Advancing the Federation Development and Execution Process (FEDEP) for Simulation Based Acquisition (SBA)

Katherine L. Morse, Ph.D
FCS MSO Chief Software Engineer
morsek@saic.com

Paul Lowe
IS&T SPDT Deputy Chief Software Engineer
paul.n.lowe@boeing.com

Introduction

- Simulation Based Acquisition (SBA) programs are confronted with two paradigms that compete for program level focus and resources.
    - The first paradigm requires modeling & simulation (M&S) teams to develop simulation environments for testing in order to "sell-off" the SBA program. This line of thought invariably demands immediate attention toward developing unique simulation based event configurations for supporting intermediate tests, experiments and capability demonstrations.
    - The second paradigm requires the same M&S teams to concurrently develop a robust collection of simulation environment tools for SBA contract delivery.
- A proposed tailoring of the Federation Development and Execution Process (FEDEP) is set forth, capturing the maturation of requirements within a Spiral Lifecycle Model (SLM), allowing these two paradigms to co-exist over the lifecycle of a development program, making the SBA process more effective.

- Goals
    - Identify alignment of FEDEP with program systems engineering processes
    - Fill gaps to create robust federation engineering process
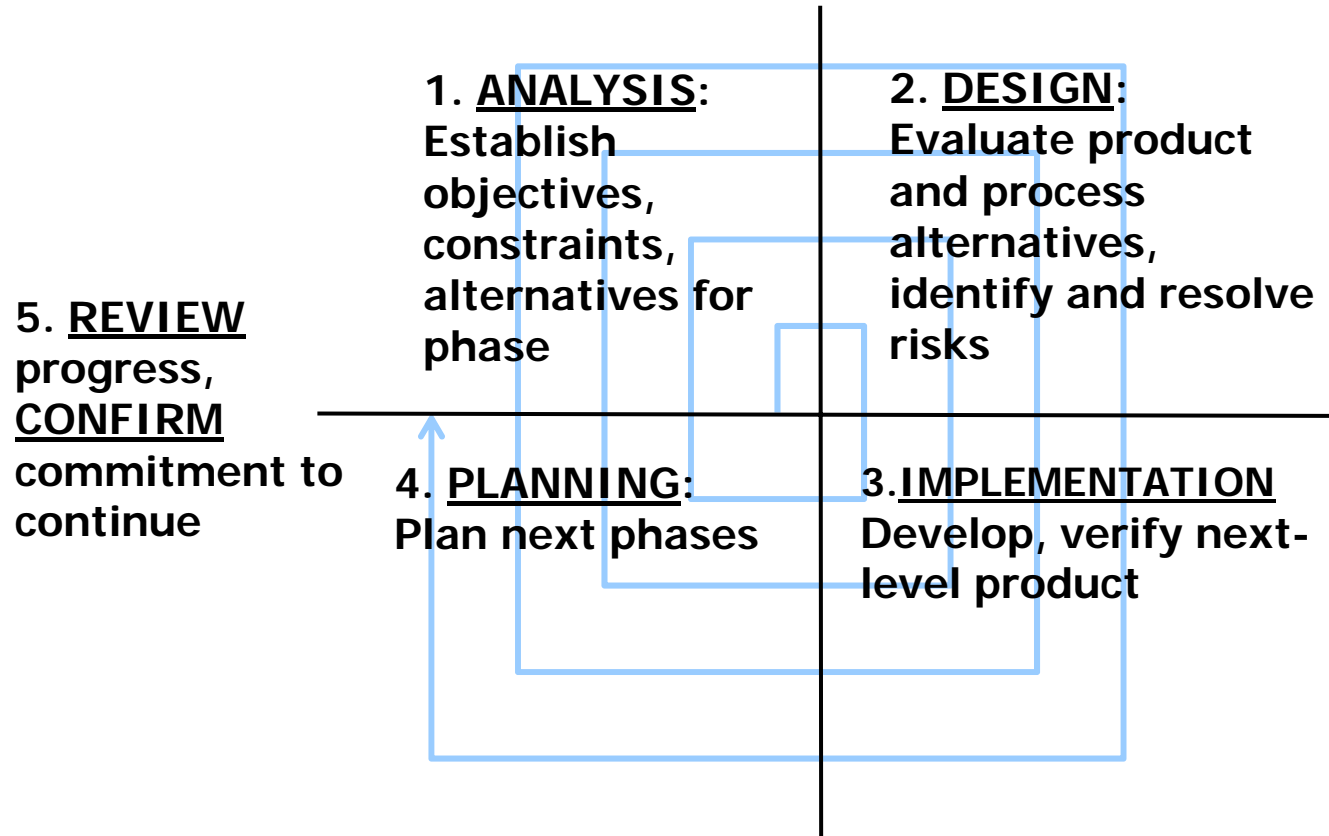   - Share lessons learned with other large SBA programs
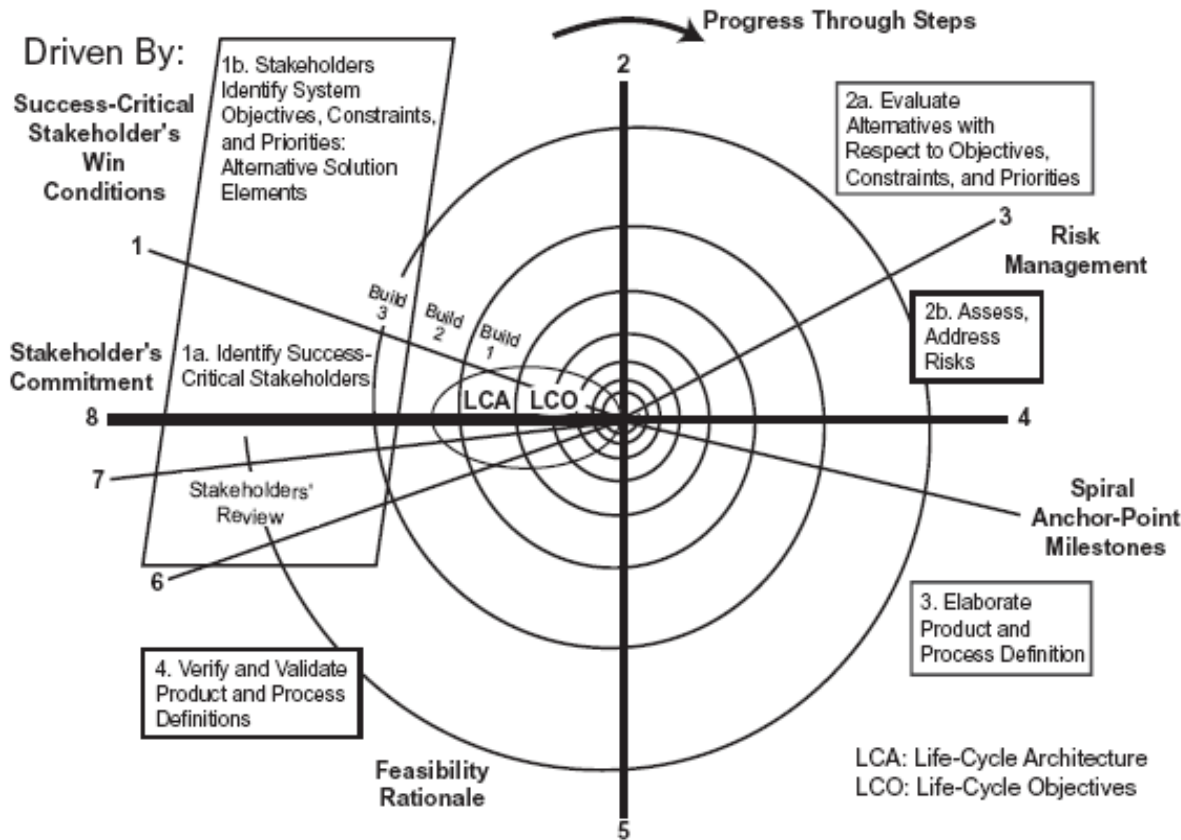- Process
    - Bottom-up mapping of FEDEP artifacts to existing program artifacts
        - Look for gaps in federation artifacts to improve FCS process
    - Top-down mapping of planning conferences through anchor points to FEDEP steps/activities/tasks
        - Identify how information and artifacts must flow from the customer through the systems engineering processes into the final federation artifacts
        - While anchor points are not used by many programs, the concept and process for performing this mapping are broadly applicable
    - Identify opportunities for reuse of artifacts from IP to IP, minimizing rework
        - Applicable to any large, iterative SBA program
    - Make recommendations for additions/modifications to program processes and artifacts
        - While some of these lessons learned are specific to FCS, many are broadly applicable

