Generating a Robust System Architecture Using ARCADIA Capella

15 Nov 2022

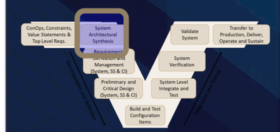
Prof. Eric Dano
ericdano@gwu.edu

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Outline

- Architecture Overview
- Proposed Architecture Development Method (Uses Example of a Modeled Missing Person Locater (MPL) System)
  - ARCADIA Capella based
    - Engineering standards define views and order to facilitate user workload
    - Supplemental elements are added to provide additional context
- Live Capella Demonstration (Time Permitting)

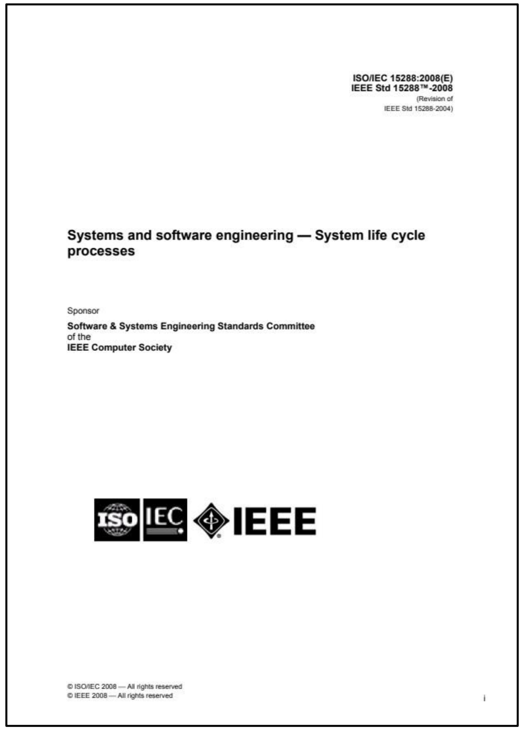
Architecture Overview



System Architecture Definition

- **<u>System Architecture</u> ≡ Fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.** **Ref: ISO/IEC/IEEE Std 15288:2008**

- System Architecture is called out as a critical activity in all SE references



ISO/IEC 15288:2008(E)
IEEE Std 15288™-2008
(Revision of
IEEE Std 15288-2004)

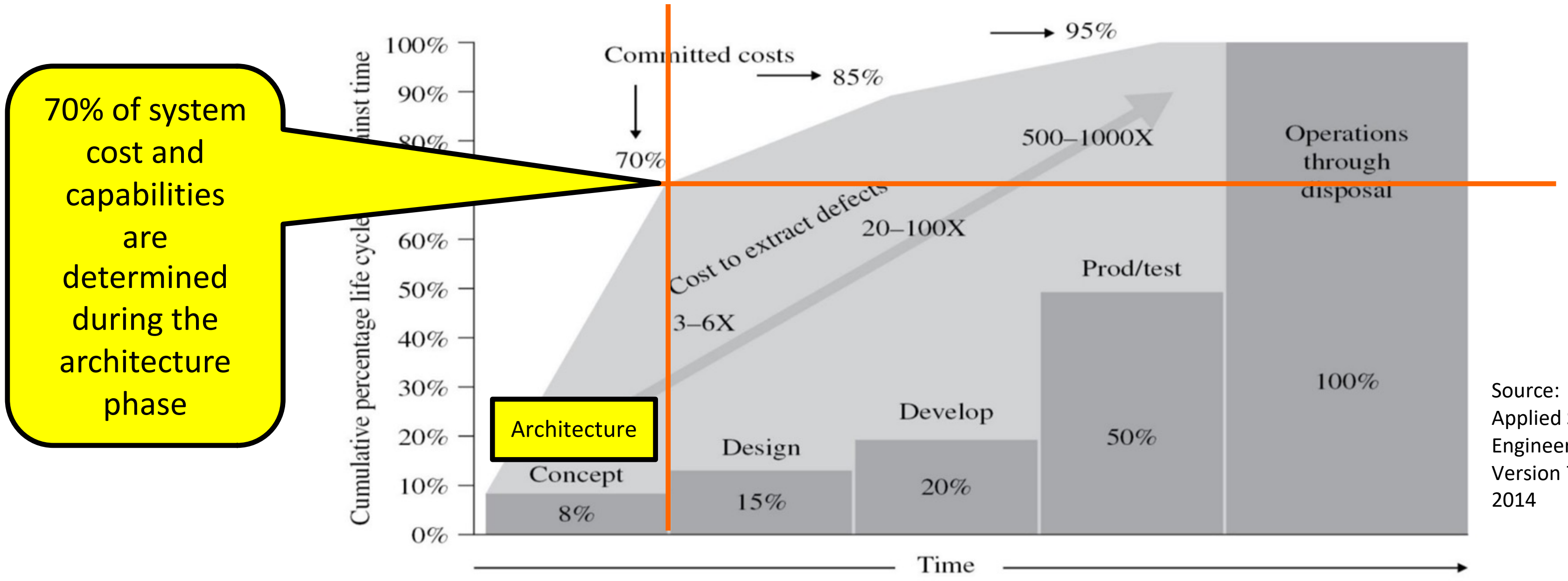
Systems and software engineering — System life cycle processes

Sponsor
Software & Systems Engineering Standards Committee
of the
IEEE Computer Society

© ISO/IEC 2008 — All rights reserved
© IEEE 2008 — All rights reserved

ISO/IEC/IEEE 15288:2008 is important since it defines the System Architecture Process

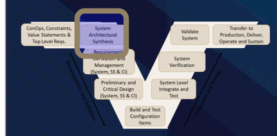
Value of System Architecture



Source: E. Honour Applied Systems Engineering Version 7.7, June 2014

- Architectural decisions made at the very beginning of the system lifecycle set the system costs (and capabilities).
- Architecture is done holistically, considering all relevant disciplines and the system's entire lifecycle
- Key attributes like Affordability, Modularity, Resilience, etc., have to be architected into a system, before they are designed into a system.

