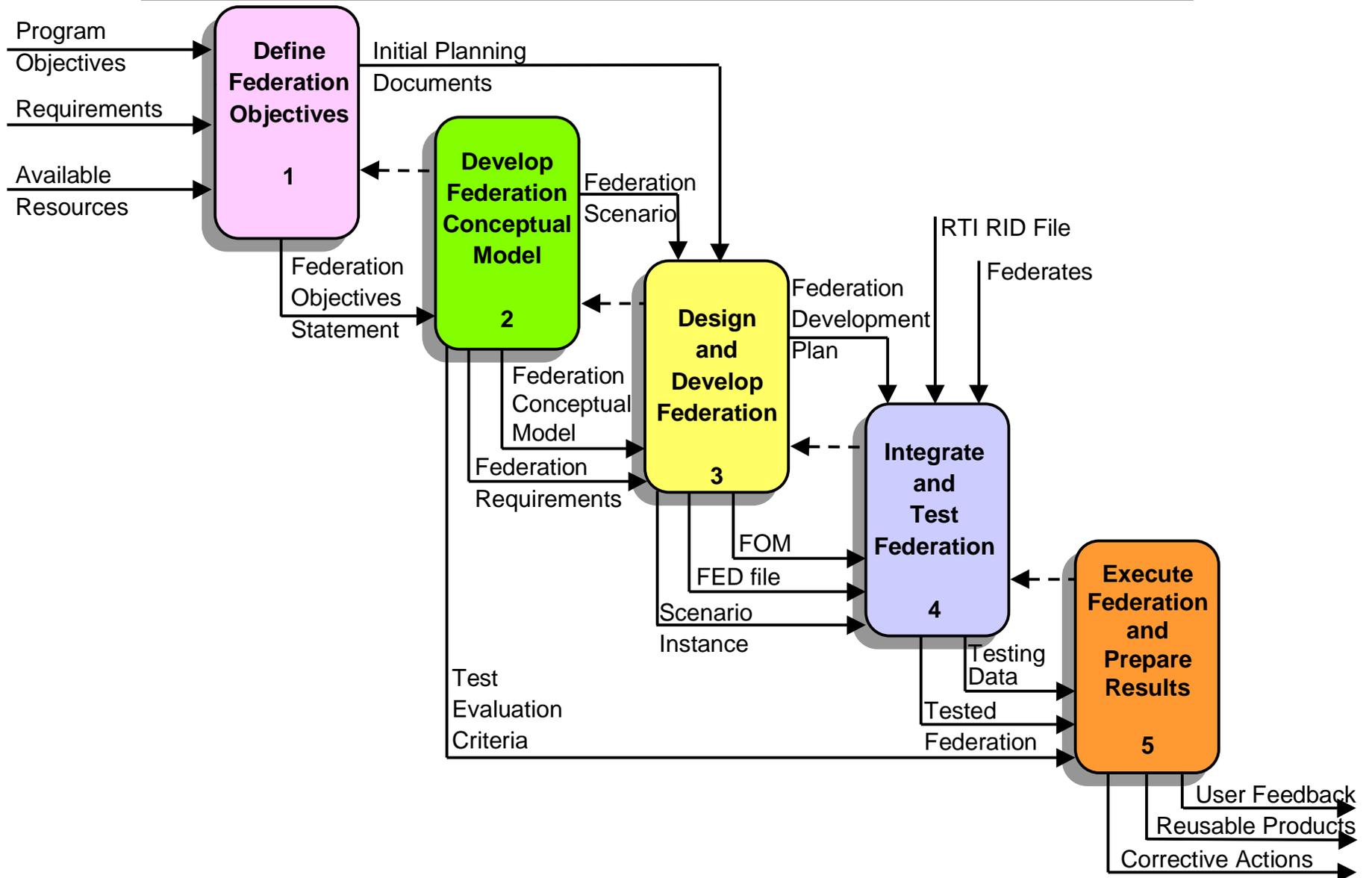
Southwest Regional Training

27 May 1999

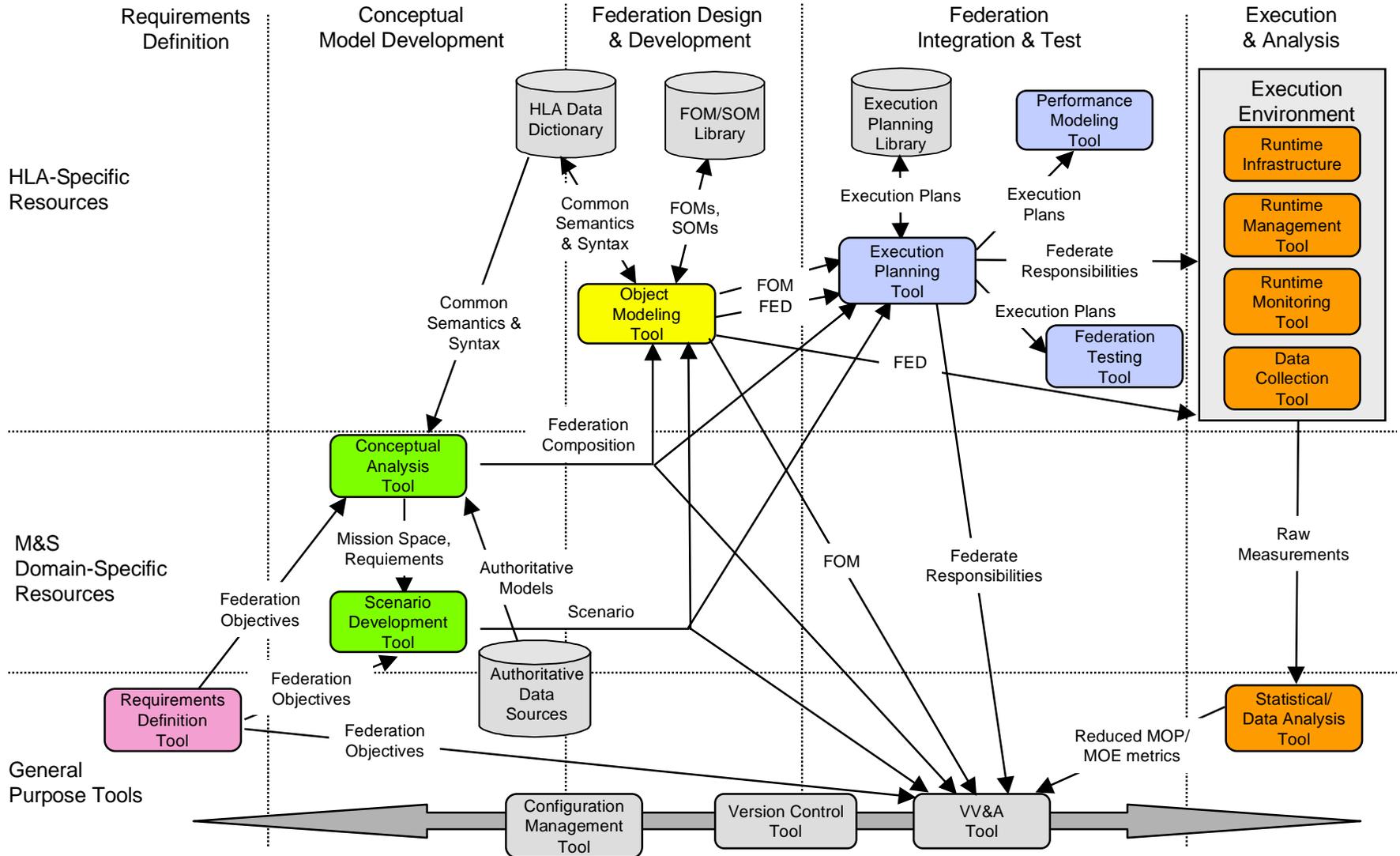
Description

- **The FEDEP ...**
 - **defines a generic framework for distributed simulation development**
 - **consists of a graphical model and supporting textual description**
 - **provides an organizing mechanism for open discussion of federation development practices and strategies at the SISO PROC Forum (and related forums)**
 - **provides a common foundation for defining functional overlays to the federation development process (e.g., VV&A, security, tools)**
 - **provides a clear, concise introduction to federation development for new HLA users**
 - **provides a common reference point for communication between federation development team members**

Five Step Process



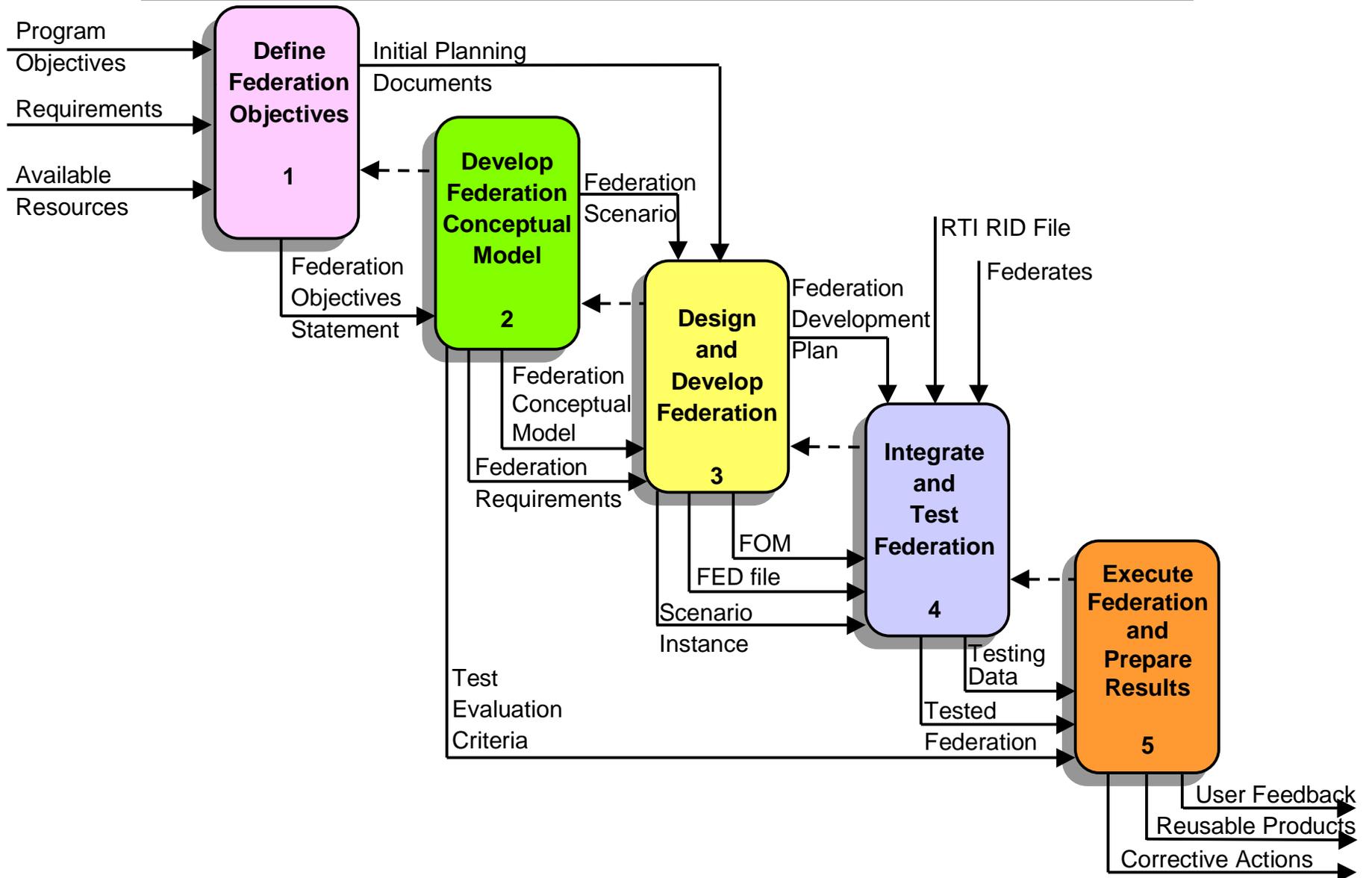
HLA Tool Architecture



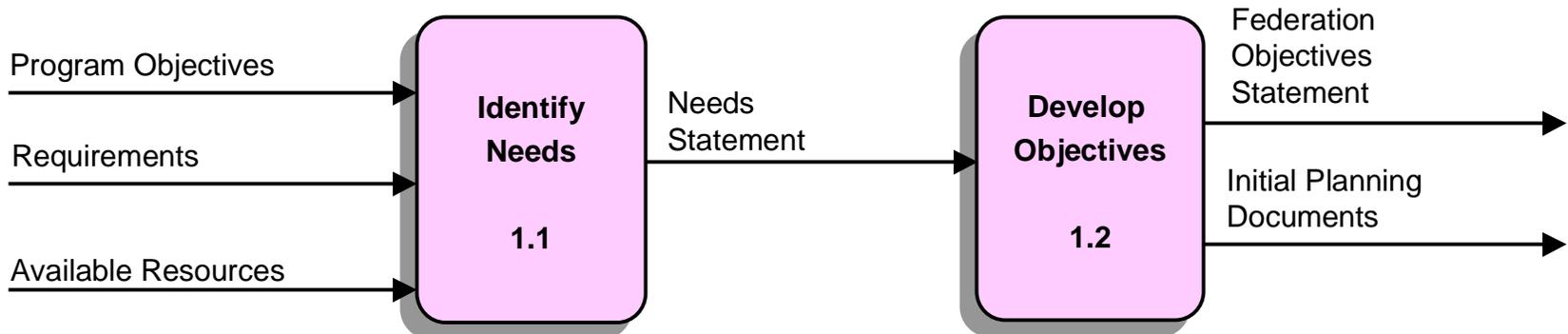
Tutorial Structure

- **A structured view of the FEDEP model**
- **Introduction to a problem set and walk through the FEDEP process to demonstrate and discuss the major products generated at each step**
- **Outline the tools that are applicable at each step of the FEDEP and discuss the alternatives that are available.**

Five Step Process



Define Federation Objectives



1.1 Identify Needs

Develop a high-level description of the problem space from the perspective of the User-Sponsor

Technical

- Key systems
- Key events
- Desired outputs
- Schedule

Programmatic

- Resource constraints
- Facilities
- Funding
- Schedule

Sample: Federation Needs Statement (1 of 2)

- **Needs Statement:**
 - **Analyze the ability to conduct joint strike operations against high value targets (command centers, POL, ...) that are protected by ground-based defensive forces.**
 - **Define a realistic force laydown with postulated 2010 threat system capabilities and associated tactics and doctrine. Utilize the Defense Planning Guidance (DPG) scenarios to the maximum extent possible.**
 - **Determine the relative performance and expected cost of different sensor, countermeasure, and weapon system technology options.**
 - **Provide explicit representations of postulated Command, Control, and Communications (C3) equipment and networks.**
 - **Ensure that environmental conditions can be varied within the simulation environment.**
 - **.....**

Sample: Federation Needs Statement (2 of 2)

- **Constraints:**
 - The simulation environment must be prepared within a 12-month period, beginning at contract initiation.
 - The total development cost for the simulation environment shall not exceed \$5.2M
 - The analysis shall be conducted at the National Test Facility (NTF). NTF resources shall be utilized to the maximum extent possible in the development of the simulation environment.
 - The security level at which the analysis is conducted must not exceed SECRET.
 - Data collected during the execution of the simulation must indicate the relative performance of each system option as measured by appropriate mission success parameters (e.g., ground targets killed, aircraft lost, etc.).
 -
 -