Processes to Be Aligned

- Planning conferences
- System of systems (SoS) spiral lifecycle model (SLM) anchor points (APs)
- Federation Development and Execution Process (FEDEP)

Army M&S Specific Guidance for Planning Conferences

IPC	MPC	FPC

• **Define the scenario: Terrain, ORBAT and Campaign Plan**
• **Define command and control (Exercise Control Cell)**
• **Define manning for exercise players, response cells, control and support**
• **Define C4I requirements**
• **Develop the training plan**
• **Establish database milestones and begin build**
• **Determine real-world logistical support**
• **Draft Memorandum of Agreement (MOA) or Pro Forma**
• **Schedule supporting training events:**
  • **Site survey. (pre-MPC)**
  • **Database builds (including 'Good Idea Cut-off Time')**
  • **Scenario Development (pre-MPC) and scripting (post-MPC)**
  • **'Train-the-trainer' for the model and ABCS (post-MPC)**
• **Joint and outside agency participation**
	• **Present coordinated Exercise Plan to the exercise director and senior reps from key organizations:**
	  • **training objectives**
	  • **exercise objectives**
	  • **organizations involved and roles/responsibilities**
	  • **exercise directive (specified tasks and coordinating instructions)**
	  • **planning timeline, tasks required and tracking status**
	  • **scenario progress, 'Road-to-War', inject requirements**
	  • **technical plan, database requirements, simulation workarounds,**
	  • **budget and contract requirements**
	  • **logistical support**
	  • **cell structure and manning requirements**
	  • **communications plan**
	  • **O/C, AAR and collection plan**
	• **Identify cell OICs**
		• **Present final coordinated plan**
		• **Publish FRAGO if required**
		  • **Review MOA milestones, update status**
		  • **Resolve outstanding issues**
		  • **Review training objectives**
		  • **Review manning**
		  • **Review conduct of the exercise**
		  • **Publish Exercise Control Group**
		  • **Review exercise budget versus changes to projected costs**
		  • **Cell OICs present and provide backbriefs**
		  • **Review training requirements (O/C, unit, operator) prior to exercise**
		  • **OPFOR review**