The Architecture and associated requirements decomposition set the system development up for success



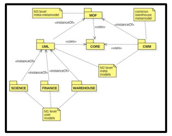
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Model Based Architecture Approaches

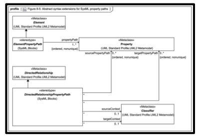
Object Oriented System Engineering Method (OOSEM) w/ SysML with Profile extensions

Language

SysML



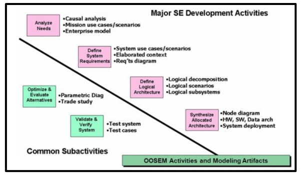
SysML
Meta Model



SysML SA
Profile Extensions

+

Methodology/ Framework



OOSEM

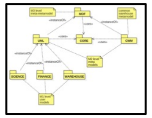
N/A

Framework

+

MBSE Tool

Magic Draw



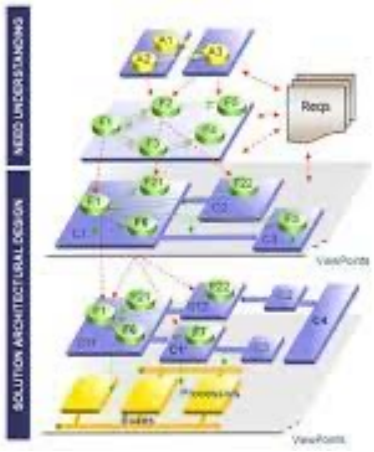
SysML Plugin

Supplemental
Views

ARChitecture Analysis and Design Integrated Approach (ARCADIA)/ Capella

ARCADIA

Language/ Methodology



ARCADIA Language (A mix of SysML, NAF simplified/ modified to focus on SA)

Arcadia Meta Model

+

MBSE Tool

Capella

Requirements
Plug-In
+
Supplemental Views

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

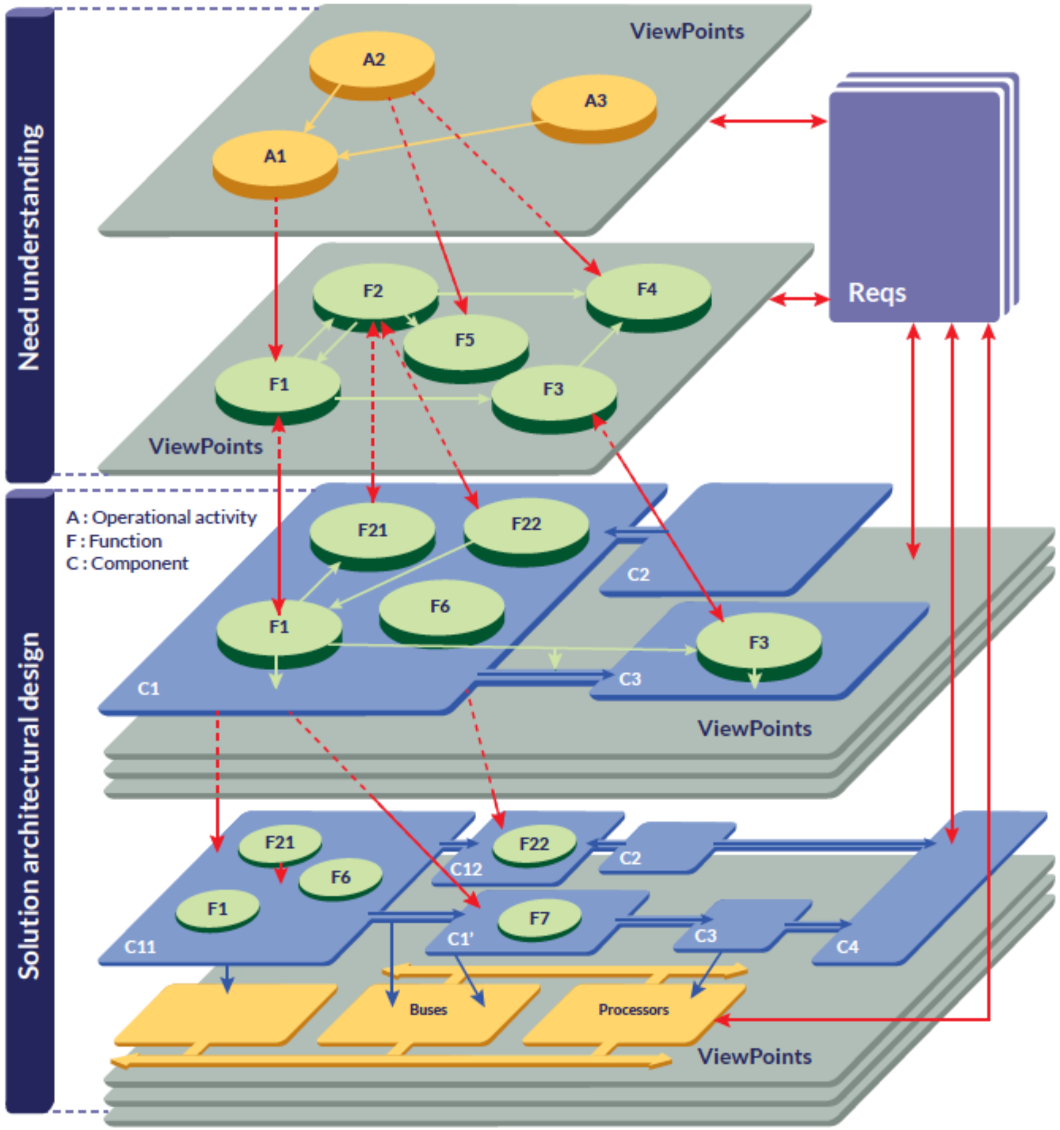
Clarification

The objective is facilitating the digital transformation for the architecting all Engineered Systems.
All approaches that move us closer to this objective are GOOD approaches

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

The ARCADIA Method

- The ARCADIA is an architecture-centric system and software architecture engineering method
- The ARCADIA method was created by Thales (FR) in 2007
  - 4-Step Architecture Development Methodology (ADM)
  - A methodology is provided for each ADM step
  - Common methods in each ADM step reduce learning curve
  - Some auto-generated views
- Capella is the MBSE tool portion of the ARCADIA Method.
  - Capella is open-source software and can be downloaded for free
  - There are some free and commercial add-ons available that can extend Capella's capabilities.
  - Significant error prevention logic is built into tool
  - Artifacts are importable between ADM steps
  - Great tool for rapidly creating functional thread artifacts for a project's SFR.
  - SIEMENS has created adaptors which extend Capella to be interoperability with domain tools (MBE)



THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

ARCADIA ADM

ARCADIA ADM aligns with the IEEE 15288:2015 Process

1 **Operational Analysis**

Define Stakeholder Needs and Environment

Capture and consolidate operational needs from stakeholders
Define what the users of the system have to accomplish
Identify entities, actors, roles, activities, concepts

2 **System Analysis**

Formalize System Requirements

Identify the boundary of the system, consolidate requirements
Define what the system has to accomplish for the users
Model functional dataflows and dynamic behaviour

3 **Logical Architecture**

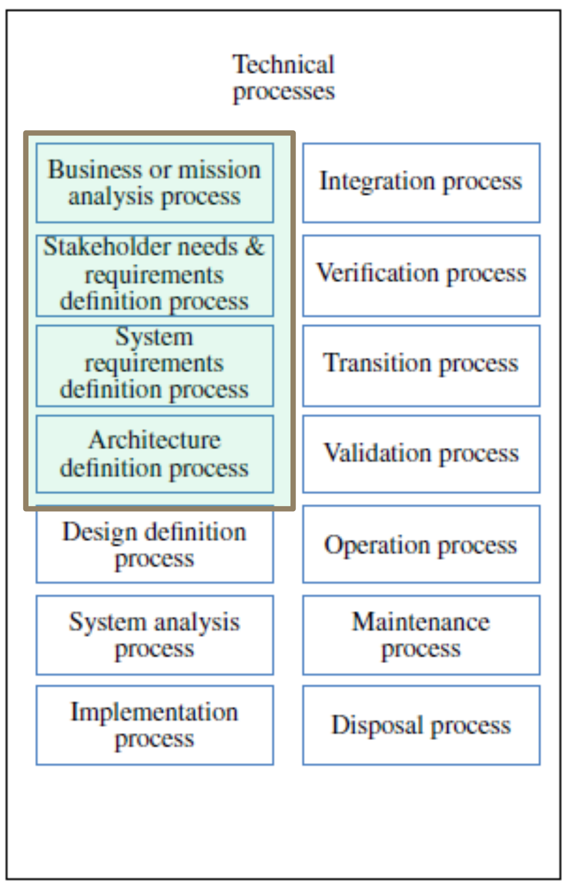
Develop System Logical Architecture

See the system as a white box
Define how the system will work so as to fulfill expectations
Perform a first trade-off analysis

4 **Physical Architecture**

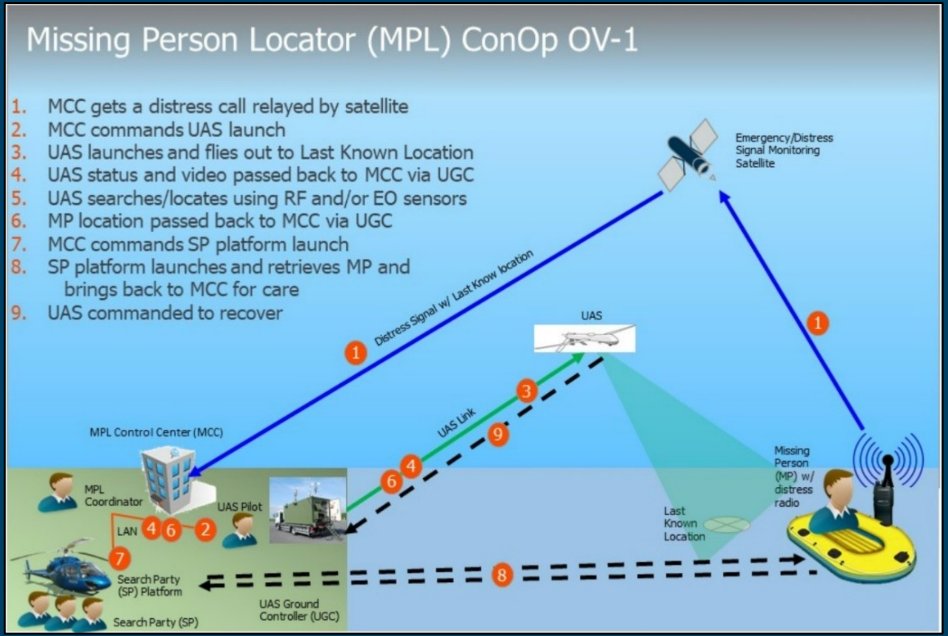
Develop System Physical Architecture

How the system will be developed and built
Software vs. hardware allocation, specification of interfaces, deployment configurations, trade-off analysis



Unlike many other tools, Capella is extremely robust in the Conceptual Architecture

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC



Proposed Approach

Example: Missing Person Location (MPL) System to locate and rescue people in distress on the sea

ARCADIA Based Architecture Synthesis Method

The ARCADIA Method will be used with the proposed modifications below:

Defined Architectural Description (subset of ARCADIA Framework)

1. A subset of the defined ARCADIA views in each ADM Step will be used to minimize redundancy and further reduce modeling time <u>without effecting system thinking/architecture robustness</u>
2. The order the views are created in each ADM Step are proscribed to minimize operator workload and to take advantage of auto-generated views