1.2 Develop Objectives

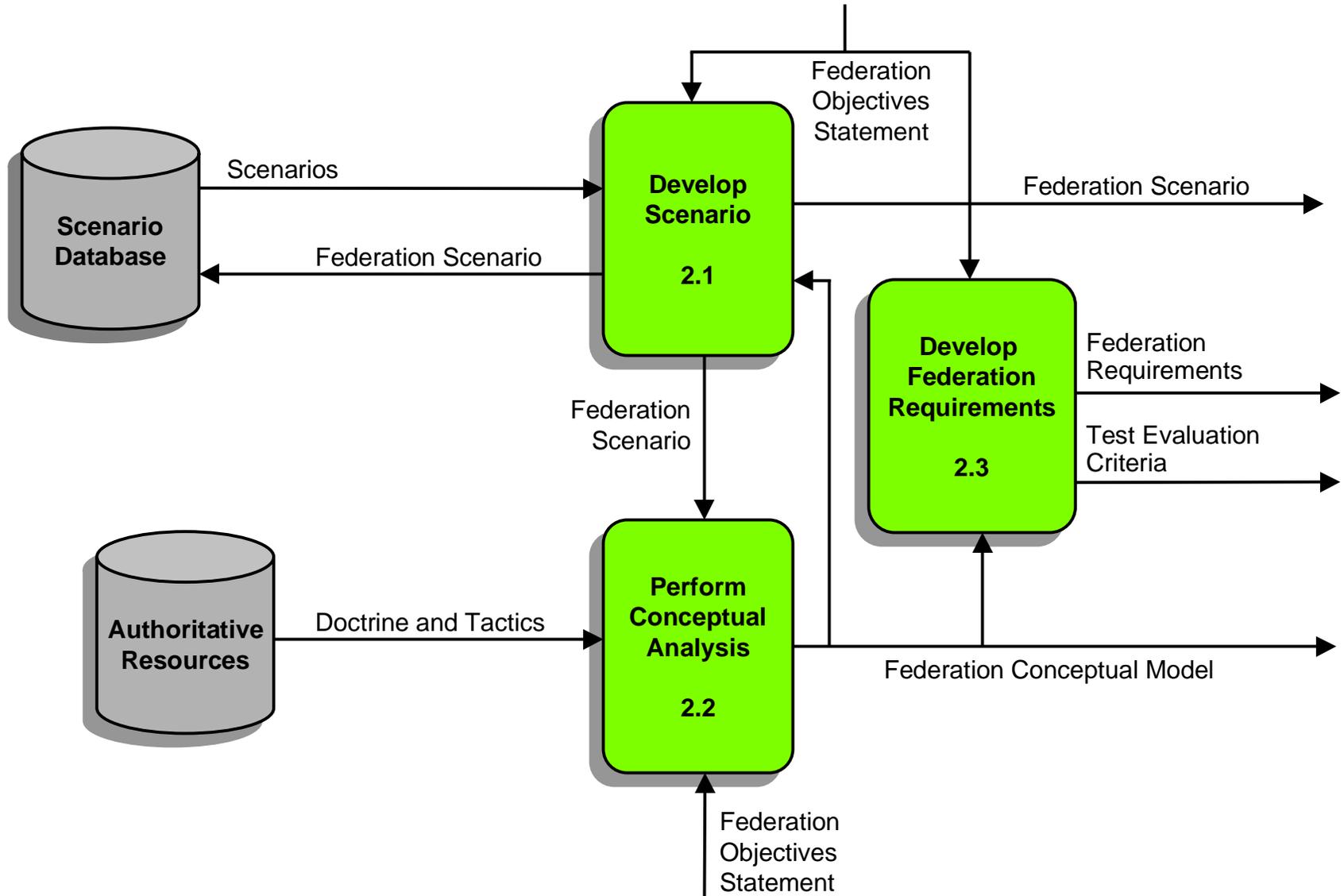
Transform high-level User/Sponsor expectations (as defined in the Needs Statement) into a concrete description of federation objectives

- Activity by which User/Sponsor clearly and unambiguously communicates his/her intent to the initial federation development team; usually achieved over **several iterations**
- Provides the additional detail necessary to begin early federation development activities
- Typically involves an **early assessment of feasibility and risk**
- Also involves **development of initial planning documents** to guide early federation development activities (e.g., program milestones, security, VV&A, tools)

Sample: Federation Objectives Statement

- **Federation Objectives Statement:**
 - **Construct a realistic, faster-than-real-time simulation environment which supports both offensive strike and defense suppression missions at the entity level of representation.**
 - **Provide an accurate representation of an airborne command center (AWACS) in the simulation environment, through which all simulated military operations will be directed and coordinated.**
 - **Develop a reactive capability (as necessary) for all simulated forces. Since there are no humans-in-the-loop, all decision-making must be automated.**
 - **Create an initial scenario which:**
 - **occurs in year 2020**
 - **is located in Korea (summer season), and**
 - **includes both day and night operations**
 - **Identify both Air Force and Navy assets**

Develop Federation Conceptual Model

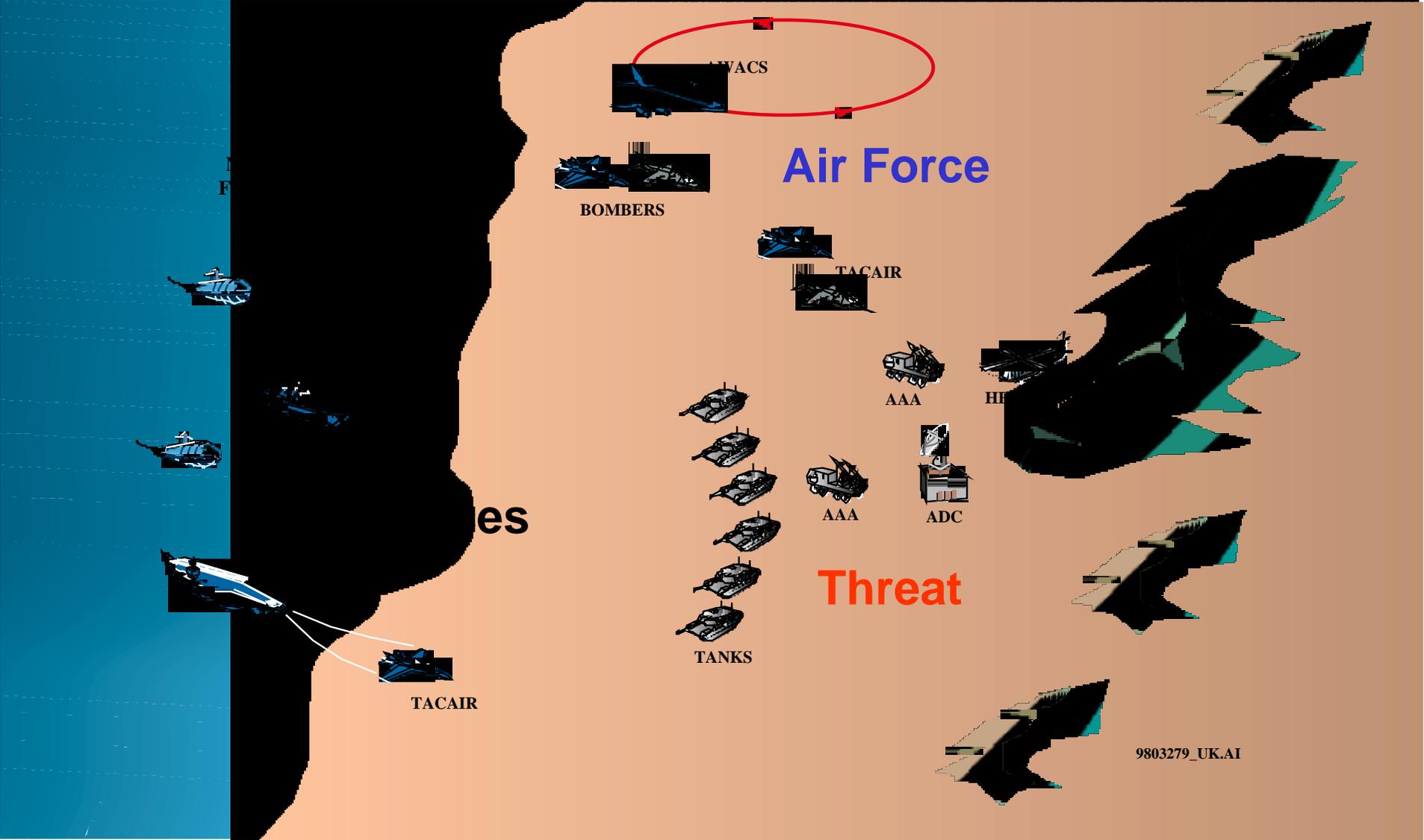


2.1 Develop Scenario

Develop the storyline describing the sequence of activities that when executed in the federation will achieve the federation objectives

- **Identifies specific geographical areas and environmental conditions**
- **Defines specific force laydowns and time-sequencing of key events necessary to fully represent the problem space**
- **Specifies initial/termination conditions (i.e., length of scenario run)**
- **Provides boundary conditions on the extents of the federation conceptual model**
- **May be recorded in many different forms**

Sample: Federation Scenario

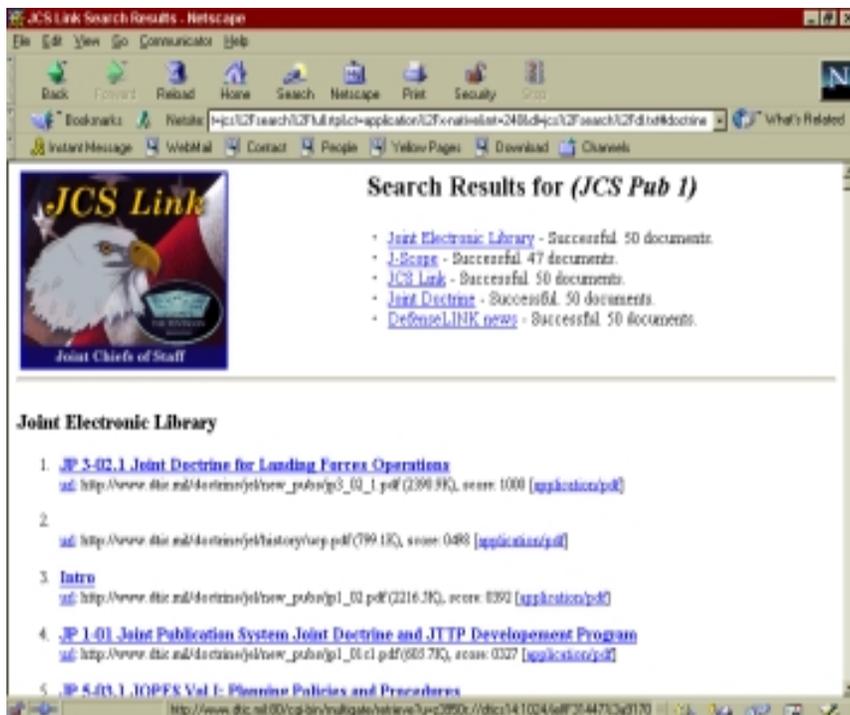


2.2 Perform Conceptual Analysis

Develop a “real world” description of the entities and actions that must be represented in the federation environment

- Based on authoritative sources
- Defines the **specific missions or operations** necessary to accomplish defined goals
- Defines **relationships between entities**, and the **behaviors** which may be exhibited by those entities
- **Independent** of the capabilities offered by potential federates
- Can be expressed in many different forms
- Provides the real world domain representation necessary for scenario development

Sample: Conceptual Model for Strike Mission

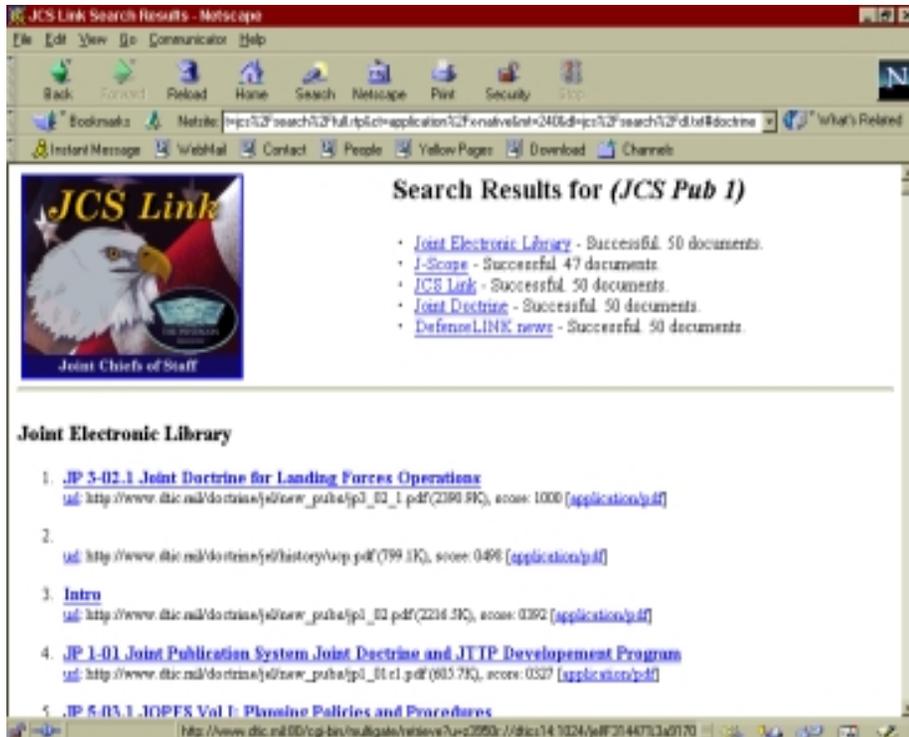


The purpose of the strike mission is to **destroy** high-valued targets deep in enemy territory while **avoiding**, **degrading**, or directly **engaging** hostile air defense assets encountered enroute to the target area. Many different categories of offensive assets can be employed in support of a strike mission, such as **airborne surveillance aircraft** for general battlefield coordination, **bomber/attack aircraft** for **delivery of ordnance on target**, and fighter escort **aircraft** for protection of the attack force.