Spiral Lifecycle Model (SLM)

1. ANALYSIS:
Establish objectives, constraints, alternatives for phase

2. DESIGN:
Evaluate product and process alternatives, identify and resolve risks

5. REVIEW progress, **CONFIRM** commitment to continue

4. PLANNING:
Plan next phases

3.IMPLEMENTATION
Develop, verify next-level product

SLM Tailoring Process

Driven By:

Progress Through Steps

Success-Critical Stakeholder's Win Conditions

1b. Stakeholders Identify System Objectives, Constraints, and Priorities: Alternative Solution Elements

2a. Evaluate Alternatives with Respect to Objectives, Constraints, and Priorities

Risk Management

Stakeholder's Commitment

1a. Identify Success-Critical Stakeholders

2b. Assess, Address Risks

Spiral Anchor-Point Milestones

Stakeholders' Review

3. Elaborate Product and Process Definition

4. Verify and Validate Product and Process Definitions

Feasibility Rationale

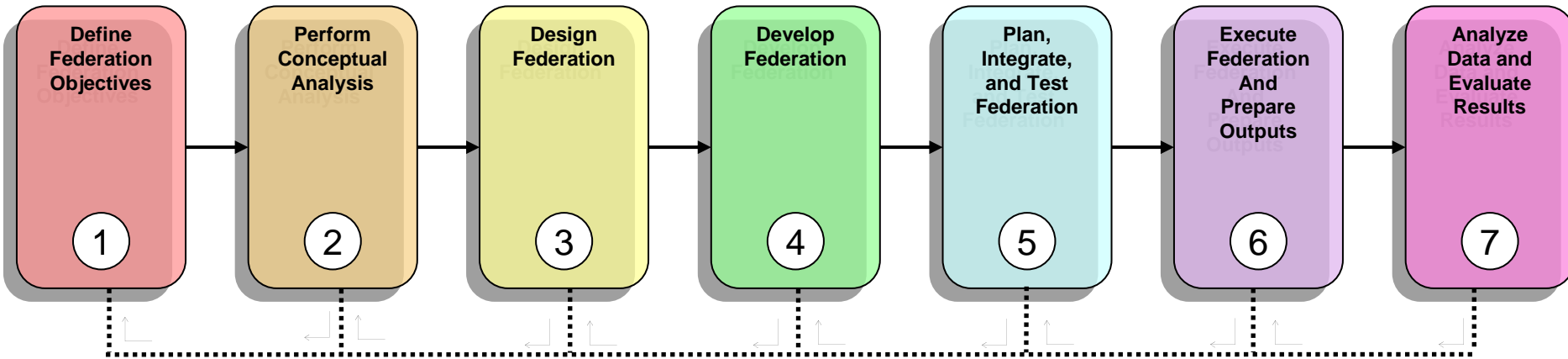LCA: Life-Cycle Architecture
LCO: Life-Cycle Objectives

At a SoS level, the SLM might be tailored for a program to include the following reviews which would occur during each phase of a SLM based program:

- Definition Anchor Point (**DAP**)
- Planning Anchor Point (**PAP**)
- Readiness Anchor Point (**RAP**)
- Assessment Anchor Point (**AAP**)

(Boehm 2004)

Anchor Point Definitions

- Definition Anchor Point (DAP)
  - Guidance for each phase focusing more and more on end-state product
- Planning Anchor Point (PAP)
  - High-level review of the plans, architecture, and risks for the entire spiral development
  - Detailed plans of the specific phase in question
  - Risks and integration challenges identified in the DAP
- Readiness Anchor Point (RAP)
  - Most significant checkpoint associated with each build
    - Represents commitment across all levels of a program that the software build can be successfully implemented within the build budget and schedule using the documented architectures and designs
    - Risk mitigation plans exist for all potential shortfalls
- Assessment Anchor Point (AAP)
 - Identify process improvements that can be made in subsequent phases

Define Federation Objectives	Perform Conceptual Analysis	Design Federation	Develop Federation	Plan, Integrate, and Test Federation	Execute Federation And Prepare Outputs	Analyze Data and Evaluate Results
1	2	3	4	5	6	7

Rather than dictating a "one-size-fits all" solution for all users, the FEDEP provides a common overarching process framework into which lower-level domain-specific management and/or engineering methodologies can be easily integrated.