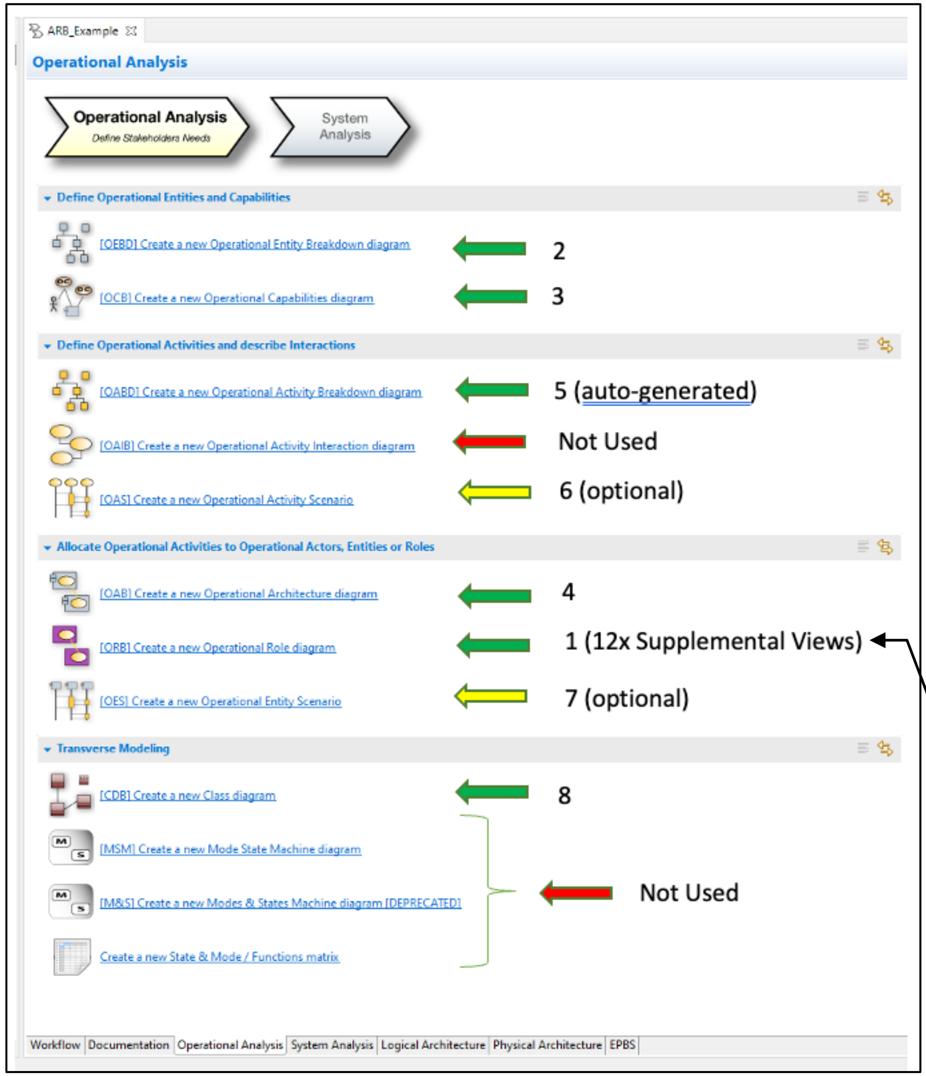
Addition of non-ARCADIA Architectural artifacts for broadened system context

3. Supplemental views will be added into the ARCADIA model for enhanced context. Supplemental Views are in four general categories
    - Customer/Project Information for better Conceptual picture of system
    - DoDAF Architectural views for better system context
    - Architecture Related products (e.g. Trades/Analyses, Risks, etc.)
    - "Illities" related. Ensures a transdisciplinary architecture is considered over the system lifecycle

Note: Supplemental views will be added into objects in each of the 4 ADM steps. No active model elements are included in the supplemental views.

Resulting Method yields a well documented, robust, transdisciplinary system architecture

1. Operational Analysis (a.k.a. Conceptual Arch.)



- Operational analysis defines what is required to meet customer needs at an abstract level
- Concept level decisions made via Trades, Analyses, M&S, and multiple system thinking tools (e.g., system dynamics)
- Defines required system elements /actors and interactions
- Defines top level Operational Capabilities and decomposes into lower-level Operational Activities required to meet customer's ConOps
- Defines Operational Activity interactions
- Maps Operational Activities to Operational Entities and Actors
- Covers the first 2 architecture processes called out in IEEE 15288:2015 in detail
- Creates Conceptual Architecture (No definition of system at this level)

1.1 ARB Composition/Staff
1.2 Project Description
1.3 Project Assumptions
1.4 Project Discriminators
1.5 Project Schedule/Milestones
1.6 Project Costing Evaluation
1.7 Project ConOp (OV-1)
1.8 Project Supplemental Information (AV-1)
1.9 Project Operational Environment
1.10 Project KPP/KSA/Value Statements
1.11 Project Integrated Dictionary (AV-2)
1.12 Project Acronyms (AV-2)

Operational Analysis - View Acronyms (in order used)

[ORB]	Operational Role (Blank) Diagram is just used to add supplemental views
[OEBD]	Operational Entity Breakdown Diagram
[OCB]	Operational Capability (Blank) Diagram
[OAB]	Operational Architecture (Blank) Diagram
[OABD]	Operational Activity Breakdown Diagram – Auto-Generated
[OPD]	Operational Prosses Diagram – Intermediate product for [OAS]
[OAS]	Operational Activity Scenario – OPTIONAL
[OES]	Operational Exchange Scenario – OPTIONAL
[CDB]	Class Diagram (Blank)

Operational Analysis - Supplemental Views (step 1)

ARB Composition and Required Staff

ARB Composition and Staffing

Position	Name(s)	Mandatory Member	Prgm	Ind.
ARB Chair		X	X	
System Architect or Chief Engineer		X	X	
ITE Lead		X	X	
Reliability Maintainability Safety (RMS)		X	X	
Integrated Logistics Support (ILS)		X	X	
Operations/NPI/P2L		X	X	
Discipline Leads/SMEs		X	X	
Cyber SME		X	X	
Independent Assessor		X		X
Independent Assessor		X		X
Product Line Engineering Lead			X	
Capture Lead/Program Manager			X	
PEM/SEITL			X	
I&T Lead			X	
Program Protection Lead			X	
Others as needed			X	X

Multi-discipline/Independents Ensure Architecture is Complete

Project Description

Missing Person Locator (MPL):

- Capability Provided: Unmanned Aerial System (UAS) Based Missing Person Locater
- Customer: US Coast Guard
- Customer Value Statements: Time to geolocation, time to recovery
- Expected EMD RFP/Proposal Dates:
  - Final: 1 April 2030
  - Response Due: 1 October 2030
  - Award Date: 1 January 2030
- Project Value: $100M CPFF
- Adjacent Market Access: ES based UASs
- Work Split/Partners: General Atomics (UAS)
- Price To Win: $100K per ship set (UAS and sensors, Ground control and MPL Command Center interface)
- Capture Activity Status: Study phase concluded.