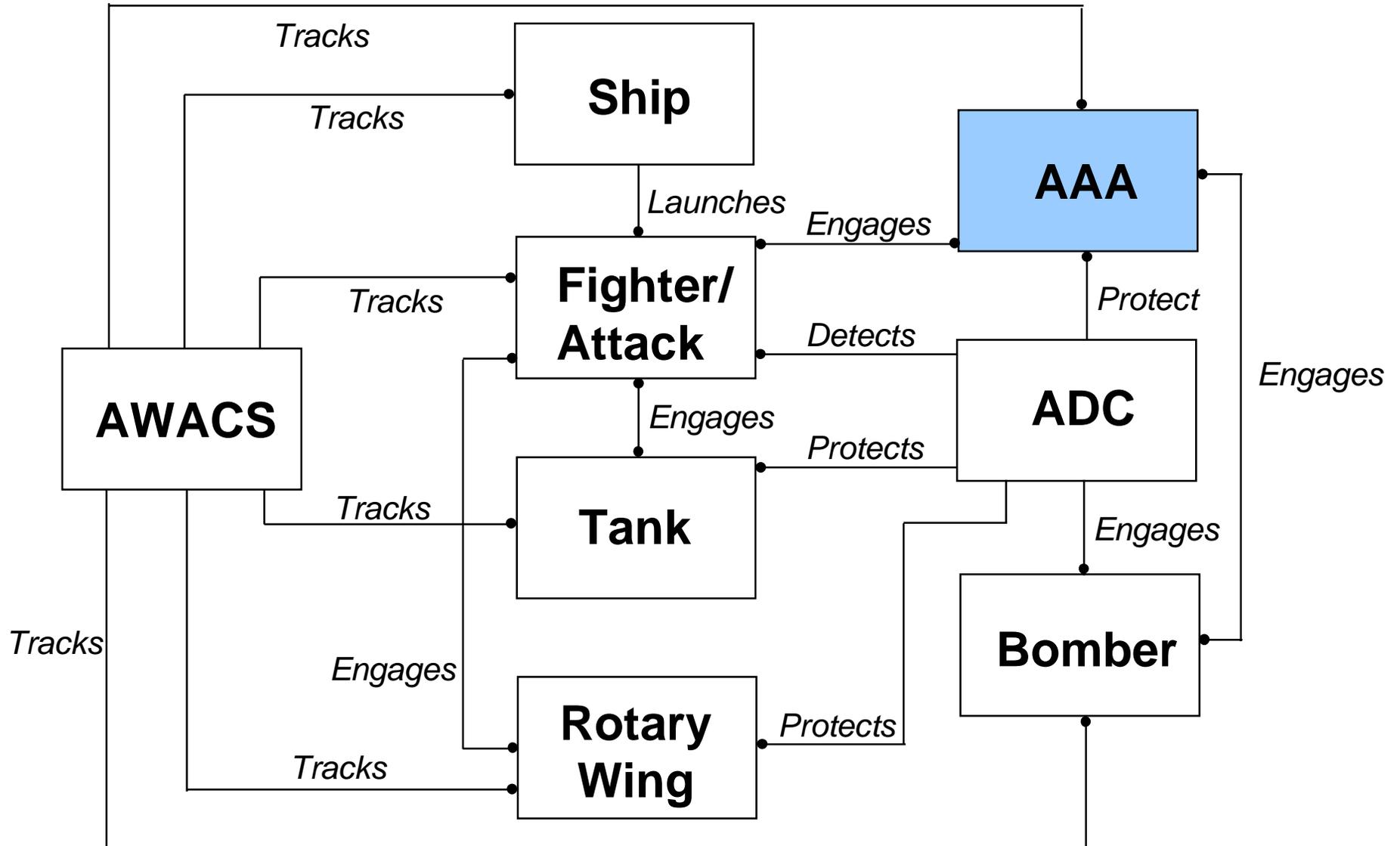
There are also many different offensive tactics that can be used in a strike mission, depending on the perceived strength of the enemy air defense network and the types of offensive assets available ...

Sample: Conceptual Model for Air Defense

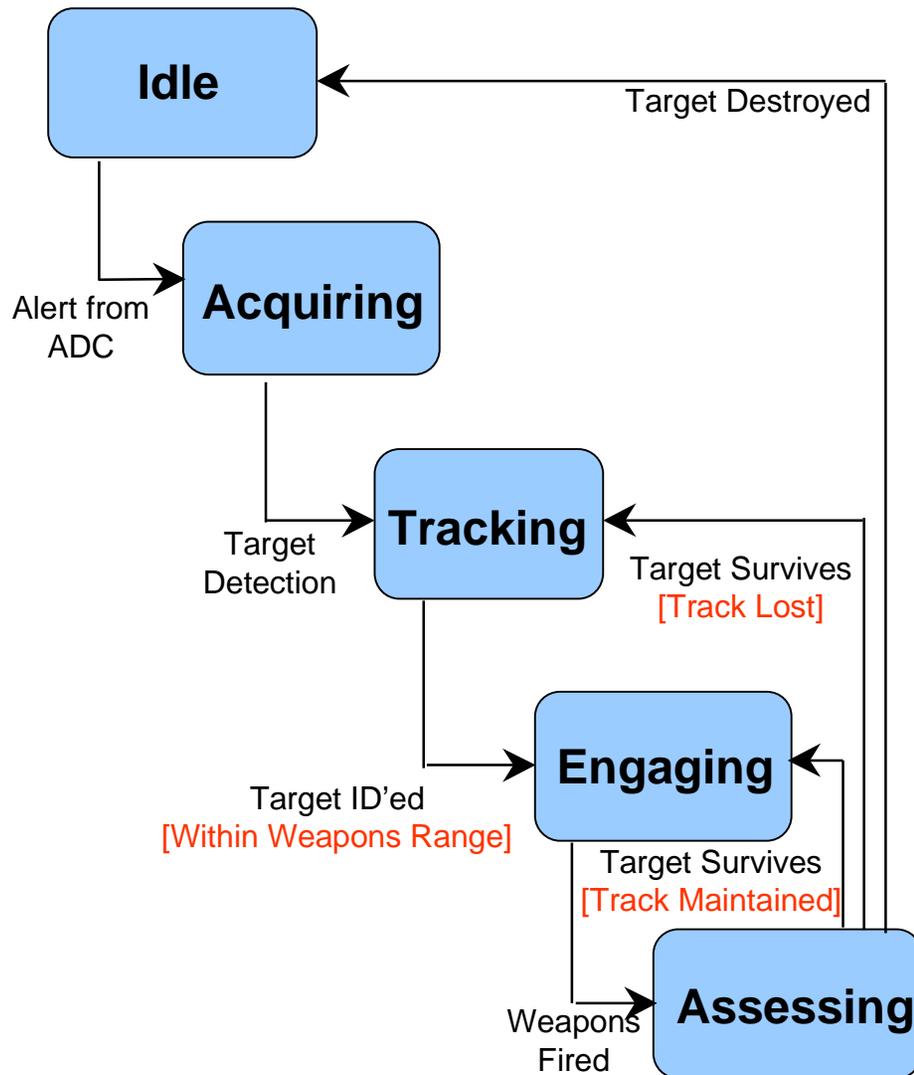
The purpose of the air defense mission is to **detect**, **track**, and **identify** all air vehicles entering the assigned air space, and to **intercept** and **destroy** hostiles soon after they enter defense sensor coverage. A typical **ground-based air defense** network consists of one or more **Air Defense Centers (ADCs)**, each with an **Early Warning (EW) radar system**. Each ADC is linked to a number of **Surface-to-Air Missile (SAM)** or **Anti-Aircraft Artillery (AAA)** sites via an integrated C3 network. As long-range detection occurs via the EW radars, the ADCs will assign one or more SAM or AAA sites to track and identify the target ...



Sample: Conceptual Model - Graphical (Static View)



Sample: Conceptual Model for AAA Site



The function of an **Anti-Aircraft Artillery (AAA) site** is to provide **point defense** for targets considered to be of high-value to enemy forces. A AAA Site is typically in an “**Idle**” mode unless a report of a potential hostile aircraft is received from an Air Defense Center (ADC) equipped with a long range radar system. When a report is received, the AAA site will immediately attempt to **acquire** the target using its Fire Control System (FCS). Once the target is **detected**, the AAA Site will begin to derive a fire control solution on the target by determining the contact’s heading and speed ...

2.3 Develop Federation Requirements

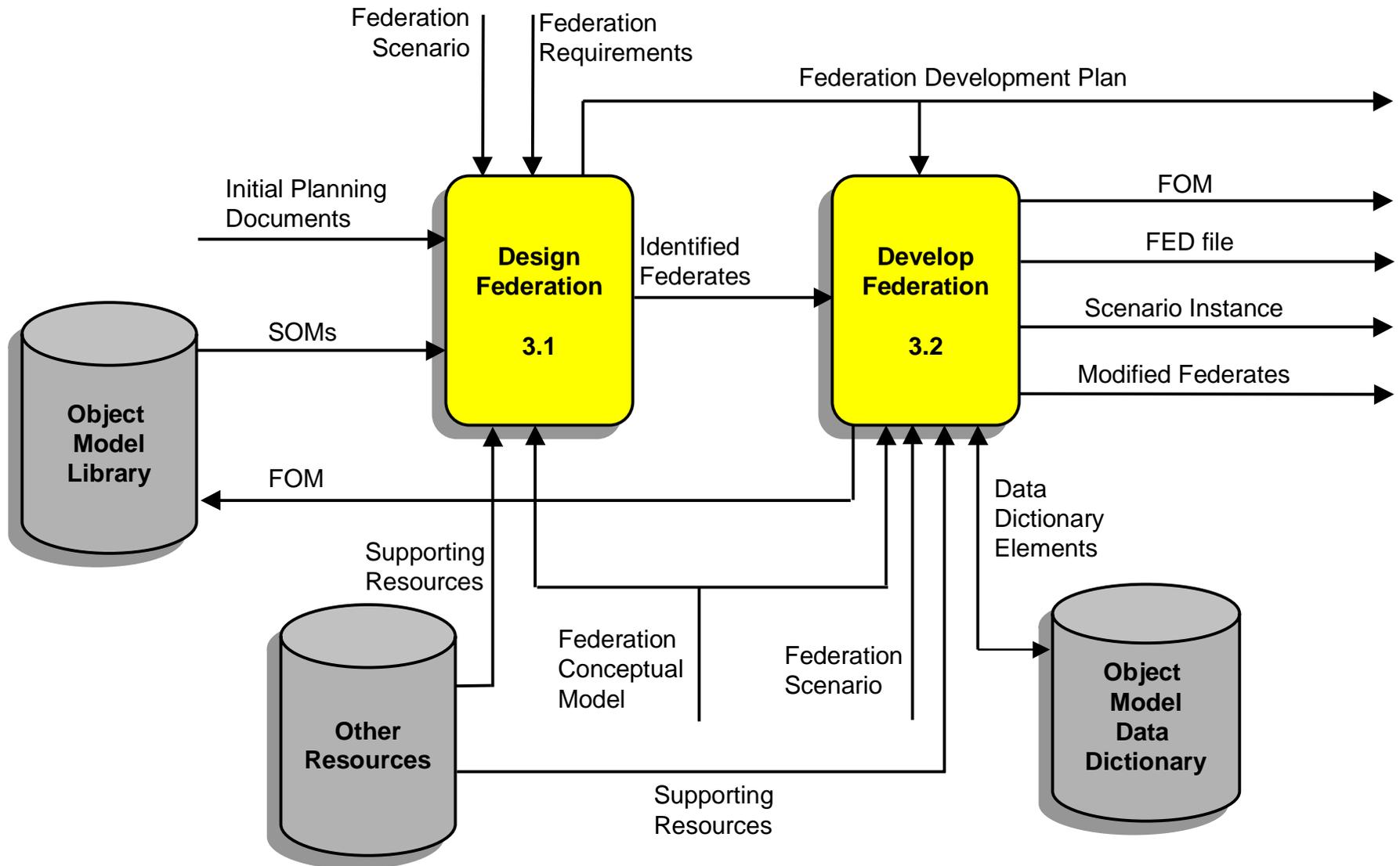
Develop a set of detailed federation requirements to guide subsequent federation design, development, and implementation activities

- **Refinement of federation objectives**
- **Goal is to make them explicitly testable**
 - **Provide criteria for federate selection**
 - **Provide success criteria during federation testing activities**
- **May be further refined during federation design and development**

Sample: Federation Requirements

- **Federation Requirements:**
 - AWACS aircraft should be represented at **six degrees of freedom**, and must possess the capability to **track both air and ground forces** (offensive and defensive) with sufficient resolution (**normally 0.5 degrees**) to coordinate all offensive operations. Explicit representations of a **Synthetic Aperture Radar (SAR)** system and **JTIDS communications equipment** must be included in the **AWACS avionics configuration**.
 - Fighter-attack aircraft must be represented at six degrees of freedom, and must possess both **high resolution radar**, **IR imaging sensors**, and appropriate **air-to-air/air-to-ground weaponry** for escorting a bomber force through defended territory.
 - AAA sites should possess a **dual mode (acquisition/tracking)** fire control radar system and weaponry (artillery size, rate-of-fire, ...) typical of postulated **2020** Korean air defense systems.

Design and Develop Federation

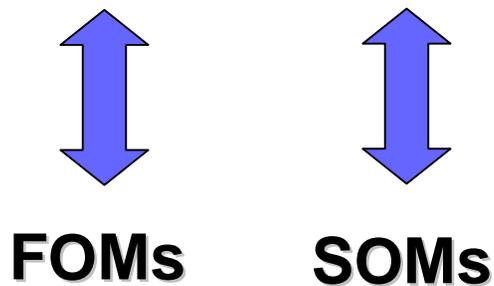
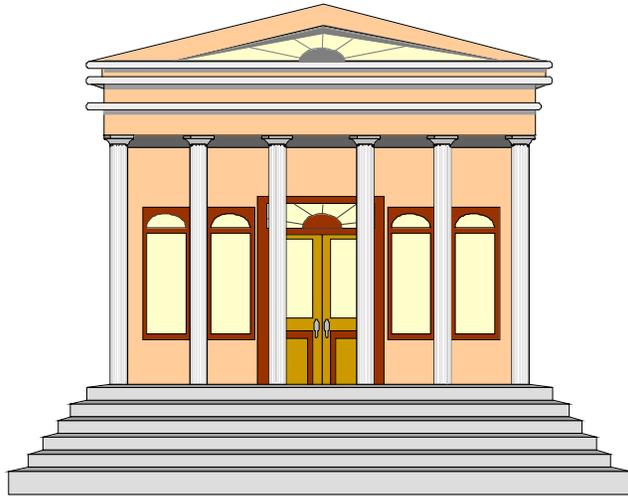


3.1 Design Federation

Establish the membership of the federation and apply a systems engineering approach to produce a federation that meets the requirements.

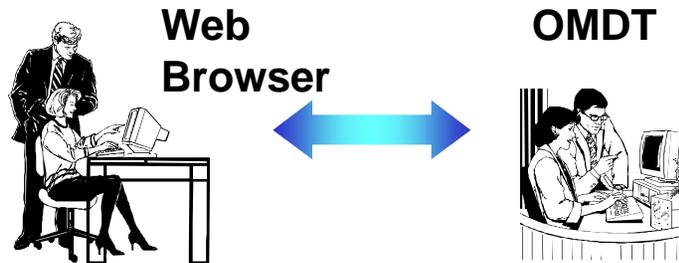
- **Identify candidate federates**
 - **Determine key required functionality and compare to simulations with known capabilities**
- **Evaluate candidates and select best choices**
- **Assign responsibility for required functionality to each federate**
- **Prepare federation development plan**

OML Purpose

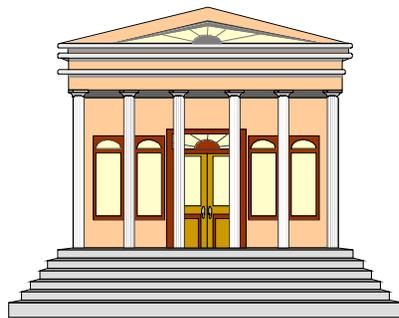


- Provides a central library to support the reuse of FOMs and SOMs
- Supports the Federation Development and Execution Process (FEDEP)
 - Provides a basis for the evaluation of existing federates as possible federation members by the examination of their SOMs
 - Permits access to persistent and “reference” FOMs and SOMs
 - Supports the piece-part creation of new FOMs and SOMs from existing FOMs and SOMs
 - Provides existing FOMs and SOMs as design examples
- Provides developers and sponsors with a means of advertising their federate/federation capabilities

OML Functionality



OMT DIF



OML Server

The OML is a WWW-based application which provides:

- Searching across all FOMs and SOMs in the library
- Browsing within an individual FOM or SOM
 - Web pages correspond to the OMT tables (routing space table, object classes, attributes, etc.)
 - Hyperlinks between OMT components
- Checking in new FOMs/SOMs to the library
- Checking out a copy of a FOM/SOM from the library using a standard OMT Data Interchange Format (DIF)
- Editing additional OM metadata to fully describe a FOM/SOM
- Registering owners of FOMs/SOMs

Sample: Using OML to Identify Candidate Federates

The screenshot shows the HLA Object Model Library (OML) website in a Netscape browser window. The browser title is "HLA Object Model Library: Welcome - Netscape". The address bar shows the URL: <http://www.omlibrary.epgc4i.com/app/main.cfm?CFID=6275&CFTOKEN=11534>. The website has a navigation menu with links for Search, Models, Utilities, About, and MSRR. The main content area displays a list of models:

Model Name	Version	Category	Date	Description	Details	Check-out
Eagle SOM	2.0	SOM	02-Oct-97	HLA Federation Prototyping	Details	Check-out
Engineering Federation	1.91	FOM	27-Aug-97	???	Details	Check-out

Below the model list, there is a section titled "Class List for: *Eagle SOM (2.0)*". This section contains a table with the following columns: Class Name, PS Capabilities, and Description.