Approved for Public Release, Distribution Unlimited, PM FCS 22 Oct 2007, case 07-271.
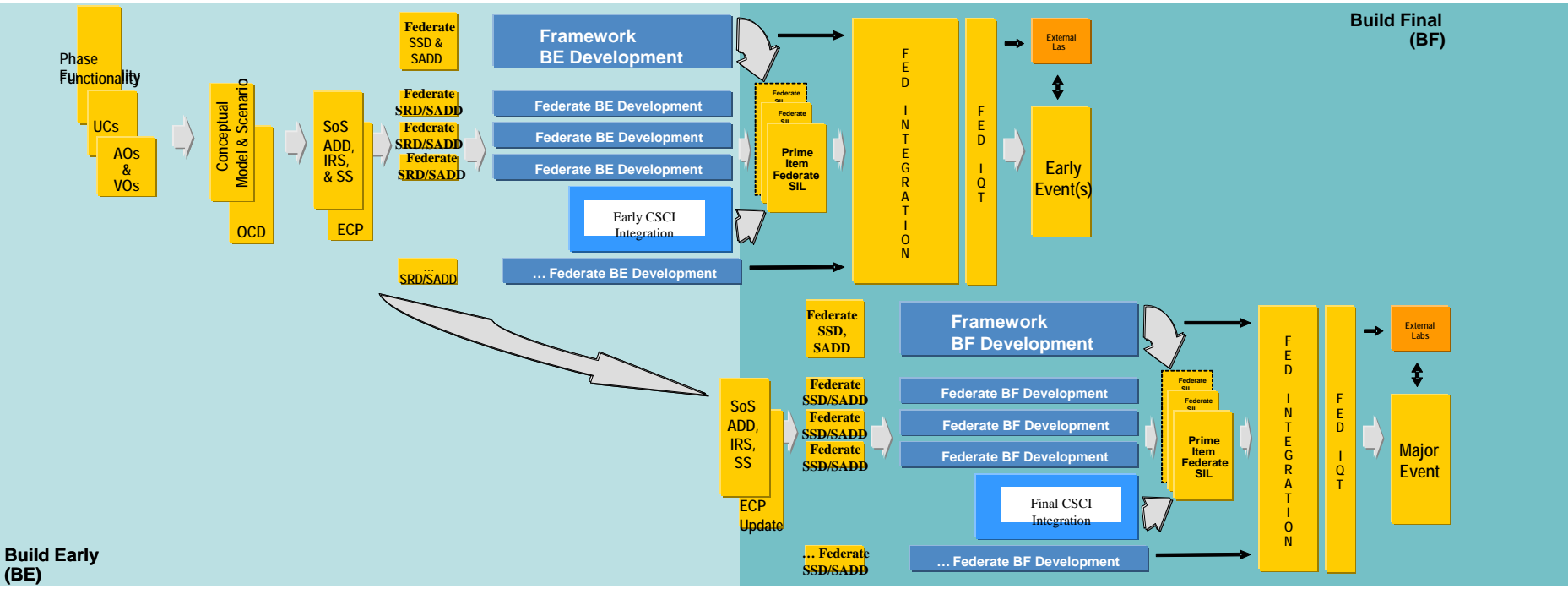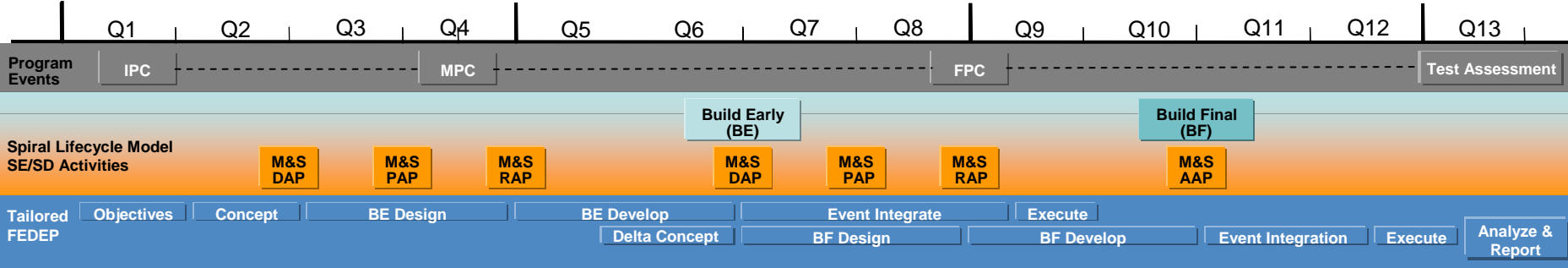
- One of the key goals of this paper was to identify how the FEDEP might be used in large SBA programs such as FCS.
  - Key to this is understanding how systems engineering processes interact.
- We mapped the planning conferences through the anchor points to the FEDEP recommended inputs, tasks, and outcomes.
  - Anchor points represent gating conditions or controls on the FEDEP, not technical inputs.
    - To complete this mapping, we defined four new anchor point credentials focused on reviewing planning conference outputs as inputs to the FEDEP.
- Where do the planning conferences and anchor points impact the FEDEP?

Approved for Public Release, Distribution Unlimited, PM FCS 22 Oct 2007, case 07-271.

11

- 1.1 - Identify User/Sponsor Needs
  - Any known constraints which may affect how the federation is developed and executed (e.g., due dates, security requirements)
- 1.2 - Develop Objectives
  - Assess federation feasibility and risk.
  - Define and document an initial federation development and execution plan.
  - Develop initial planning documents, including: Federation development and execution plan showing an approximate schedule and major milestones.
- 2.1 - Develop scenario(s)
  - Federation scenario(s)
- 2.2 - Develop federation conceptual model
  - Federation conceptual model
- 2.3 - Develop federation requirements
  - Federation requirements
  - Federation test criteria

- 3.2 - Prepare federation design
  - Federation design
  - Federation architecture (including supporting infrastructure design)
  - Implied requirements for federate modifications and/or development of new federates
- 3.3 - Prepare plan
  - Integration plan
- 4.1 - Develop FOM
  - FOM
  - FED/FDD
- 5.3 - Test Federation
  - Tested (and if necessary, accredited) federation
- 7.2 - Evaluate and feedback results
  - Lessons learned
  - Final report
  - Reusable federation products