Project Assumptions

Assumptions:

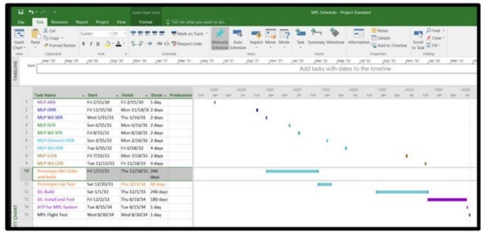
- Will build 1 non-deliverable prototype UGS and UAS with link
- Will deliver 2 operational UGSs and UASs with links
- Hand In – MPL Command Center IRS/IDD and ICD
- Capital – UAS Datalink Simulator
- GFE UGS antenna will be used
- ESS required on all WRAs
- MOSA alignment will be prioritized in a trade studies and analyses
- Lab NCA-8000 will be available for system development
- Verification testing will be accomplished via live flight tests
- MTBF of 200 hours (Note: not in spec, based on 40x 5 hour missions)

Project Discriminators

Project Discriminators:

- Early prototyping of UAS Ground Station link and UAS
- Company has won last three MCC upgrade contracts with outstanding CPARS from the customer
- Sole teaming agreement with General Atomics (strongly favored by selection committee)

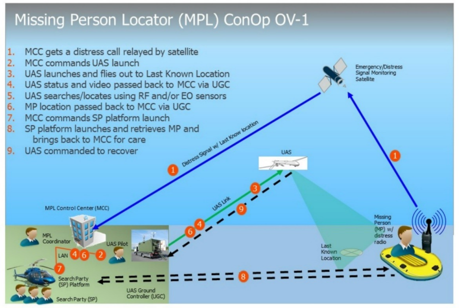
Project Schedule



Cost Assessment

Element	Amount ($K)
Labor Dollars – Direct	$
Labor Dollars – OH	$
Direct Material	$
Material Handling	$
Other Costs	$
Adjustments	$
Facilities	$
Profit	$
Sell Price	$

Project Graphical ConOp (OV-1)



MPL Additional Information - AV-1

Project Additional Information:

1. ***Identification*** – The MPL is a UAS based Search and Rescue at sea system.
2. ***Purpose*** – The MPL will perform Optical and RF detection and location of a Missing Person lost at sea, so they can be rescued and returned to safety.
3. ***Scope*** – The MPL EMD phase will develop and demonstrate the identified critical technology elements for later follow on LRIP.
4. ***Intended Users*** – The NPLwill be operated by the US Coast Guard.
5. ***Context*** – The MPL will be used to reduce the rescue time for people lost at sea, while minimizing required helicopter and search party loitering time

MPL Operational Environment

Project Operational Environment:

- Temperature (operating -35 to +50 deg C, non-op -50 to +70 deg C, storage -50 to +85 deg C)
- Altitude (operating 0 to 35,000 ft, non-op 0 to 40,000 ft)
- Temperature-Altitude to MIL-STD-810 Method 504.1, Category 5
- Function shock to MIL-STD-810E, Method 516.4, Procedure I
- Crash Safety Shock to MIL-STD-810, Method 516.4, Procedure V
- Vibration (IAW customer provided profile adjusted for isolation)
- Humidity 100% Relative Humidity (RH) up to 40 deg C including conditions w/ condensation
- Sand & Dust environment to satisfy Method 510.3 of MIL-STD-810
- Salt Fog in accordance with Method 509.3 of MIL-STD-810E
- Fungus to satisfy Method 508.4 of MIL-STD-810.
- EMI/EMC to MIL-STD-461 for Class A1b equipment, verified by MIL-STD-462 for: CE102, CS101, CS114, CS115, CS116, RE102 and RS103
- Power to MIL-STD-704F

MPL KPPs, KSAs and Value Statements

KPP	Status	Performance
Time to MP geolocation		Analysis shows 30 minute margin
Time for recovery		Analysis shows 5 minute margin
Operational Availability (Ao)		Supportability feature achieve Ao >0.9
KSA	**Status**	**Performance**
Line of Bearing Accuracy		Meets specification with 5% margin
RF Field of View		Omnidirectional coverage
Optical Field of View		10% greater than specification
Image Processing - Object detection in ocean		based on Digital Twin testing
Mean Time to Repair (MTTR)		Meets spec with 30% margin
Value Statements	**Status**	**Details**
Modular Open System Approach	N/A	Customer desires maximum modularity to reduce NRE/time for new capability incorporation and to mitigate obsolescence

Project Dictionary - AV-2

Project Integrated Dictionary:

Search and Rescue - The act of locating and rescuing a person(s) in distress.

Electronic Support (ES) - Division of electronic warfare involving actions tasked by, or under direct control of, an operational commander to search for, intercept, identify, and locate or localize sources of intentional and unintentional radiated electromagnetic energy for the purpose of immediate threat recognition, targeting, planning and conduct of future operations.

Project Acronyms - AV-2

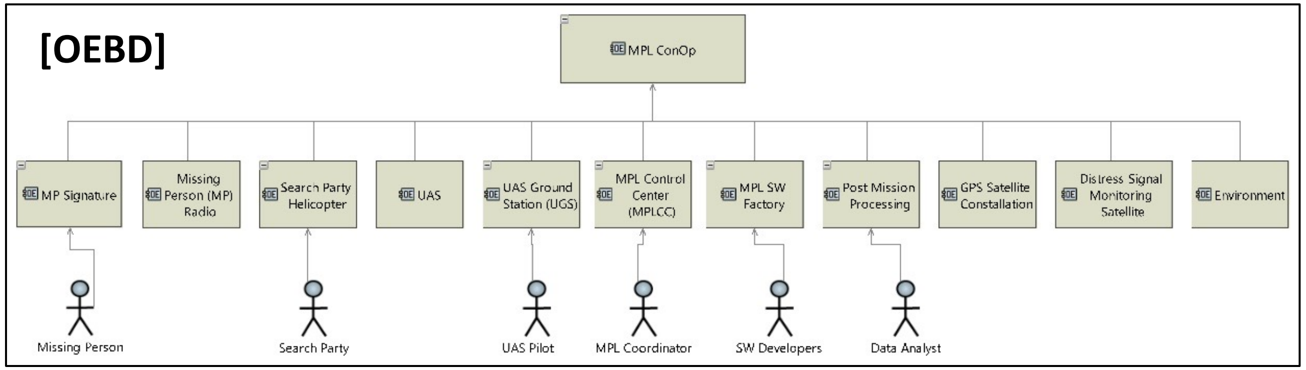
Project Integrated Dictionary:

KPP – Key Performance Parameter
KSA – Key System Attribute
MCC - MPL Command Center
MOSA – Modular Open System Approach
MP - Missing Person
MPL - Missing Person Locator
SAR - Search and Rescue
SP - Search Party
UAS - Unmanned Aircraft Systems
UGC - UAS Ground Controller

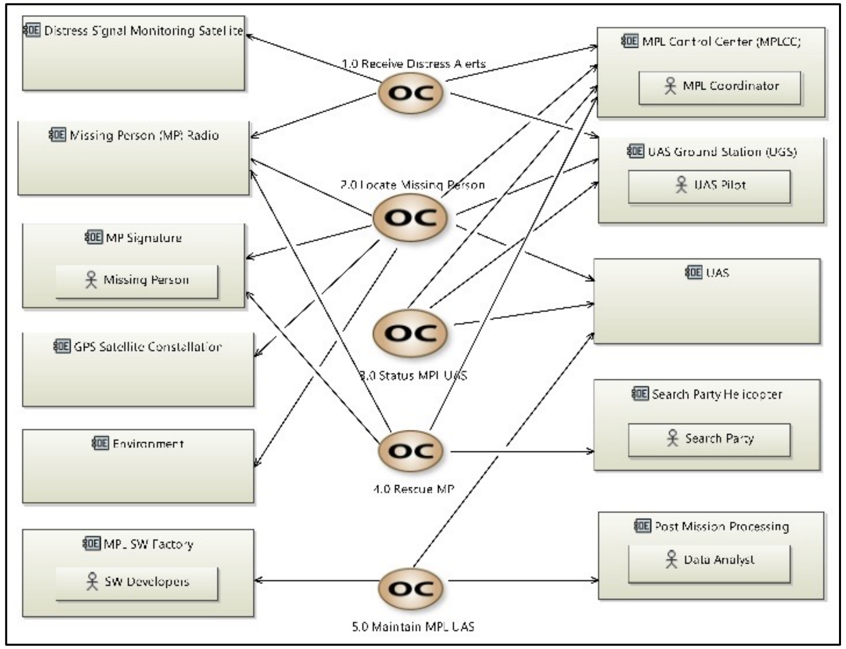
Mix of Programmatic and DoDAF Supplemental views into the model add valuable Context

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

1. Operational Analysis – Views (1 of 4)



[OCB]

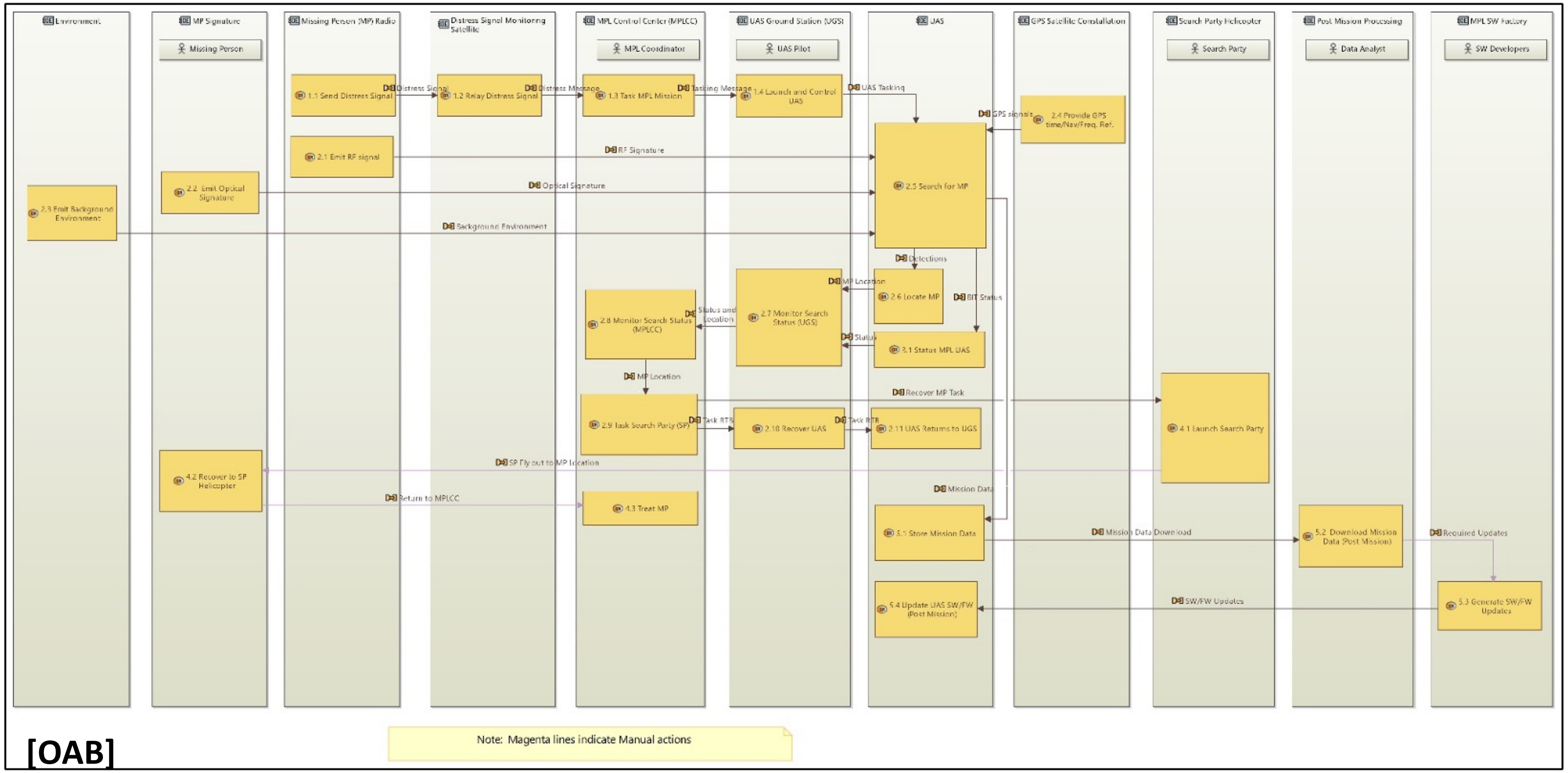


2. [OEBD] defines required Operational Entities and Actors required to meet the customer's needs, ConOps, requirements, constraints and value statements documented in the 12 supplemental views.