Class Name	PS Capabilities	Description
AFloatPlayer	PS	A class for all subject that reside on the surface of water.
AggregateGroundPlayer	PS	An aggregate platform class for objects on the ground.
AIR DEFENSE UNITS	P	(THIS IS A THIRD LEVEL CLASS INHERITING FROM GROUND MOVERS IT INCLUDES ALL RESOLUTION UNIT AIR DEFENSE UNITS SUCH AS ADA BATTERY)

The browser's status bar at the bottom indicates "Document: Done".

3.1 Design Federation

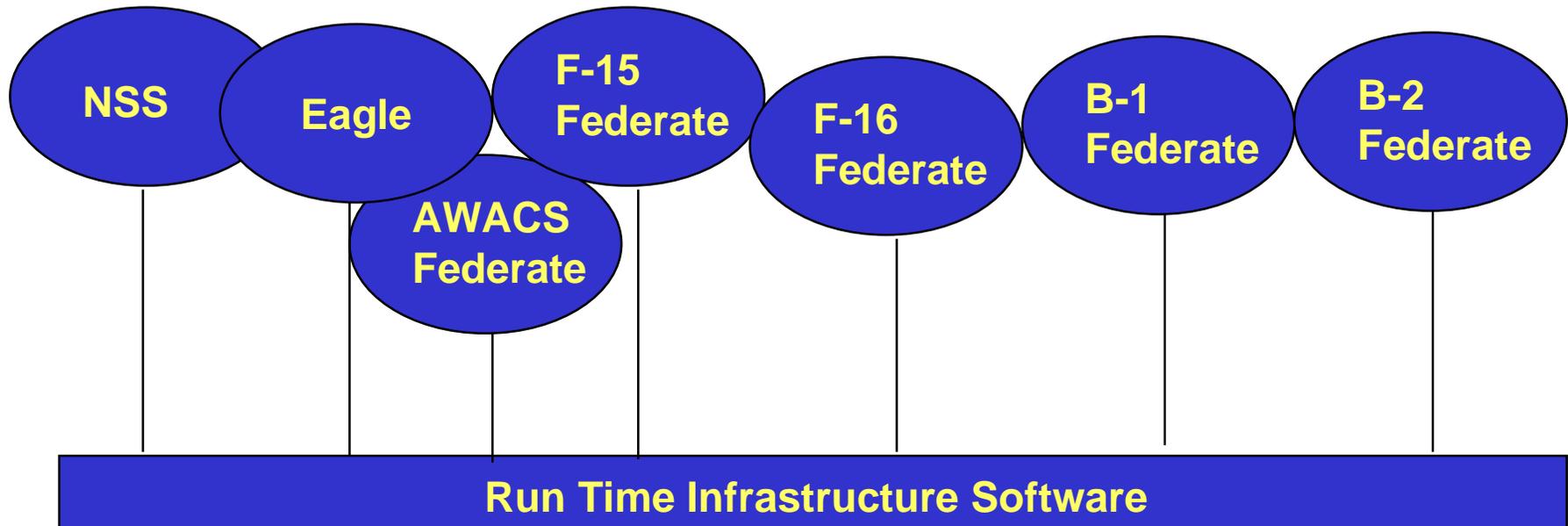
Establish the membership of the federation and apply a systems engineering approach to produce a federation that meets the requirements.

- Identify candidate federates
- Evaluate candidates and select best choices
- Assign responsibility for required functionality to each federate
- Prepare federation development plan

Evaluate Candidates and Select Best Choices

Operational Criteria	Technical Criteria	Logistical Criteria
<ul style="list-style-type: none">• Ability to support the conceptual model• Credibility• Ability to support MOEs and MOPs• Explicit representations of command and control• Changeable unit and system behaviors	<ul style="list-style-type: none">• Number of objects the federate can represent• Duration of run supported by federate• Previous interfaces to other simulations including human and hardware-in-the-loop• HLA compliance	<ul style="list-style-type: none">• Scenario availability• Ability to run closed loop and faster than real time• Portability• Ability to distribute• Ability to operate at collateral SECRET• Availability of developer support

Sample: Federate Selection Results



3.1 Design Federation

Establish the membership of the federation and apply a systems engineering approach to produce a federation that meets the requirements.

- Identify candidate federates
- Evaluate candidates and select best choices
- **Assign responsibility for required functionality to each federate**
- Prepare federation development plan

Sample: Federate-Class Mapping

	Objects								Interactions			
	Ship	AAA	Tank	AWACS	ADC	F/A	Bomber	Rotary Wing	Detected Target	Engaging Target	Weapon Detonate	Target Destroyed
Eagle		X	X		X			X			X	X
NSS	X					X			X	X	X	X
AWACS Simulator				X								
F-15 Simulator						X			X	X	X	X
F-16 Simulator						X			X	X	X	X
B-1 Simulator							X			X	X	X
B-2 Simulator							X			X	X	X

3.1 Design Federation

Establish the membership of the federation and apply a systems engineering approach to produce a federation that meets the requirements.

- Identify candidate federates
- Evaluate candidates and select best choices
- Assign responsibility for required functionality to each federate
- Prepare federation development plan
 - Guides the remainder of federation activities for individual federates and full federation
 - Extends the planning initiated during the development of federation objectives

Prepare Federation Development Plan

- **Get developer teams committed to the project**
 - Coordinate with federate owner manager; draft MOA if needed
 - Insure availability of s/w engineer and operator personnel
 - Draft SOW and cost estimate
- **Coordinate with sponsor to identify trade-offs, if necessary**
 - For some federates, the amount of work may exceed available time and budget
- **Identify useful run-time tools and versions to be used by federation developers**
- **Prepare a detailed development plan**
 - Schedule for individual federate developments
 - Milestones and dates for federation-wide integration and test activities

At this stage, the federation team has expanded to include representatives from all federates.

3.2 Develop Federation

Develop the set of software and data products necessary to support the federation application

- **Develop the Federation Object Model (FOM)**
- **Resolve other interoperability issues**
 - Shared data and algorithms
 - Federation operating agreements (as necessary)
- **Implement required federate modifications**
- **Develop data collection strategy**
- **Build simulation input files or databases**

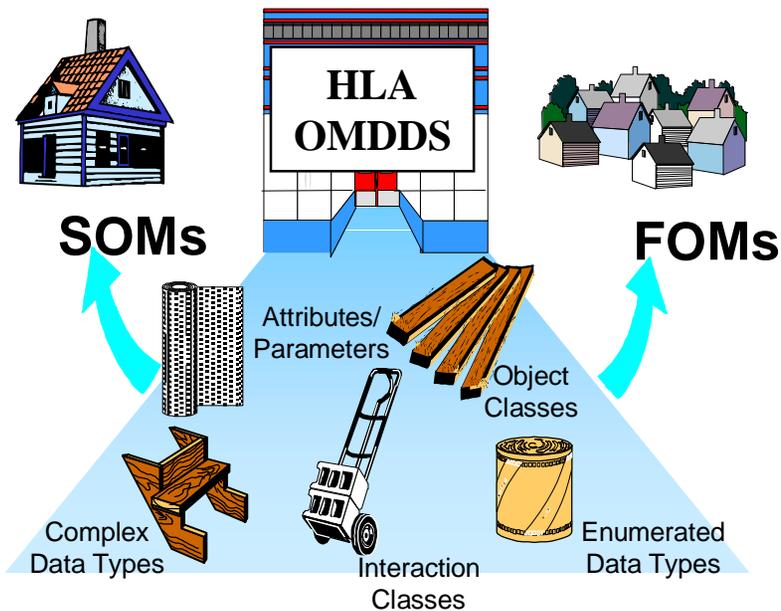
Develop the FOM

- **Choose appropriate FOM development strategy**
 - **Build from scratch**
 - **Merge SOMs from participating federates**
 - **Acquire and extend existing FOM**
 - **Acquire and modify reference FOM**
- **Determine subscription requirements and publication responsibilities at the class and attribute/parameter level**
- **Define nomenclature for all federation data elements**
 - **Utilize the Object Model Data Dictionary (OMDD)**
- **Identify (and negotiate, if necessary) structure and required characteristics of all FOM data**
- **Capture all relevant information in OMT tables**
 - **Utilize the Object Model Development Tool (OMDT)**

Sample: Moving from Conceptual Model to FOM

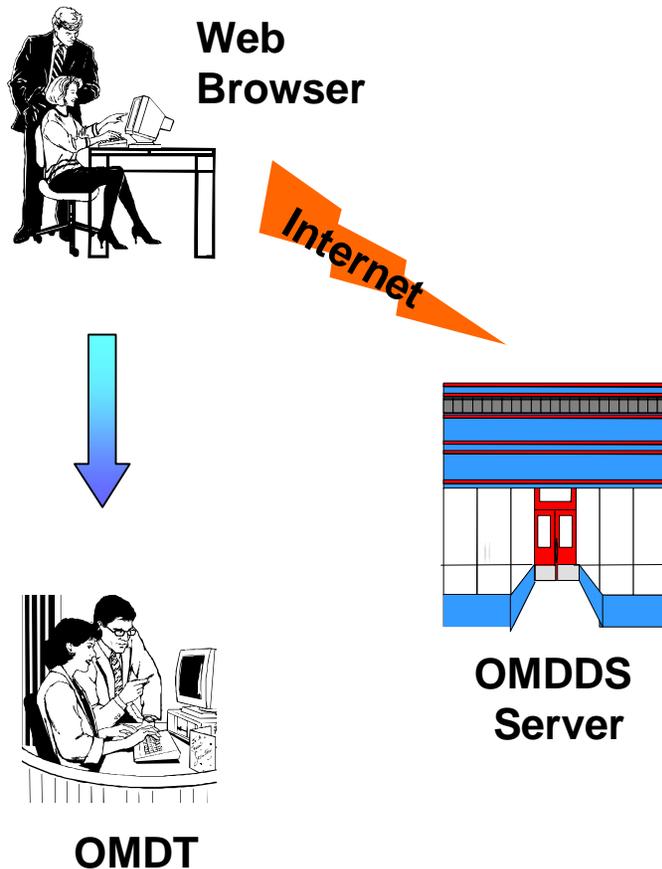
	Ship	AAA	Tank	AWACS	ADC	F/A	Bomber	Rotary Wing	Detected Target	Engaging Target	Weapon Detonate	Target Destroyed
Eagle		P	P	S	P	S	S	P			PS	PS
NSS	P	S	S			P			P	P	PS	PS
AWACS Simulator	S	S	S	P		S	S	S	S	S		S
F-15 Simulator		S				P	S	S	P	P	PS	PS
F-16 Simulator		S				P	S	S	P	P	PS	PS
B-1 Simulator		S			S	S	P			P	PS	PS
B-2 Simulator		S			S	S	P			P	PS	PS

OMDD Purpose



- Speeds development of FOMs and SOMs using an OMDT through reuse of names, lexicons, data representations, and enumerations
- Permit access to standards-based contents for the construction of FOMs and SOMs
- Relate simulations to real world systems (OMDDS contents are mapped to Defense Data Dictionary System)
- Improve the understandability and reusability of FOMs and SOMs through the use of common names, lexicons, data representations, and enumerations

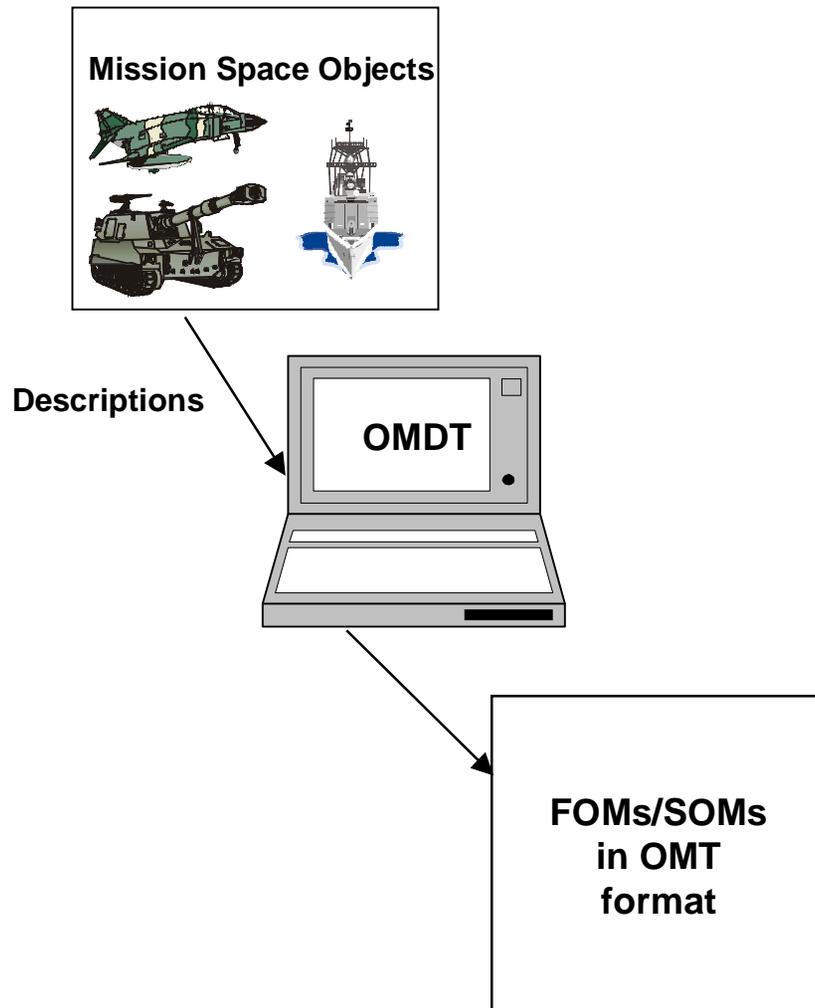
OMDD Functionality



The OMDDS is a WWW-based application which offers:

- Searching within categories of OMDD contents
- Browsing within categories of OMDD contents
- Viewing links from OMDD contents to OML and Defense Data Dictionary System contents
- Managing a user's selection of OMDD contents across multiple Web sessions
- Exporting a user's selected OMDD contents for SOM/FOM development

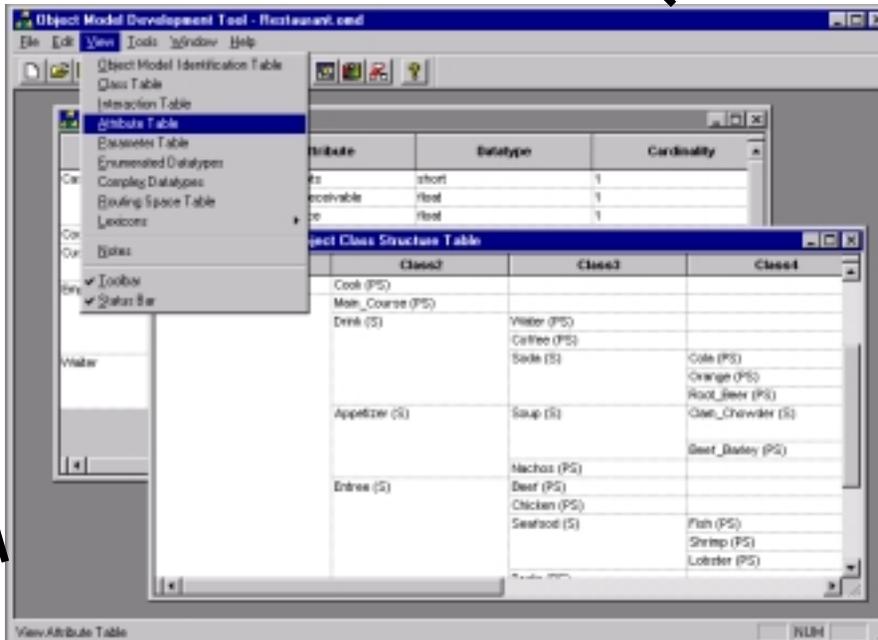
OMDT Purpose



- **Workstation based tool which supports the HLA user in**
 - Development of SOMs and FOMs
 - Generation of federation execution details for RTI initialization
 - Reusing existing HLA OMs Development status
- **C++, Sun, SGI and Win95 systems**
- **Available now and is moving into maintenance**
 - Upgrades for new specs or DIFs
- **To download, register with the HLA Software Distribution Center at <http://hla.dmso.mil>, topic “HLA Software Distribution Center”**

OMDT Functionality

HLA OMT Specification



- Desktop application
 - Windows 95 look/feel
- HLA OMT-derived User Interface
 - Enforces HLA Object Model concepts
 - Reduces learning curve
- Provides user support through automatic
 - Consistency checking across OMT tables
 - OM library interface for upload and download of FOMs and SOMs
 - FED file generation
 - Online help

Sample: OMDT Object Class Structure Table

The screenshot shows the OMDT software interface. The main window is titled "FEDEP-sample2 - Object Class Structure Table". It contains a table with four columns: Class1, Class2, Class3, and Class4. The table lists various military-related classes and their relationships.