These all represent points where the federation touches the SoS, but where M&S specific guidance is needed.
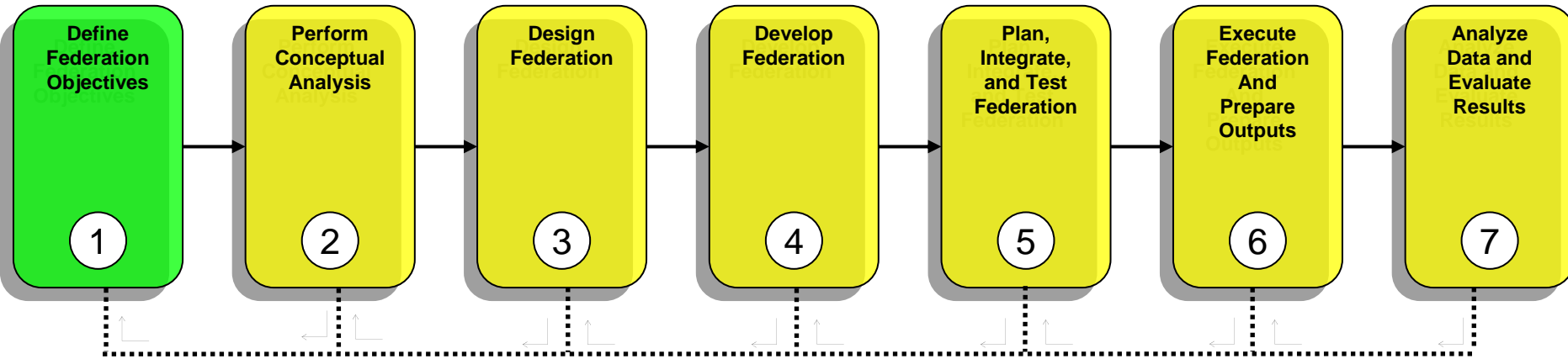
- Recognize the federation as a first-class object.
  - The tested federation, the output of FEDEP activity 5.3, must be a deliverable itself .
- Program level guidance needs to be translated into executable, M&S specific-guidance.
  - Most of our testing plans focus on testing the SoS using the federation, but there is very little information on testing the federation.
- Record both the decisions that are made and the processes by which decisions are made.
  - You may have to revisit those decisions in a later iteration, e.g. selection of existing federates to meet the requirements of a particular iteration. Knowing the criteria for the decision can expedite reevaluation.
- Federation requirements must be readily identifiable as a subset of SoS requirements.
  - Additionally, there should be continuous requirements management because delays in delivery of operational software may require filling in those items with M&S, but that too takes time.

- Recognize where M&S is the same and where it's different from your operational software.
  - For example, non-operational M&S may not need as rigorous testing as operational software, but the same CM and documentation standards probably apply.
    - However, consider the global implications of relaxing standards for M&S because it may have broader implications, e.g. reducing the level of testing for M&S may reduce your ability to fully test operational software that depends on M&S.
  - Embedded M&S is operational software and should be treated as such.
- M&S can solve your representation shortfalls, **not** your interface ones.
  - M&S has to have interfaces too, preferably the same ones used for operational systems so operational code can be dropped in readily later.

14

Approved for Public Release, Distribution Unlimited, PM FCS 22 Oct 2007, case 07-271.

Estimating Rework Across IPs

- Reviewed each FEDEP recommended task and estimated level of rework in later IPs based on the assumption that preceding IPs were successfully executed and assigned the following values:
  - 1 (green) - little to no rework in subsequent iterations. Either program level documents remain essentially unchanged or a program process is already in place that minimizes effort.
  - 2 (yellow) - some rework, but not a substantial engineering effort. Additional or updated entity or scenario representations necessitate engineering effort that ripples throughout federate and federation engineering.
  - 3 (red) - significant rework. The actual federate and federation engineering required to implement new functionality that represents the core of the iteration intent.
- Rolled up statistics to FEDEP tasks and further into FEDEP steps

	FEDEP Steps	Description	Iteration Rework Metric
1	Define Federation Objectives	Define and document a set of needs that are to be addressed through the development and execution of an HLA federation and to transform these needs into a more detailed list of specific federation objectives	1
1.1	Identify User/Sponsor Needs	Develop a clear understanding of the problem to be addressed by the federation	
1.1.2	Recommended tasks		1
1.1.2.1		Identify program objectives that motivate federation development	1
1.1.2.2		Identify available resources and known development and execution constraints	2
1.1.2.3		Document this information in a needs statement	1
1.2	Develop Objectives		
1.2.2	Recommended Tasks		
1.2.2.1		Analyze the needs statement.	1
1.2.2.2		Assess federation feasibility and risk.	2
1.2.2.3		Define and document a prioritized set of federation objectives, consistent with the needs statement.	1
1.2.2.4		Meet with the federation sponsor to review the federation objectives, and reconcile any differences.	1
1.2.2.5		Define and document an initial federation development and execution plan.	2
1.2.2.6		Identify potential tools to support the initial plan.	1

15

High Level Rework Analysis Results*

Define Federation Objectives (1)	→	Perform Conceptual Analysis (2)	→	Design Federation (3)	→	Develop Federation (4)	→	Plan, Integrate, and Test Federation (5)	→	Execute Federation And Prepare Outputs (6)	→	Analyze Data and Evaluate Results (7)

- Assume that the preceding iterations were correctly executed
- Step 1 represents the program level decisions, most of which were made in the first iteration.
- Most of the hard work (indicated in red in the spreadsheet) occurs in steps 2 – 4 because that's where the federate and federation engineering really happens.
- Most of the rework in steps 5 – 7 is the ripple effect of changes to federate and federation re-engineering, although the effort is declining again in step 7 due to presumed reuse of data analysis methods. Program processes mitigate the amount of rework, but changes still have to be documented and tested.