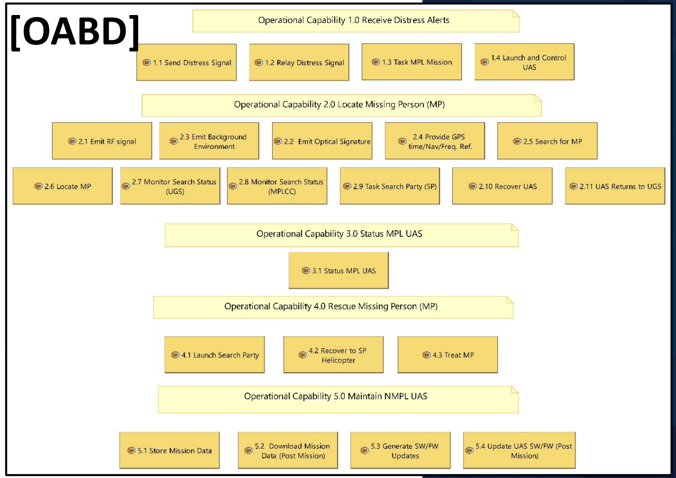
3. [OCB] defines required Operational Capabilities required to meet the customer's ConOps documented in the supplemental views. Elements/Actors from the [OEBD] are mapped to the Operational Capabilities.

Remember the "Illities" and involve the domain experts at this level !!

1. Operational Analysis - Views (2 of 4)

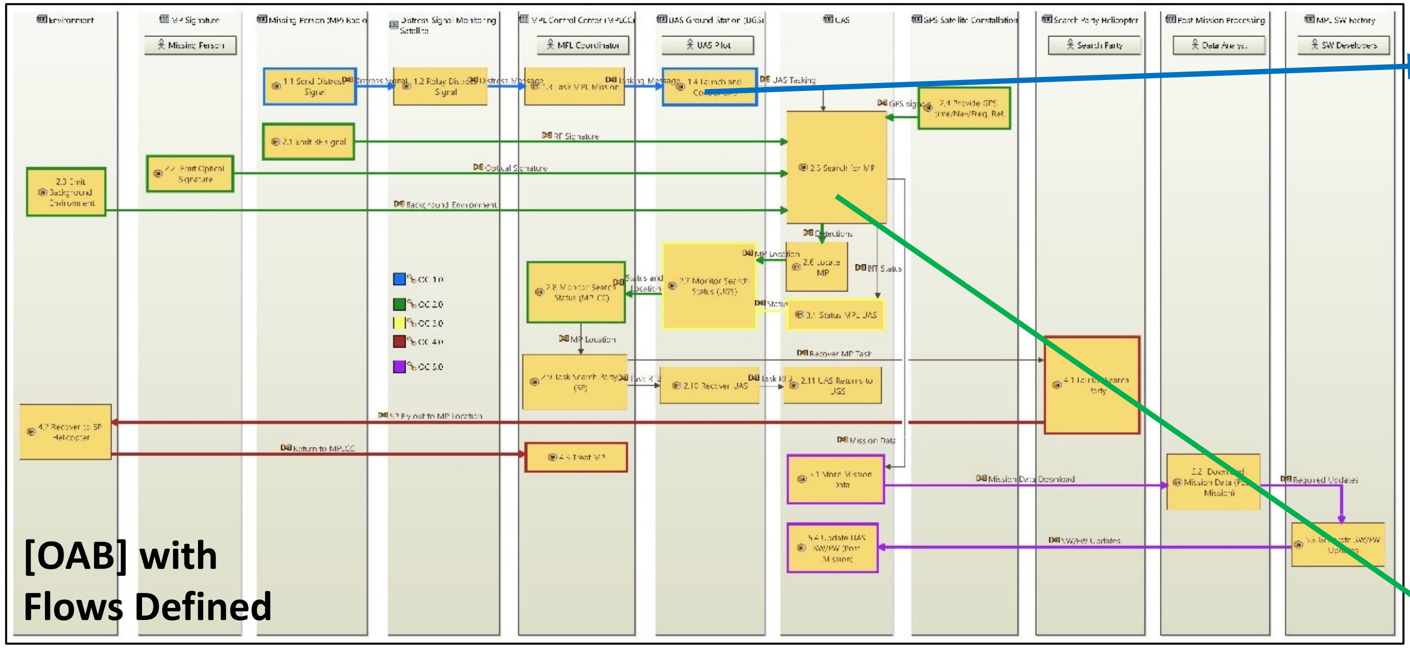


5. [OABD] is auto-generated. It imports the Operational Activities defined in the [OAB]. Notes can be added from the Right Palette tool to identify the parent Operational Capability

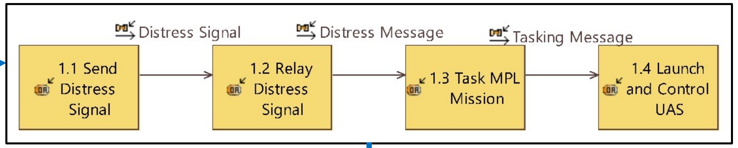


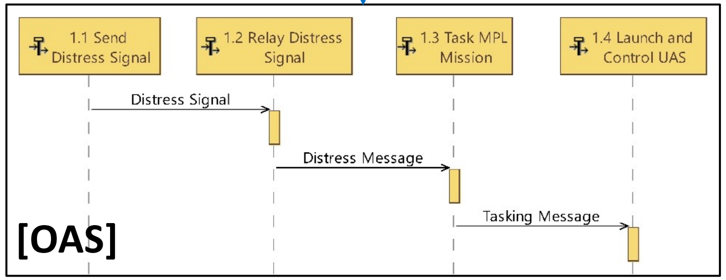
4. [OAB] Imports the [OEBD] defined entities and actors and sets them up as vertical columns. The Operational Capabilities defined in the [OCB] are decomposed into lower- level Operational Activities and placed in the appropriate Operational Entity/Actor Column. Data exchanges are added between all Operational activities. **Note**: In many Capella examples, the [OAIB] is used to first define the Operational Activities and Data Exchanges. This was removed for redundancy in this method.