Class1	Class2	Class3	Class4
Military_Platform (N)	Military_Vessel (N)		
	Air_Defense_Unit (N)		
	Military_Vehicle (N)		
	Ground_Base (PS)		
	Military_Aircraft (PS)	Rotary_Wing_Aircraft (PS)	
		Fixed_Wing_Aircraft (PS)	AWACS (PS)
			F-15 (PS)
			F-16 (PS)
			B-1 (PS)
			B-2 (PS)

OMDT Support @ <http://www.aegisrc.com>

3.2 Develop Federation

Develop the set of software and data products necessary to support the federation application

- Develop the FOM
- Identify shared data and algorithm requirements
- Establish federation operating agreements (as necessary)
- Implement required federate modifications
- Develop data collection strategy
- Build simulation input files or databases

Identify Shared Data and Algorithm Requirements

- **Assess the consistency of data and algorithms across the federation as needed to meet the federation requirements**
- **Identify critical inconsistencies that may influence the validity of the federation outcomes**
- **Identify databases that should be shared by all federates, such as**
 - **Terrain, oceanographic, and atmospheric**
 - **Order of battle and unit hierarchies**
 - **Weapon capabilities**
 - **Descriptions of fixed targets**
- **Identify algorithms that should be consistent or common to all federates, such as**
 - **Dead reckoning**
 - **Line-of-sight**

3.2 Develop Federation

Develop the set of software and data products necessary to support the federation application

- Develop the FOM
- Identify shared data and algorithm requirements
- **Establish federation operating agreements (as necessary)**
- Implement required federate modifications
- Develop data collection strategy
- Build simulation input files or databases

Establish Federation Operating Agreements

- Determine the appropriate **time management** policy for the federation
- Select **byte-order** for federation data exchanges
- Determine policy for **computing attrition** (e.g., firing entity or target entity)
- Establish federation **initialization procedure**; assess need for synchronization points
- Determine federation **save and restore** policy
- Determine **encoding and decoding scheme** for data elements that are lists

3.2 Develop Federation

Develop the set of software and data products necessary to support the federation application

- Develop the FOM
- Identify shared data and algorithms
- Establish federation operating agreements (as necessary)
- **Implement required federate modifications**
- Develop data collection strategy
- Develop federation initialization data

Implement Federate Modifications

- **For non-HLA federates, identify best technical approach for accomplishing the interface to the RTI**
 - Acquire and integrate COTS “adaptor” software
 - Integrate and modify COTS or GFE middleware software
 - Design and build organic RTI interface for federate
 - Consider techniques for achieving FOM agility
- **For HLA federates, extend existing RTI interface to meet FOM and federation operating agreements of this federation execution**
 - Expand or contract the publications and subscriptions
 - Modify use of time management services
 - Add support for synchronization points
- **Expand the federate to model new things and behaviors**
- **Adjust algorithms to meet the needs of the federation**

3.2 Develop Federation

Develop the set of software and data products necessary to support the federation application

- Develop the FOM
- Identify shared data and algorithms
- Establish federation operating agreements (as necessary)
- Implement required federate modifications
- **Develop data collection strategy**
- Develop federation initialization data

Develop Data Collection Strategy

- **Depending on the needs of your federation, determine data collection approach, e.g.,**
 - **Designate a logger federate (FOM updates may be needed)**
 - **Task each federate to log data**
 - **Follow a hybrid approach**
- **Identify MOEs, MOPs, and supporting calculations**
- **Detail data elements to be collected to support MOE and MOP computations**
- **Specify data recording format; establish file naming conventions**
- **Record any data interpretation agreements**
- **Draft a Data Collection Plan**

3.2 Develop Federation

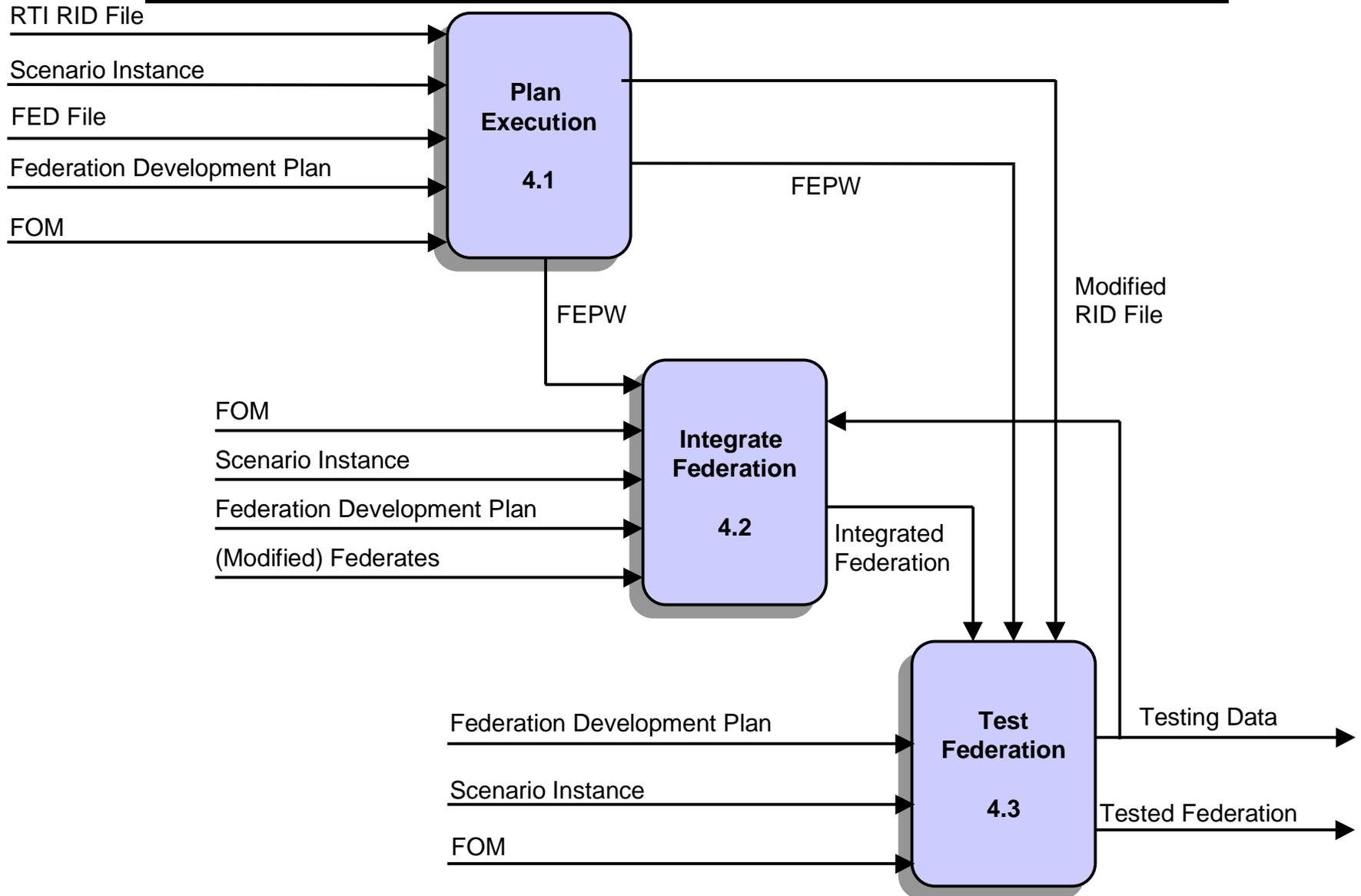
Develop the set of software and data products necessary to support the federation application

- Develop the FOM
- Identify shared data and algorithms
- Establish federation operating agreements (as necessary)
- Implement required federate modifications
- Develop data collection strategy
- **Develop federation initialization data**

Develop Federation Initialization Data

- **Build simulation input files for each federate, as necessary**
- **Required initialization data may include**
 - **Unit organization and hierarchy, including unit names, call signs, etc**
 - **Initial positions for units and assets**
 - **Initial quantities for unit holdings**
 - **Equipment**
 - **Supplies**
 - **Initial or complete set of orders to achieve the sequences of behavior needed in the scenario**

Integrate and Test Federation



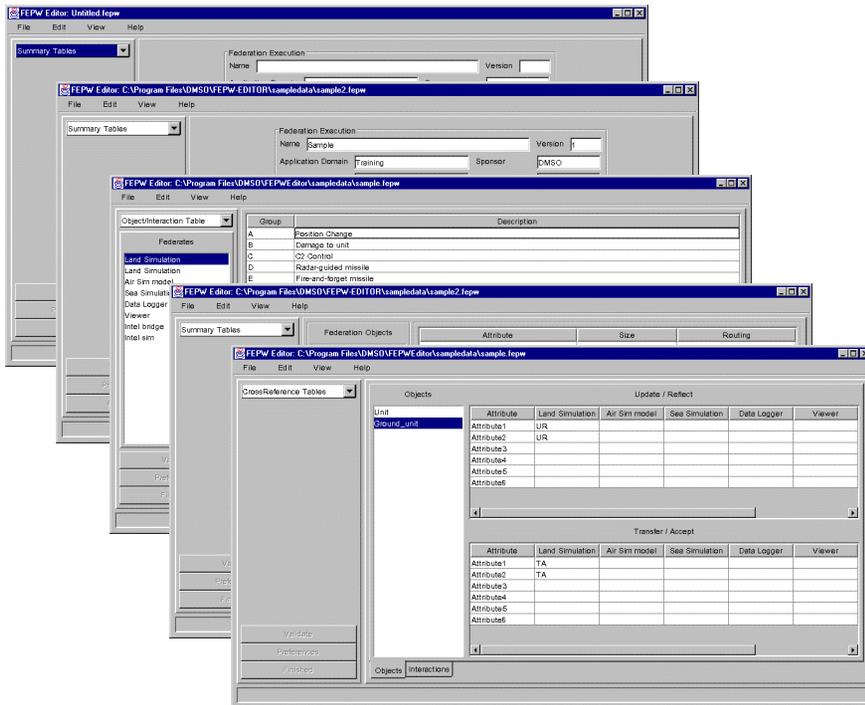
4.1 Plan Execution

Define and develop the full set of information required to support an HLA federation execution

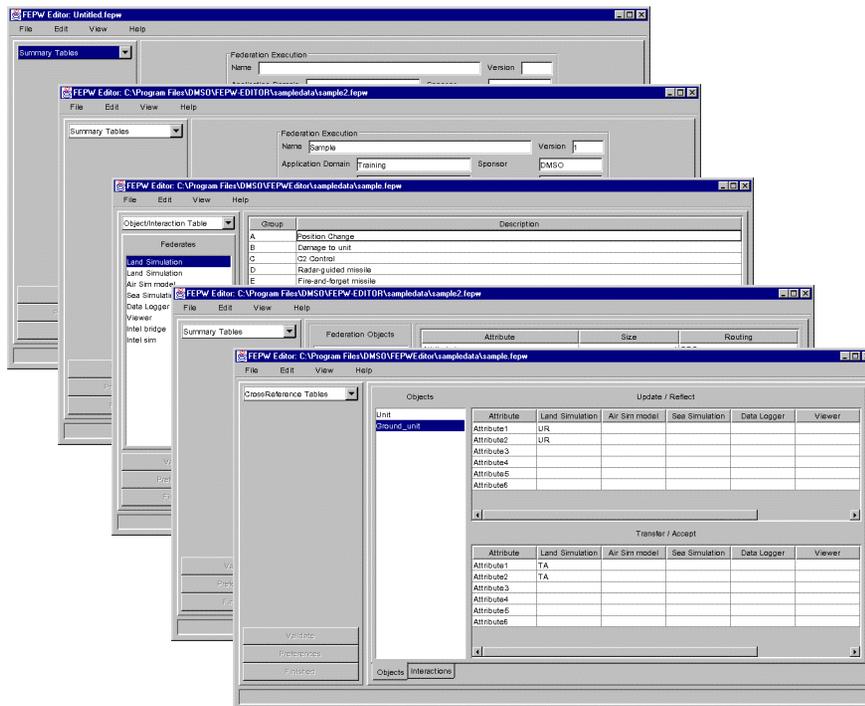
- Use the **Federation Execution Planner's Workbook (FEPW) tool** to capture key execution details
 - Equipment required
 - Expected data flow sizes and rates
 - Correlation of publishers with subscribers for all data elements
- Review facility security requirements and procedures
- Install HLA software and tools at integration and test facility
 - Computer make, model, operating system, and memory
 - Compilers and post-processing tools
 - **Federation run-time tools (e.g., FMT, FVT, DCT)**
- Refine test plans and VV&A plans
- Confirm location and schedule for integration and execution activities

FEPW Purpose

- **Federation Execution Planner's Workbook (FEPW)**
 - Assists federation developers in the planning of High Level Architecture (HLA) federation executions
 - Template for capturing the run-time federate and federation characteristics
 - Provides a mechanism for describing the federation physical structure and performance and resource requirements
- **FEPW DIF enables sharing of federation execution characteristics**



FEPW Functionality



- Federation Planning:
 - Identification of federates and capabilities
 - Identification of computing environment
 - Definition of how federation objects, attributes and interactions are distributed around the federation
- Federation testing and execution
 - Validate expected federate performance in a federation execution
 - Facilitates gathering and logging of simulation data

Sample: FEPW Federation Summary Table

FEPW Editor: C:\DMSO\FEPWEditor\sampladata\sample tut.fepw

File Help

Summary Tables
Summary Tables
Host Table
LAN Tables
RTI Services Tables
Object/Interaction Table
CrossReference Tables
Data Representation Tables

Validate

Federation Execution

Name: Sample Federation Version: 1

Application Domain: Analysis Sponsor: DMSO

Purpose: Command Center Training Analysis Date (mm/dd/yyyy): 03/04/1999

Point of Contact

Name: Mr. Fed Manager Phone: (800) 555-1212

Organization: XYZ Corp FAX:

Email: fman@xyz.com

RTI

Vendor: DMSO Version: 1.3

Concurrent Executions

Number	Name
1	Test
2	Demo
3	Prototype

Add Delete

Modified: 03/04/1999

Execution Summary Federate Summary Federation Objects Federation Interactions

Office

Microsoft

Sample: FEPW Host Table

The screenshot shows the FEPW Editor application window. The title bar reads "FEPW Editor: C:\DMSO\FEPWEditor\sampladata\sample tut.fepw". The menu bar includes "File" and "Help". On the left, a tree view shows a hierarchy of tables: "Host Table" (selected), "Summary Tables", "LAN Tables", "RTI Services Tables", "Object/Interaction Table", "CrossReference Tables", and "Data Representation Tables". The main area displays a table titled "Host Machines of Federation". The table has six columns: "Host Number", "Host Name", "LAN", "Network Address", "Geographic Location", and "Vendor/ Model". Below the table are "Add" and "Delete" buttons. A "Validate" button is located at the bottom left of the interface. The Microsoft Office taskbar is visible on the right side of the screen.