*Detail by FEDEP step in backup slides;
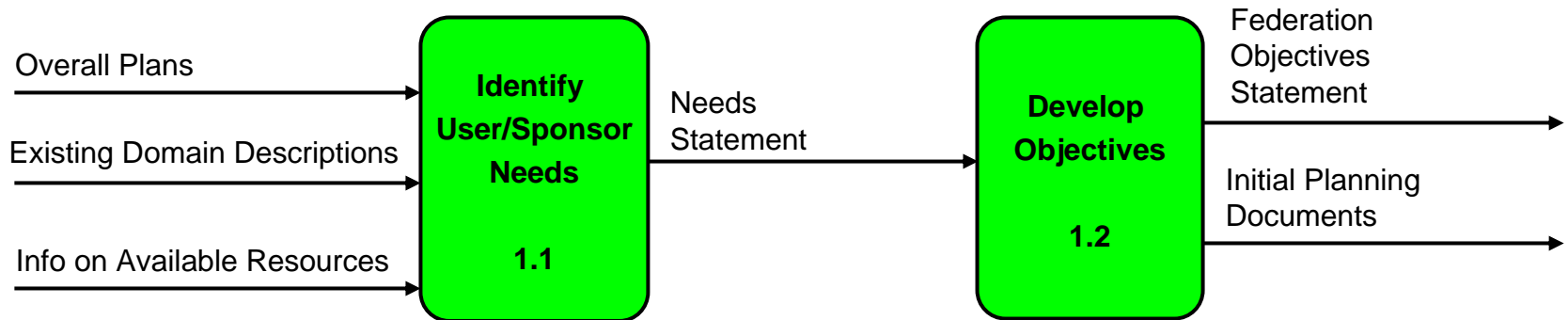detail by task in 07F-SIW-083

Final Thoughts

- Program-level processes focus on cost, schedule, and risk.

- From the SoS perspective, the federation (and the FEDEP) are test tools.

- The FEDEP is focused on the technical aspects of producing a federation.

- For FCS, we're introducing criteria for gating program-level processes down to the FEDEP to align these different focuses.

- For the broader SBA community, we're providing input to the SISO update of the FEDEP to introduce the hooks necessary for a large, iterative, SBA program.

References

- Katherine L. Morse, Paul Lowe, Gary Fuller, and Mark Riecken, "Enhancing the Simulation Based Acquisition Process Using a Spiral Lifecycle Model Based Federation Development and Execution Process," Proceedings of the 2007 Spring Simulation Interoperability Workshop, Norfolk, VA, March 25 – 30, 2007.

- Katherine L. Morse and Paul Lowe, "Advancing the Federation Development and Execution Process (FEDEP) for Simulation Based Acquisition (SBA)," Proceedings of the 2007 Fall Simulation Interoperability Workshop, Orlando, FL, September 16 – 21, 2007.
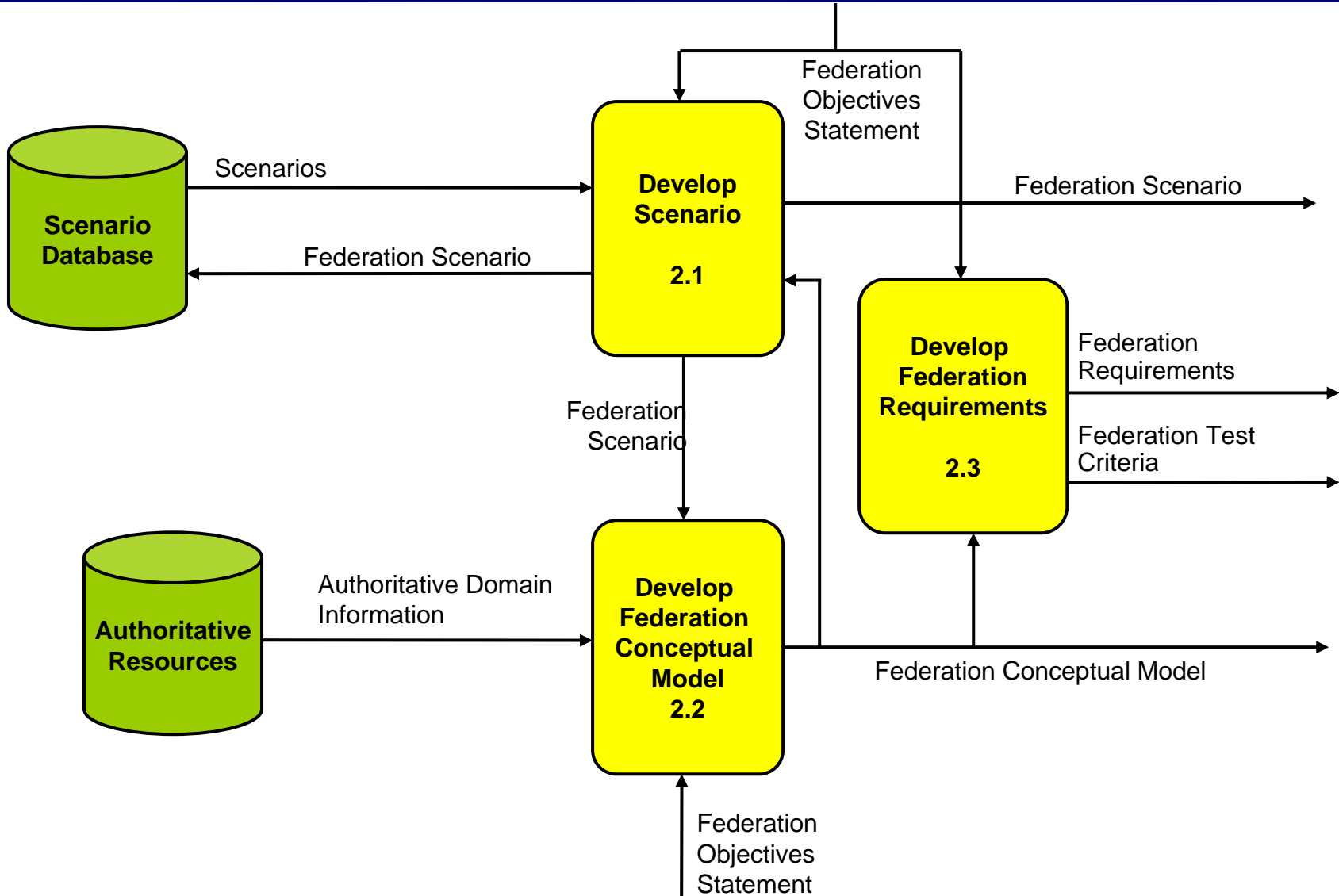
Backups

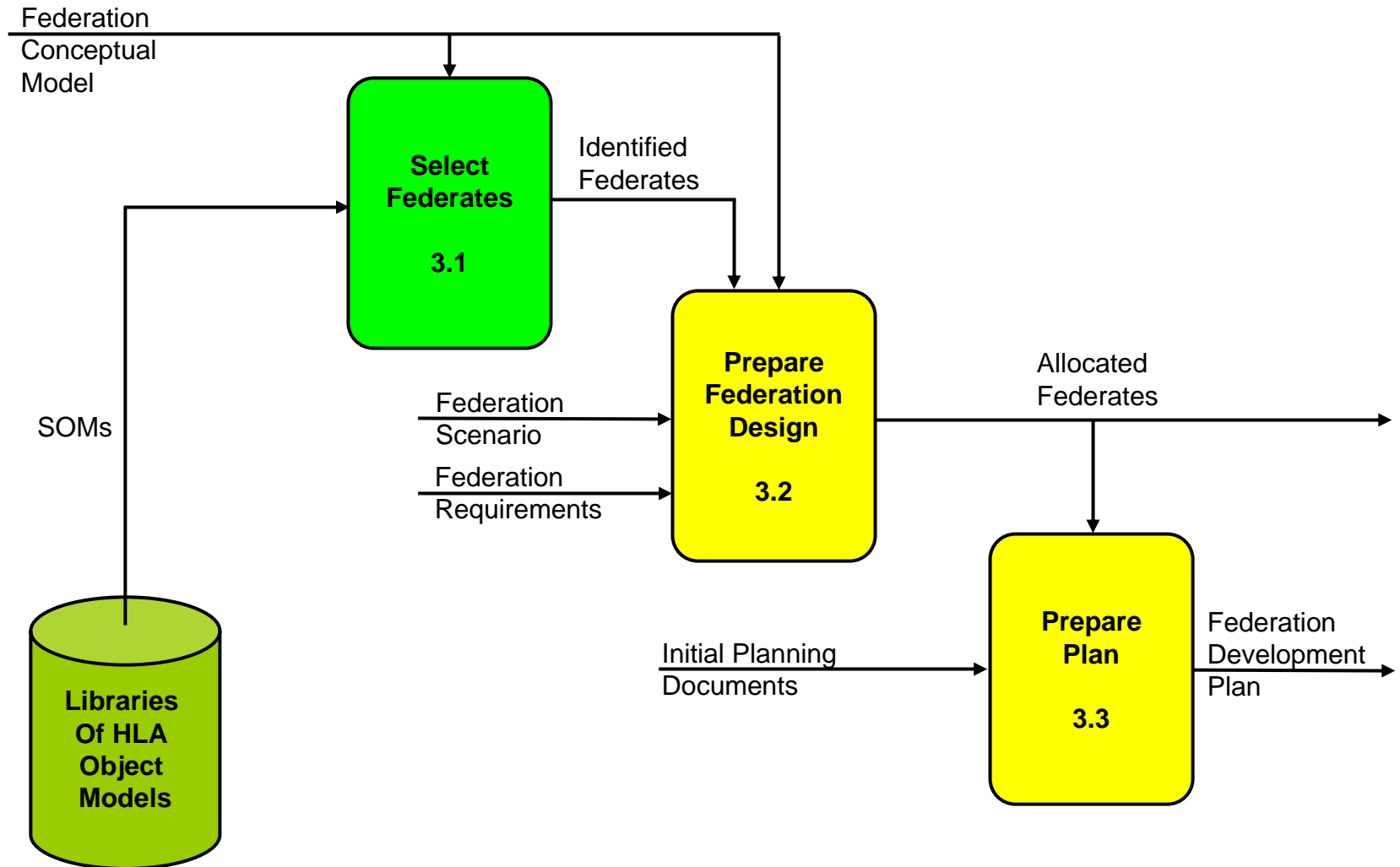## Detailed Artifact Rework Analysis
by FEDEP Step