Host Number	Host Name	LAN	Network Address	Geographic Location	Vendor/ Model
1	Bubba	1	128.29.104.119	Reston, VA	Ultra 2
2	Big Guy	1	128.29.104.120	Reston, VA	Ultra 2
3	Eagle	2	128.29.102.101	San Diego, CA	Octane
4	Hawk	2	128.29.102.103	San Diego, CA	Latitude CP
5	Pigeon	2	128.29.102.105	San Diego, CA	Ultra 2
6	CanGoose	2	128.29.102.107	San Diego, CA	P233
7	Falcon	2	128.29.102.109	San Diego, CA	Octane
8	Wolverine	2	128.29.102.111	UNKNOWN	Octane
8	Wolverine	3	128.29.100.131	UNKNOWN	Octane
9	Badger	3	128.29.100.133	San Diego, CA	Ultra 2

Sample: Data Representation Table

FEPW Editor: C:\DMSO\FEPW Editor\sampladata\sample.tut.fepw

File Help

Data Representation Tables

Summary Tables
Host Table
LAN Tables
RTI Services Tables
Object/Interaction Table
CrossReference Tables
Data Representation Tables

Byte Ordering: BigEndian

Base Type Definitions

Data Types	Size (bits)	Code	
float	32	OMT	IEEE single-precision fl
double	64	OMT	IEEE double-precision 1
short	16	OMT	16-bit two's compleme
unsigned short	16	OMT	16-bit integer value (0 t
long	32	OMT	32-bit two's complemer
unsigned long	32	OMT	32-bit integer value (0 t
long long	64	OMT	64-bit two's complemer

Add Delete

Complex Types

- Type1
- Type2
- Location_2D_Struct**
- Loc_Point_List
- Ordnance_Load_Struct

Add Delete

Validate

Elements of Complex Type

Element	Type	Size
Longitude	long	
Latitude	float	

Add Delete

Office

Microsoft

4.2 & 4.3 Integrate and Test Federation

Bring federation team together in a unified operating environment to determine whether the federation executes correctly and meets the User-Sponsor requirements.

- Encourage federate teams to conduct technical tests at their facility *prior to* larger federation tests
 - Using surrogate applications that publish or subscribe to selected data (e.g., TestFederate)
- **Two categories of federation testing**
 - Inter-federate testing
 - Full federation testing

Federation Testing

- **Inter-federate Testing**
 - Plan and conduct technical tests between **select subsets** of the federation
 - Select federates that have a high correlation between their publications and subscriptions
 - Good technique for reducing complexity when the federation has more than 3 federates
 - Use very small scenarios to reduce re-start time
 - Focus on **accuracy of data exchanges**; later look at the load on the individual federates
- **Full Federation Testing**
 - Bring **all federate teams together** in single location, if possible
 - Provide a **complete development environment** (e.g., compilers, debuggers) to allow for on-site bug fixing and re-testing
 - Plan for two types of tests:
 - **Basic functionality**
 - **Stress testing** (i.e., federation execution under heavy load)

FVT Purpose

The screenshot shows the FVT interface with a table of federate conformance data. The table has columns for federates (DSE_FEDERATE, Collector, dccc, FMT) and rows for various attributes (Entity, Vehicle_Type, Appearance, Best_Effort, Resend_Interaction, Entity_Interaction, From_Identifier, Target, World_Position, Orientation). Each cell contains a status icon (U, R, T, S) and numerical values.

	DSE_FEDERATE	DSE_FEDERATE	DSE_FEDERATE	DSE_FEDERATE	Collector	dccc	FMT
Entity	# 2	# 1	# 0	# 1	# 0	# 0	#
Vehicle_Type	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0
Appearance	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0
Best_Effort	U 5 R 395	U 141 R 189	U 0 R 253	U 120 R 61	U 0 R 34		
Resend_Interaction							
Entity_Interaction							
From_Identifier	T 0 S 5	T 3 S 0	T 0 S 3	T 0 S 3	T 0 S 1	T 0 S 1	T 0 S 1
Target	T 0 S 5	T 3 S 0	T 0 S 3	T 0 S 3	T 0 S 1	T 0 S 1	T 0 S 1
World_Position	T 0 S 5	T 3 S 0	T 0 S 3	T 0 S 3	T 0 S 1	T 0 S 1	T 0 S 1
Orientation	T 0 S 5	T 3 S 0	T 0 S 3	T 0 S 3	T 0 S 1	T 0 S 1	T 0 S 1

- The Federation Verification Tool (FVT) assists federation developers in the Federation Integration and Test phase of the FEDEP
- Supports federation developers in verifying that each federate is meeting their data exchange responsibilities
 - “Federation Requirements” are defined by the FOM, FED, and FEPW (object/interaction table)
- FVT can be used to
 - Assess consistency across federation plans (FOM, FED, FEPW)
 - Assess individual federate conformance to federation requirements
 - Assess federation conformance to federation requirements during integration

FVT Functionality

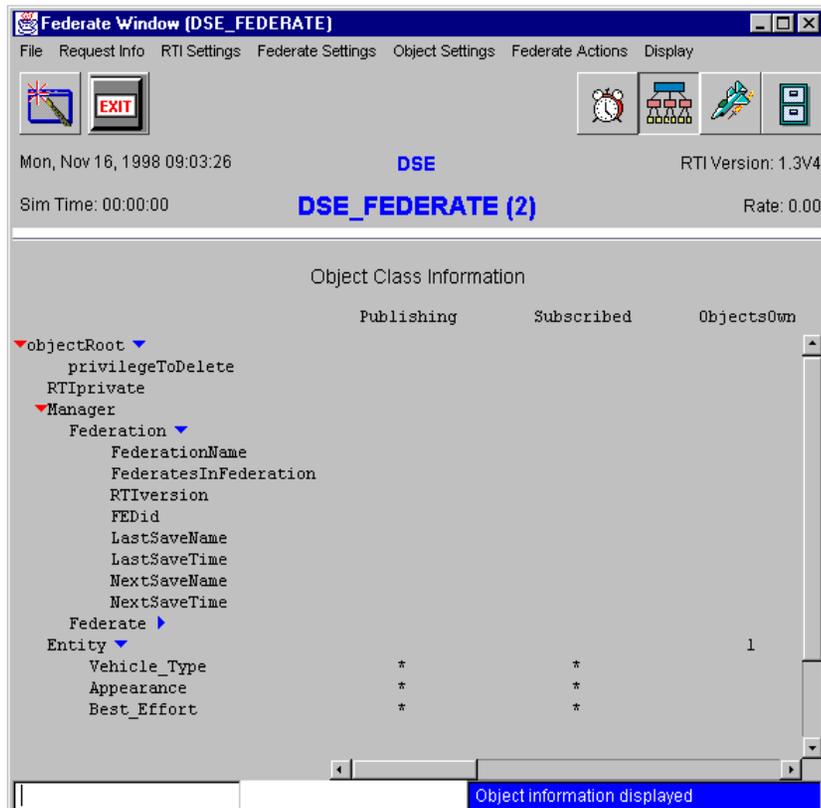
The screenshot shows the FVT software interface with a data table. The table has columns for federates (DSE_FEDERATE, Collector, dccc, FMT) and rows for various interaction types (Entity, Resend_Interaction, Entity_Interaction). The data is as follows:

	DSE_FEDERATE	DSE_FEDERATE	DSE_FEDERATE	DSE_FEDERATE	Collector	dccc	FMT
Entity	# 2	# 1	# 0	# 1	# 0	# 0	#
Vehicle_Type	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0
Appearance	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0	U 0 R 0
Best_Effort	U 5 R 395	U 141 R 189	U 0 R 253	U 120 R 61	U 0 R 34	U 0 R 0	U 0 R 0
Resend_Interaction							
Entity_Interaction							
From_Identifier	U 0 S 5	U 3 S 0	U 0 S 3	U 0 S 3	U 0 S 1	U 0 S 1	U 0 S 1
Target	U 0 S 5	U 3 S 0	U 0 S 3	U 0 S 3	U 0 S 1	U 0 S 1	U 0 S 1
World_Position	U 0 S 5	U 3 S 0	U 0 S 3	U 0 S 3	U 0 S 1	U 0 S 1	U 0 S 1
Orientation	U 0 S 5	U 3 S 0	U 0 S 3	U 0 S 3	U 0 S 1	U 0 S 1	U 0 S 1

At the bottom of the window, it shows 'Joined Rhino.fed Done.' and 'Not Editable Discreps: 1'.

- How does FVT operate?
 - FVT joins the federation as a federate, collects data from federates through standard RTI interface, and then checks federate behavior against the federation requirements.
- What does it do?
 - The current focus of FVT is to verify that each federate updates and reflects the objects and sends and receives the interactions it is responsible for in the federation.

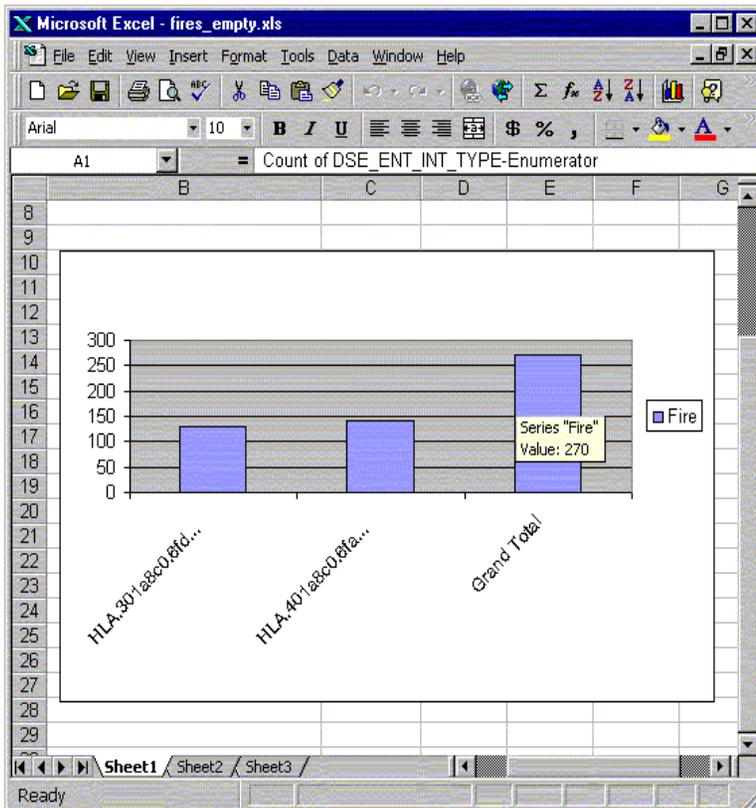
FMT Purpose



- **Displays MOM data**
 - Who is in the exercise?
 - What are they subscribing and publishing?
 - How time is advancing?
- **Provides MOM-based control**
 - Force subscription and publishing
 - Initiate RTI logging
- **Acts on behalf of a federate**
 - Respond to synchronization, save or restore requests

DCT Purpose

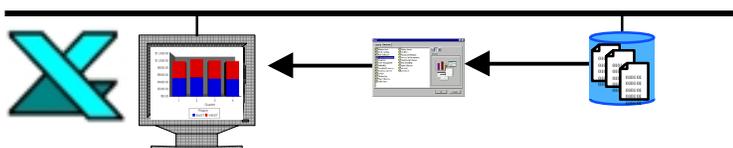
- **Purpose of Data Collection Tool**
 - Analysis of tactical/strategic operations
 - Analysis of simulation model behavior (V&V)
 - Measures Of Effectiveness, Performance, and Success
 - Maximize use of Commercial Off The Shelf (COTS) data analysis tools (DBMS, Spreadsheet applications)
- **Capture and Storage of Federation Data**
 - Persistent history and recording of Federation Object Model (FOM) data for a federation execution
 - Characterization of federation state
 - Temporal evolution of objects
 - Events/interactions exchanged between federates