Step 1 - Define Federation Objectives

Overall Plans →

Existing Domain Descriptions →

Info on Available Resources →

Identify User/Sponsor Needs

1.1

→ Needs Statement →

Develop Objectives

1.2

→ Federation Objectives Statement →

→ Initial Planning Documents →

Step 2 - Perform Conceptual Analysis

Federation Agreements →

Scenario Instance(s) →

FDD →

Federation Development Plan →

FOM →

Plan Execution

5.1

Federation Environment Description →

Federation Environment Description →

List of Selected (Existing) Federates →

Modified / New Federates →

Implemented Federation Infrastructure →

Supporting Databases →

Integrate Federation

5.2

Integrated Federation →

Federation Test Criteria →

Federation Development Plan →

Federation Agreements →

Test Federation

5.3
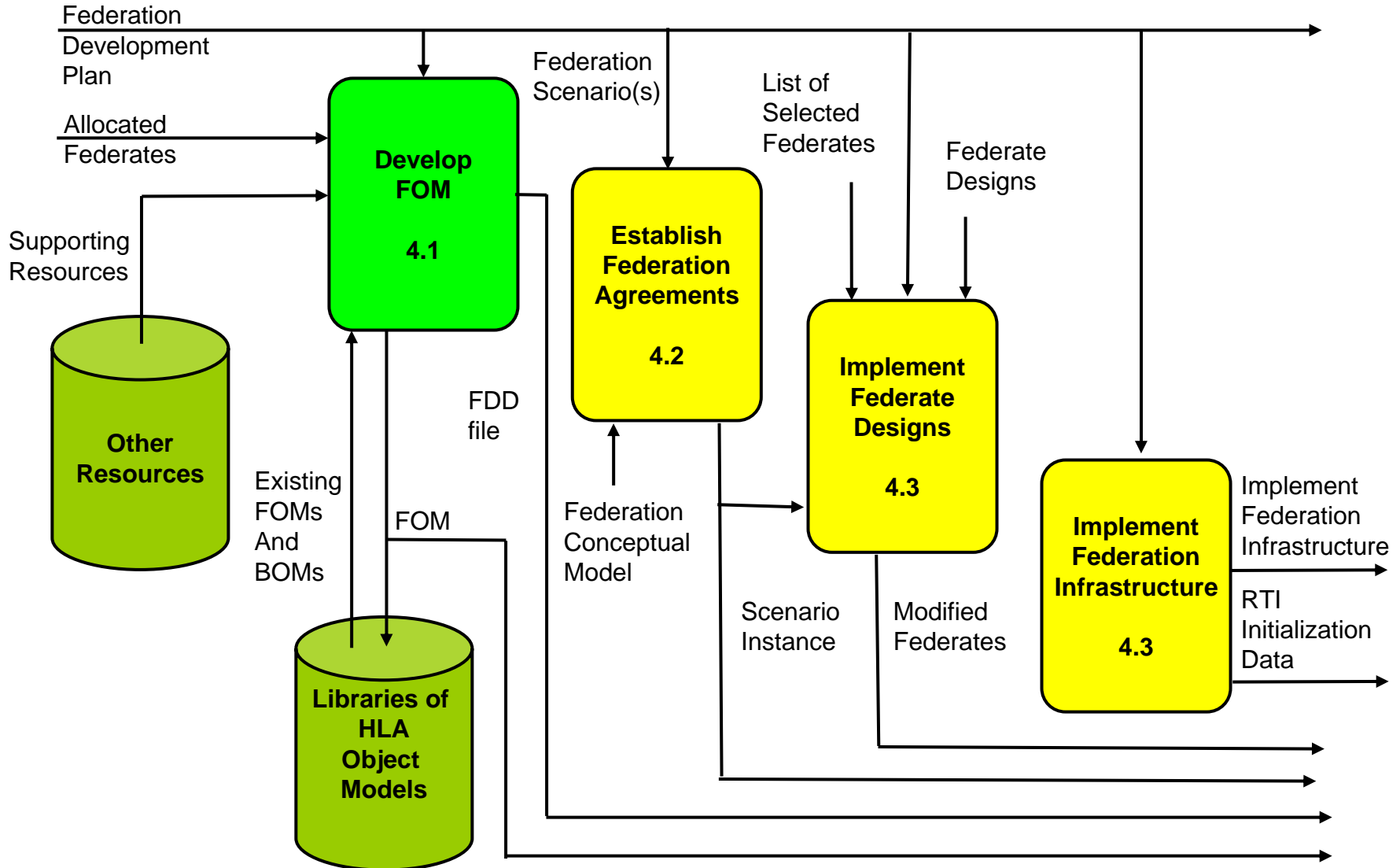
Tested Federation →

Step 6 - Execute Federation and Prepare Outputs

Federation Environment Description → **Execute Federation 6.1**

Tested Federation →

Federation Development Plan →

Federation Agreements →

Documented Execution Problems

Raw Execution Outputs

Prepare Federation Outputs 6.2

Derived Outputs

Step 7 - Analyze Data and Evaluate Results