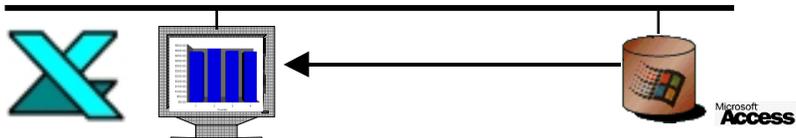
DCT Functionality

Excel

Data Collection Files



Excel



Collection Files

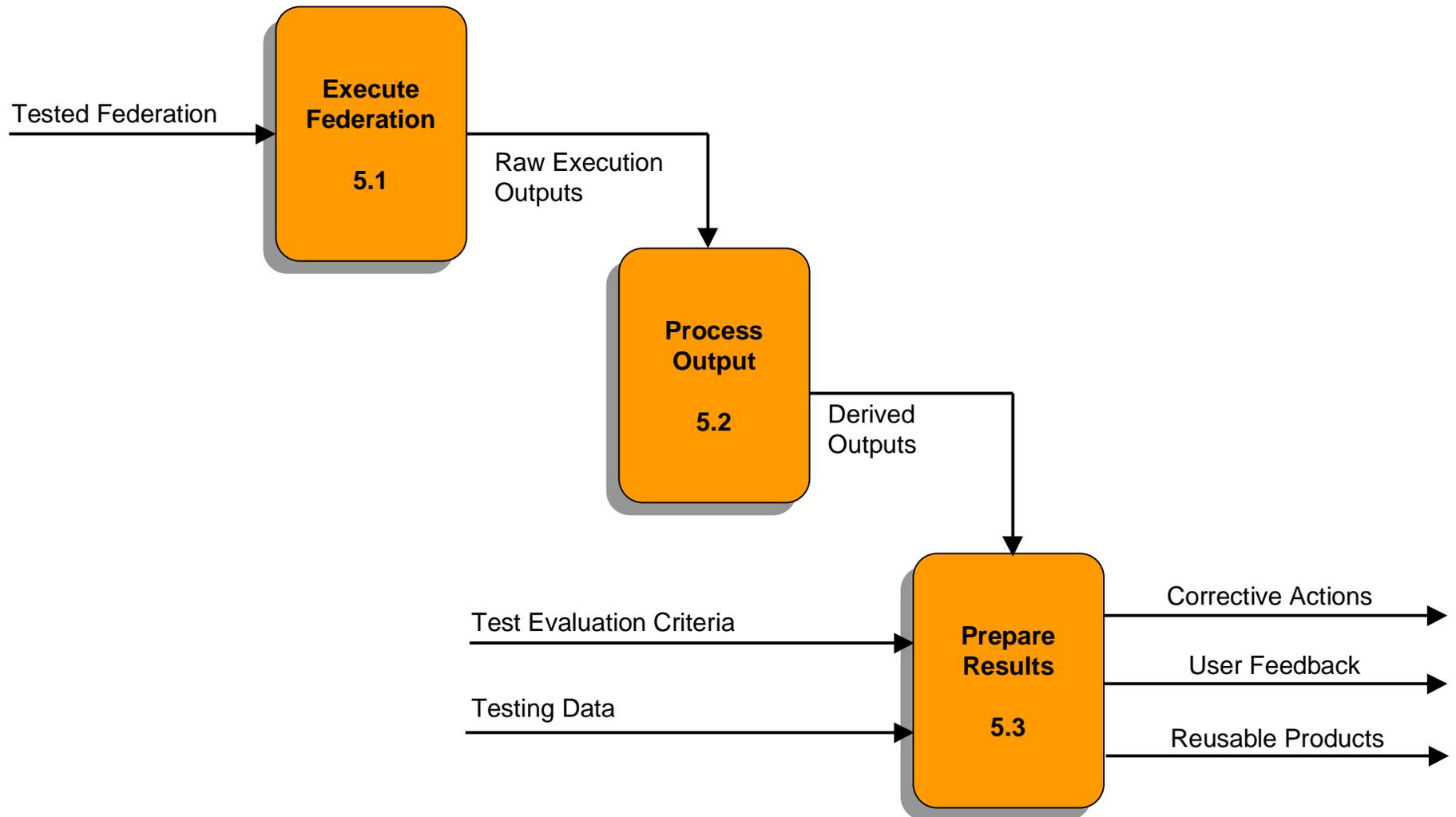
- *MS Excel import*

Database Support

- *MS Excel import*
- *MS Access Report Wizard*

- **Federation Independence**
 - FOM based Database Schema Generation
 - FOM- aware Data Collector
- **Data Collection Control Interface**
 - Java Collection Control Federate
 - Launch, Start/Stop, Pause/Resume and change collector subscriptions
 - Subscription according to type and/or DDM region
- **Supports COTS Database and Analysis Tools**
 - Microsoft Access and Excel
- **Support for all 1.3 Interface Specification compliant RTIs**
 - Respond to synchronization, save or restore requests

Execute Federation and Prepare Results

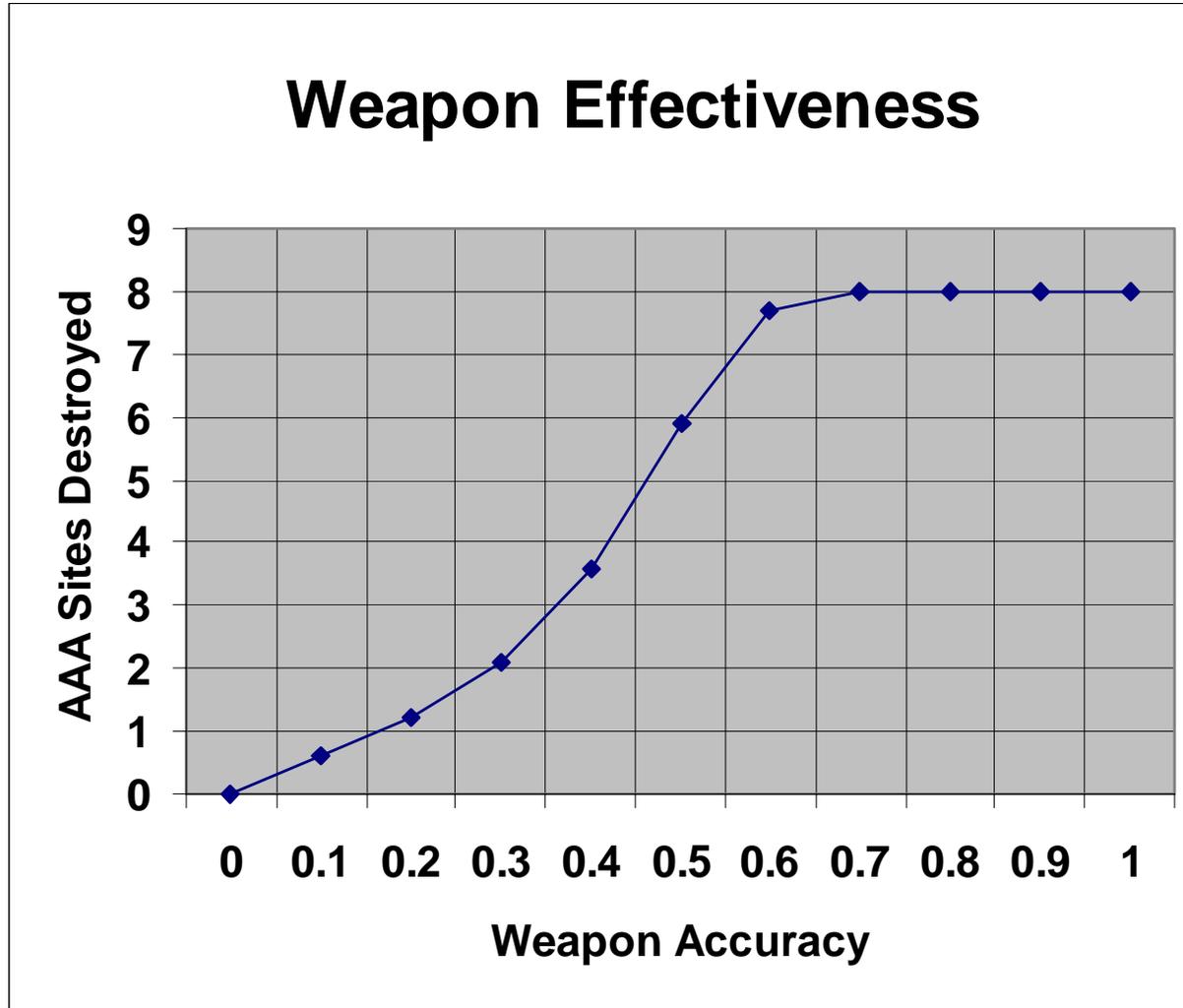


5.1 & 5.2 Execute Federation & Process Output

Conduct and monitor one or more federation executions to support the required event (e.g., training exercise, test event, or analysis runs).

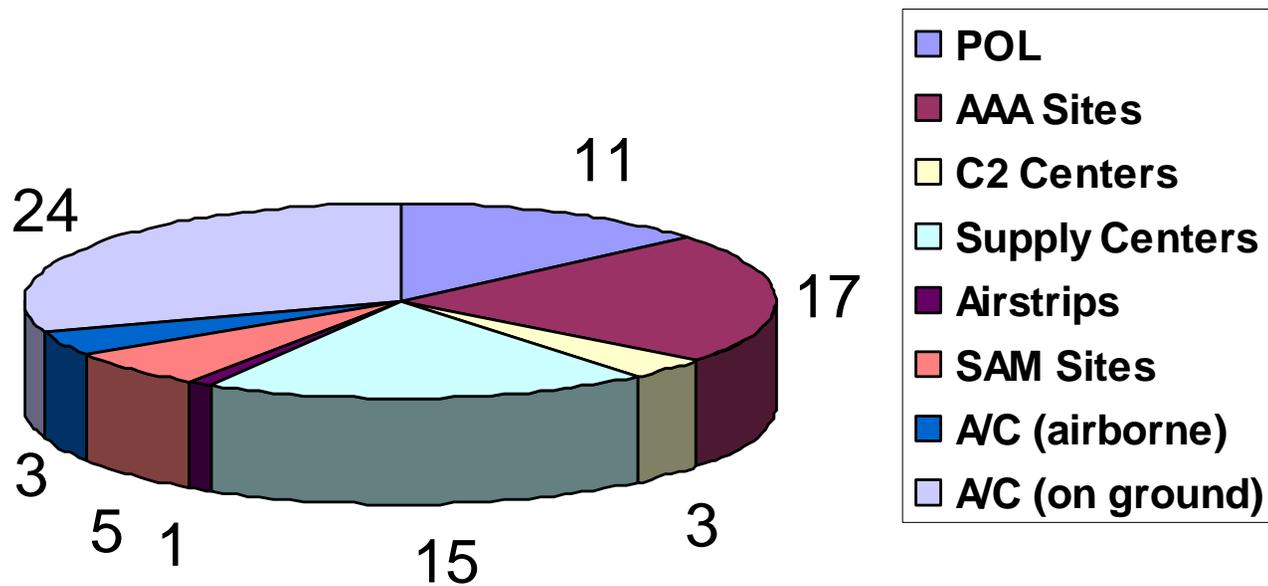
- **Initiate federation execution in controlled environment**
- **Collect and reduce data**

Sample: Processed Output



Sample: Processed Output

Targets Destroyed

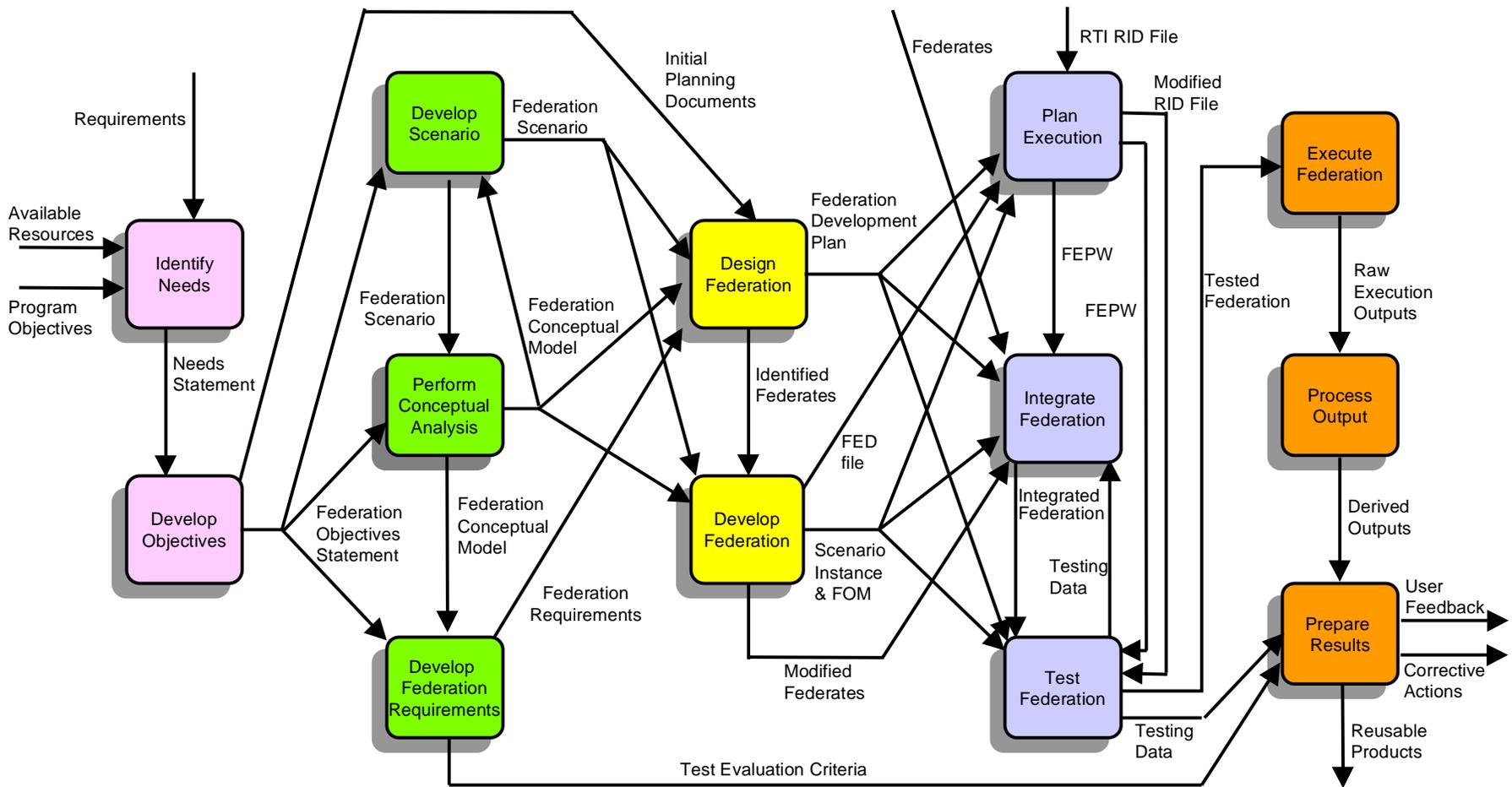


5.3 Prepare Results

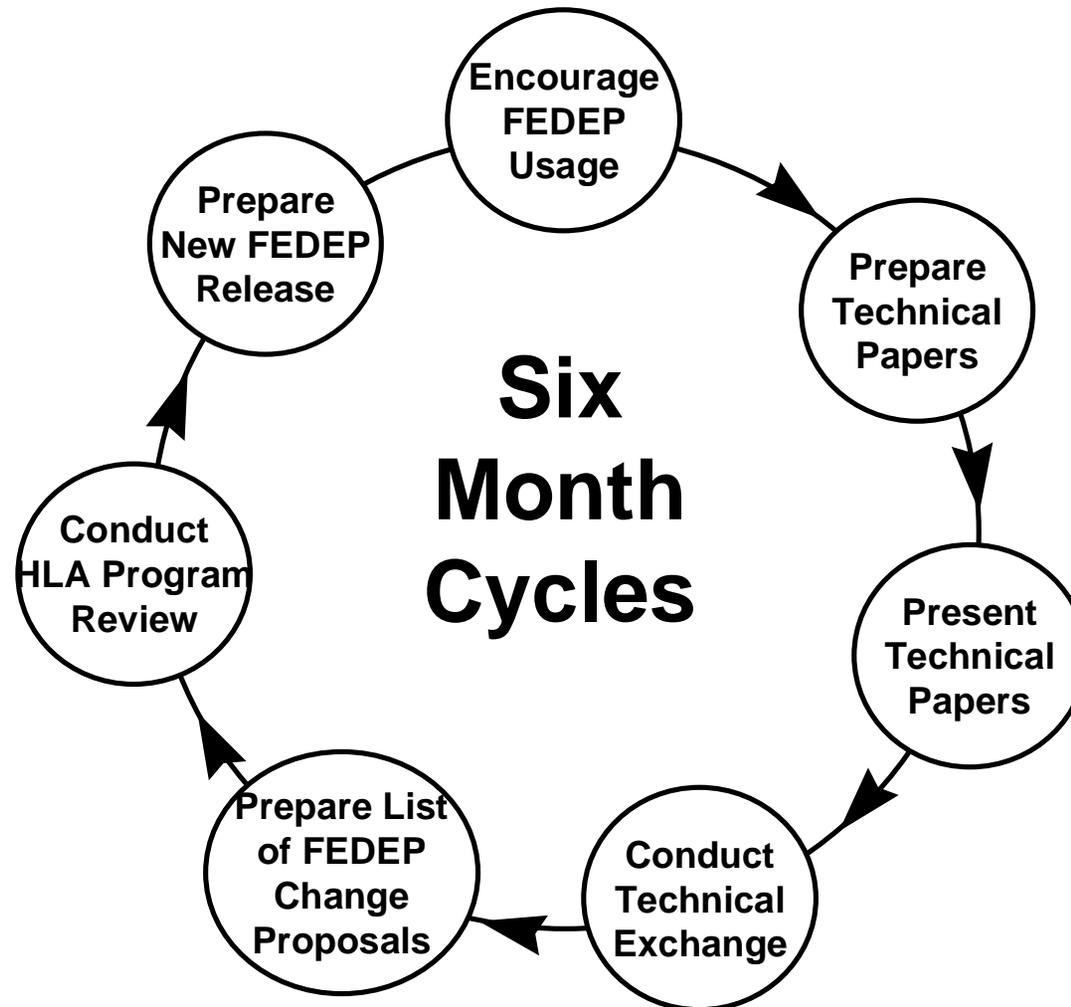
Determine success of federation execution, prepare final report, and archive reusable federation products

- **Determine whether all federation requirements have been achieved**
 - If so, prepare and deliver final report to User/Sponsor and disband the federation team
 - If not, identify and perform additional federation activities as required
- **Identify and archive reusable federation products as appropriate**
 - FOM and SOM(s)
 - FEPW
 - Conceptual model
 - Planning documents

Develop and Execute HLA Federation



FEDEP Development Concept of Operations



HLA Tools Philosophy and Architecture

- **HLA is an architecture, not software -- however to facilitate cost-effective implementation of HLA,**
 - **DMSO is developing an initial suite of HLA support software (including Runtime Infrastructure software and a variety of integrated tools)**
- **DMSO facilitates open development of supported tools**
 - **Open access to the specification**
 - **Creation of formal data interchange formats (DIFs) where necessary**
 - **Industry participation in DIF development**
- **Developed to:**
 - **Help identify requirements for tools to automate specific portions of the FEDEP**
 - **Established as an open framework for the development of HLA tools**
 - **Promotes an interoperable set of tools**
 - **Creates an extensible tool set that will evolve as requirements grow**

DMSO Sponsored HLA Supporting Software /Tools

- **Object Model Tools - Available now**
 - Object Model Development Tool (OMDT)
 - Object Model Library (OML)
 - Object Model Data Dictionary (OMDD)
- **Federation Execution Planning and Verification - Available now**
 - Federation Execution Planners Workbook (FEPW)
 - RTI Initialization Data (RID) Editor
- **Runtime Tools**
 - Runtime Infrastructure (RTI)
 - 1.3- Available now
 - 1.3 NextGeneration - In Beta
 - Federation Verification Tool (FVT) - Available now
 - Federation Management Tool (FMT) - Available now
 - Data Collection Tool (DCT) - Available now

HLA Website

The image shows a screenshot of the HLA Website as viewed in the Netscape browser. The browser window title is "HLA Homepage - Netscape". The address bar shows the URL "http://hla.dmsomil". The browser interface includes a menu bar (File, Edit, View, Go, Communicator, Help), a toolbar with icons for Back, Forward, Reload, Home, Search, Netscape, Print, Security, and Stop, and a bookmarks bar. The website content is displayed in a yellow-themed layout. On the left is a vertical navigation menu with blue circular icons and text links. The main content area features a blue header for "HLA Helpers" with sub-links for "New Users", "DoD Policy", and "Developers". Below this is a large heading "High Level Architecture (HLA)" followed by two paragraphs of text. The first paragraph defines HLA as a general purpose architecture for simulation reuse and interoperability. The second paragraph describes the development of HLA by the US Department of Defense under the leadership of the Defense Modeling and Simulation Office (DMSO). The browser's status bar at the bottom shows the current page URL: "http://hla.dmsomil/cgi-bin/hla-cgi/tools/bboard.pl".

HLA Helpers
[New Users](#) [DoD Policy](#) [Developers](#)

High Level Architecture (HLA)

The High Level Architecture (HLA) is a general purpose architecture for simulation reuse and interoperability. HLA is based on the premise that no simulation can satisfy all uses and users. An individual simulation or set of simulations developed for one purpose can be applied to another application under the HLA concept of the federation: a composable set of interacting simulations. The intent of the HLA is a structure that will support reuse of capabilities available in different simulations, ultimately reducing the cost and time required to create a synthetic environment for a new purpose, and the possibility of distributed collaborative development of complex simulation applications.

The HLA was developed by the US Department of Defense under the leadership of the [Defense Modeling and Simulation Office \(DMSO\)](#) to support reuse and interoperability across the large numbers of different types of simulations developed and maintained by that organization. The HLA itself is generally applicable across the wide range of simulation applications and the HLA specifications and supporting software is freely available to the broader simulation development and user community. HLA was adopted as the Facility for Distributed Simulation Systems 1.0 by the Object Management Group (OMG) in November 1998.

Search:

http://hla.dmsomil/cgi-bin/hla-cgi/tools/bboard.pl

HLA Tools Bulletin Board



HLA Tools Bulletin Board

Commercially Developed Tools

Government Sponsored Tools

- A service to inform the community of tools and services available to support HLA implementations
- Products are listed alphabetically by company name
- Solicitation for product postings is made through the Bulletin Board
- Bulletin board is available through HLA Home Page to DoD, Academia and Industry

Summary

- **The FEDEP defines a generic, tailorable framework for the development of distributed systems**
- **A set of software tools have been developed under DMSO sponsorship to support HLA-specific FEDEP activities**
- **Commercial vendors are actively developing new software tools to augment (or supplant) existing HLA tools**
- **The FEDEP (and associated tool architecture) will continue to evolve based on inputs from the HLA user community**
 - **SIW PROC Forum provides an open forum for sharing federation development experiences**



HLA Federation Development and Execution Process (FEDEP) & Supporting Tools

Bob Lutz and Chris Turrell

Southwest Regional Training

27 May 1999

Back-Up Slides

FEDEP History

- **FEDEP V1.0 (Sep 96)**
 - Based on profederation experience and feedback
 - OMT Working Group provided forum for sharing federation development practices/approaches
- **FEDEP V1.1 (Nov 97)**
 - Complete existing process description
 - Review uniformity of process description
 - Implement “minor” changes to FEDEP diagram
- **FEDEP V1.2 (July 98)**
 - Include roles/products in process description
 - Federation reuse
- **FEDEP V1.3 (Dec 98)**
 - Improved graphical representation

OMDD Export

OMDDS - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: <http://s3.arl.utexas.edu/omdds/code/index.cfm> What's Related

Instant Message WebMail Contact People Yellow Pages Download Channels

Component To Browse: [Object Class](#) [Interaction Class](#) [Generic Element](#) [Complex Data Type](#) [Enumerated Data Type](#)

Name Begins with Letter: A-E F-L M-P Q-S T-Z

Selection	Component Name
<input type="checkbox"/>	Aggregate Entity
<input checked="" type="checkbox"/>	Air Defense Unit
<input checked="" type="checkbox"/>	Air Mission
<input checked="" type="checkbox"/>	Aircraft
<input type="checkbox"/>	Unknown File Type
<input type="checkbox"/>	

EXPORT AREA FOR CTURRELL - Netscape

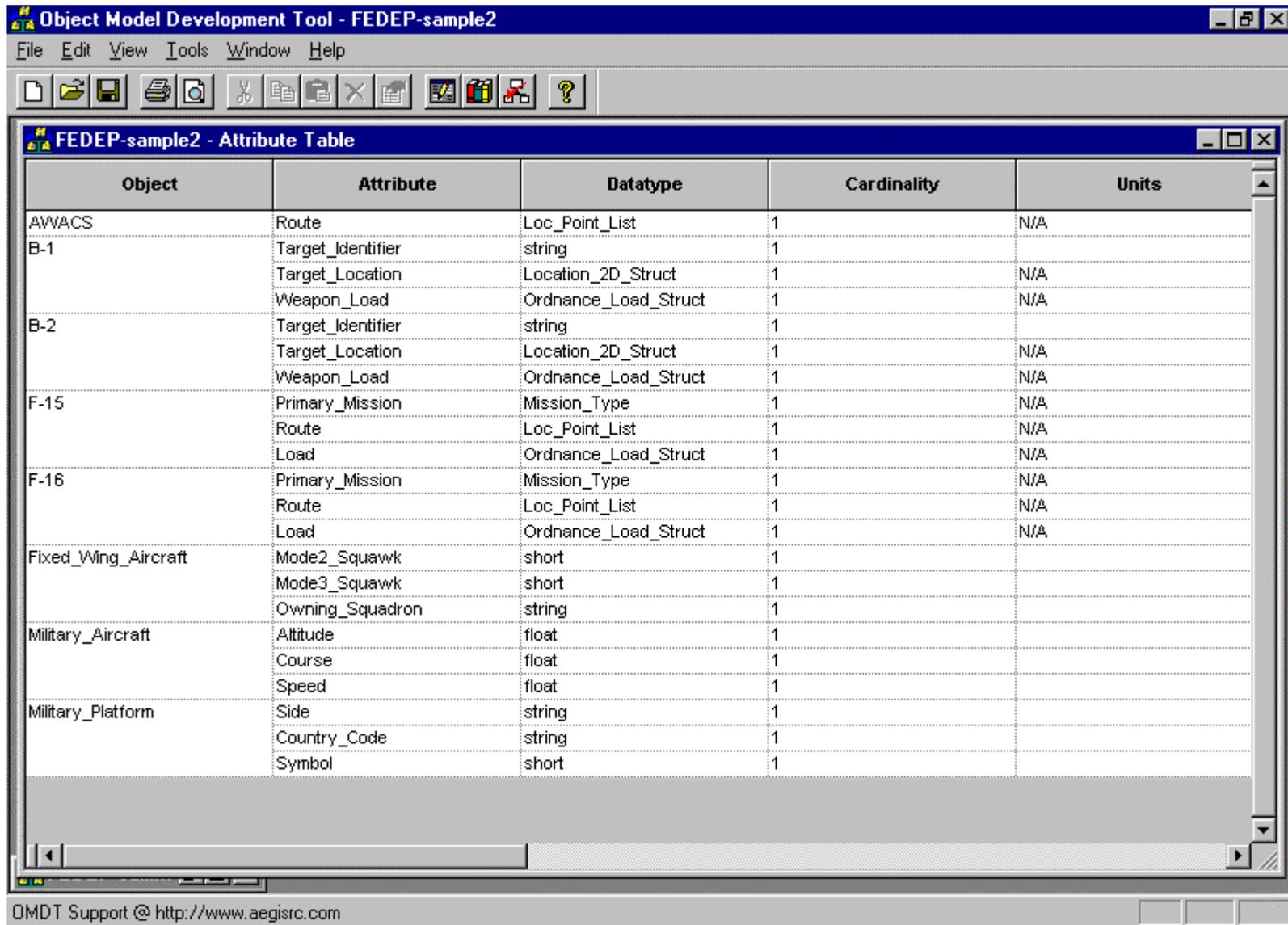
EXPORT AREA FOR CTURRELL

Export	Delete	Type	Name
<input checked="" type="checkbox"/>	<input type="checkbox"/>	CLASS	<i>Air_Mission</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	CLASS	<i>Air_Defense_Unit</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	CLASS	<i>Aircraft</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	CLASS	<i>Amphibious_Assault_Vehicle</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	CLASS	<i>Close_Air_Support_Request</i>

NOTE: Items shown in italics are OMDD components related to the items you selected for inclusion in the export area.

Document: Done

Sample: OMDT Attribute Table



Object Model Development Tool - FEDEP-sample2

File Edit View Tools Window Help

FEDEP-sample2 - Attribute Table

Object	Attribute	Datatype	Cardinality	Units
AWACS	Route	Loc_Point_List	1	N/A
B-1	Target_Identifier	string	1	
	Target_Location	Location_2D_Struct	1	N/A
	Weapon_Load	Ordnance_Load_Struct	1	N/A
B-2	Target_Identifier	string	1	
	Target_Location	Location_2D_Struct	1	N/A
	Weapon_Load	Ordnance_Load_Struct	1	N/A
F-15	Primary_Mission	Mission_Type	1	N/A
	Route	Loc_Point_List	1	N/A
	Load	Ordnance_Load_Struct	1	N/A
F-16	Primary_Mission	Mission_Type	1	N/A
	Route	Loc_Point_List	1	N/A
	Load	Ordnance_Load_Struct	1	N/A
Fixed_Wing_Aircraft	Mode2_Squawk	short	1	
	Mode3_Squawk	short	1	
	Owning_Squadron	string	1	
Military_Aircraft	Altitude	float	1	
	Course	float	1	
	Speed	float	1	
Military_Platform	Side	string	1	
	Country_Code	string	1	
	Symbol	short	1	

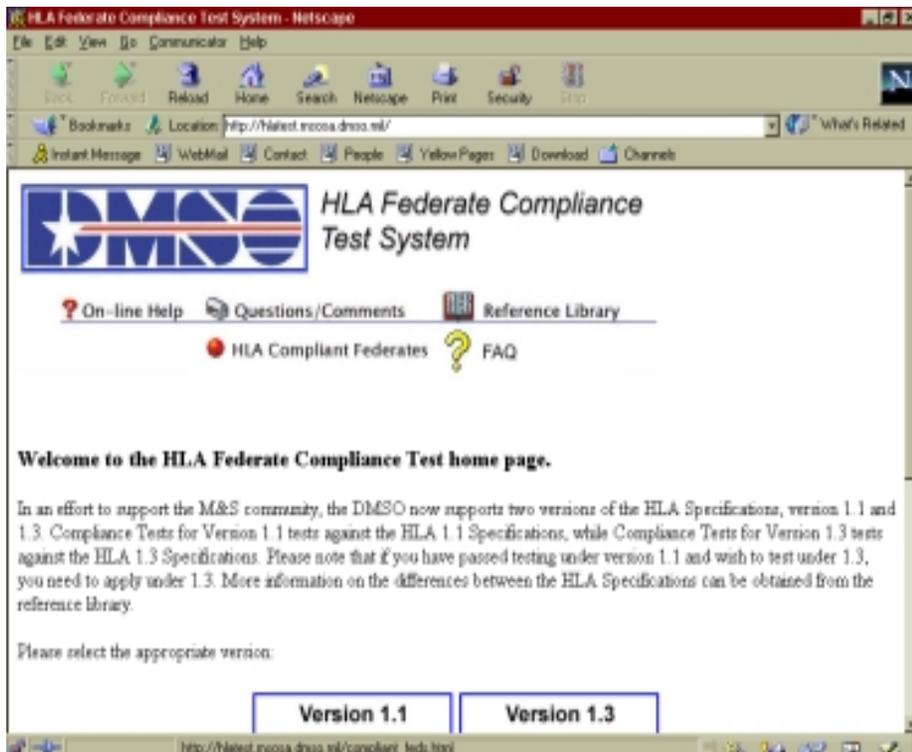
OMDT Support @ <http://www.aegisrc.com>

4.2 & 4.3 Integrate and Test Federation

Bring federation team together in a unified operating environment to determine whether the federation executes correctly and meets the sponsor requirements.

- **Three categories of federation testing**
 - Inter-federate testing
 - Full federation testing
 - **Compliance testing**
- **Encourage federate teams to conduct technical tests at their facility prior to larger federation tests**
 - Using surrogate applications that publish or subscribe to selected data (e.g., TestFederate)

Federate Compliance Testing Philosophy



- Federates are tested to the HLA Specifications (IF, OMT, Rules)
- Test process is straight forward:
 - Federate Under Test (FUT) submits a description of its capabilities via a SOM and Conformance Statement (CS) to the Certification Agent (CA)
 - Federate demonstrates it can use the specifications correctly through a set of tests
 - CA verifies the federate conforms to the specifications by analyzing test results
- Test process is supported by automated tools to reduce time and cost associated with testing
 - FVT joins the federation as a federate, collects data from