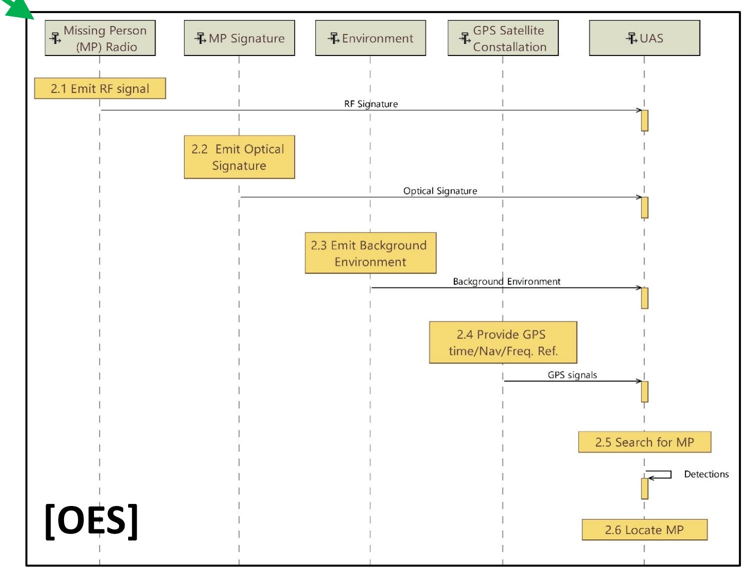
1. Operational Analysis – Views (3 of 4)



[OPD]







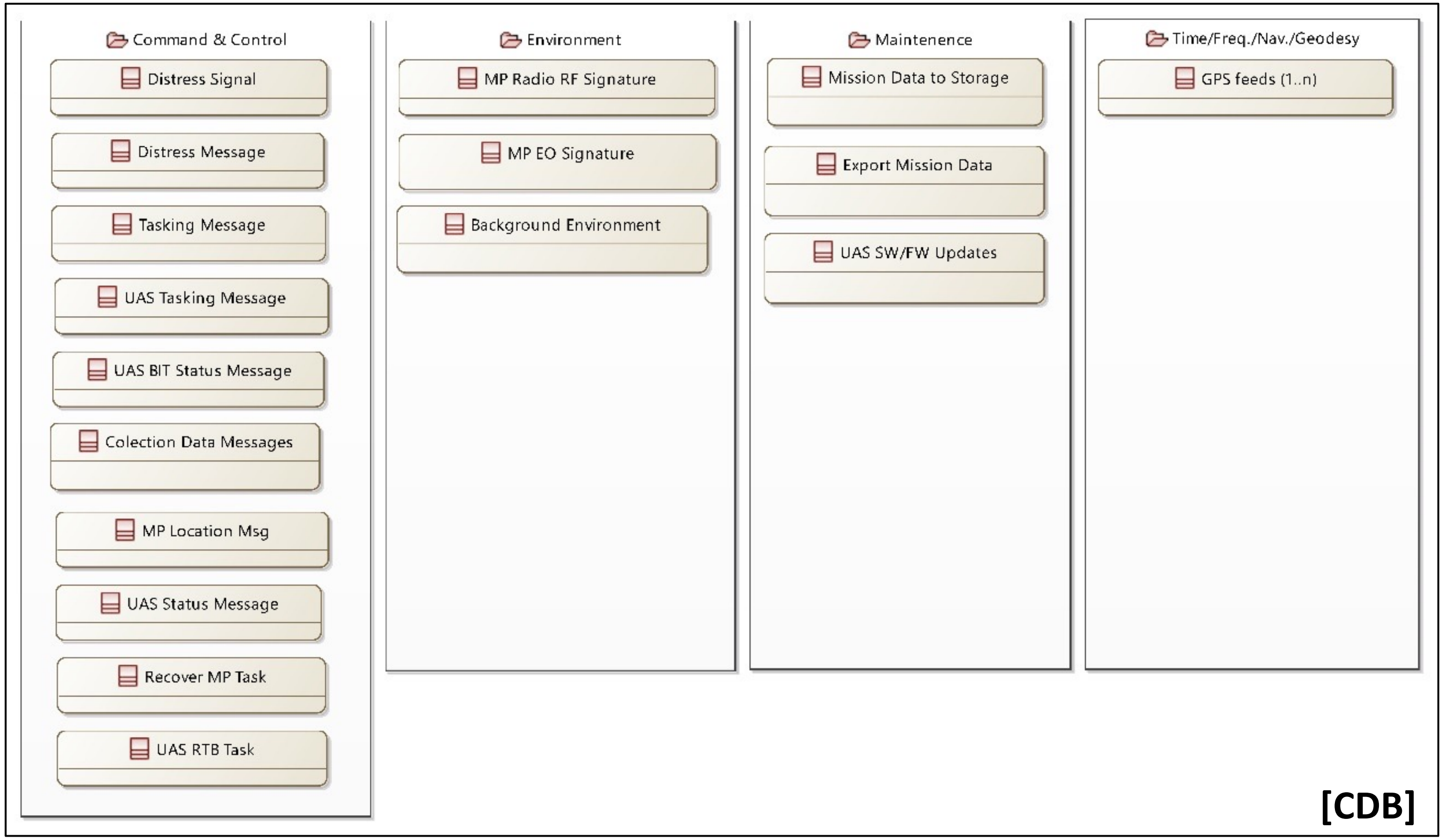
6. Activity threads can be added to applicable portions of the [OAB] by defining the path for each thread (activities and exchanges) and creating an intermediate [OPD] which extracts the selected thread. The [OPD] can be selected and transitioned directly into and [OAS].

7. [OES] is created from the activity explorer. Since previous artifacts defined all Operational Elements, they can be pulled in directly from the Right Tool Palette (first entities, then activities and then data exchanges). The tool will only offer adding exchanges between the 2 selected entities

Note: [OAS] and [OES] diagrams must be linked to an Operational Capability and will show up in the Capability's directory in the Project Explorer

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

1. Operational Analysis – Views (4 of 4)



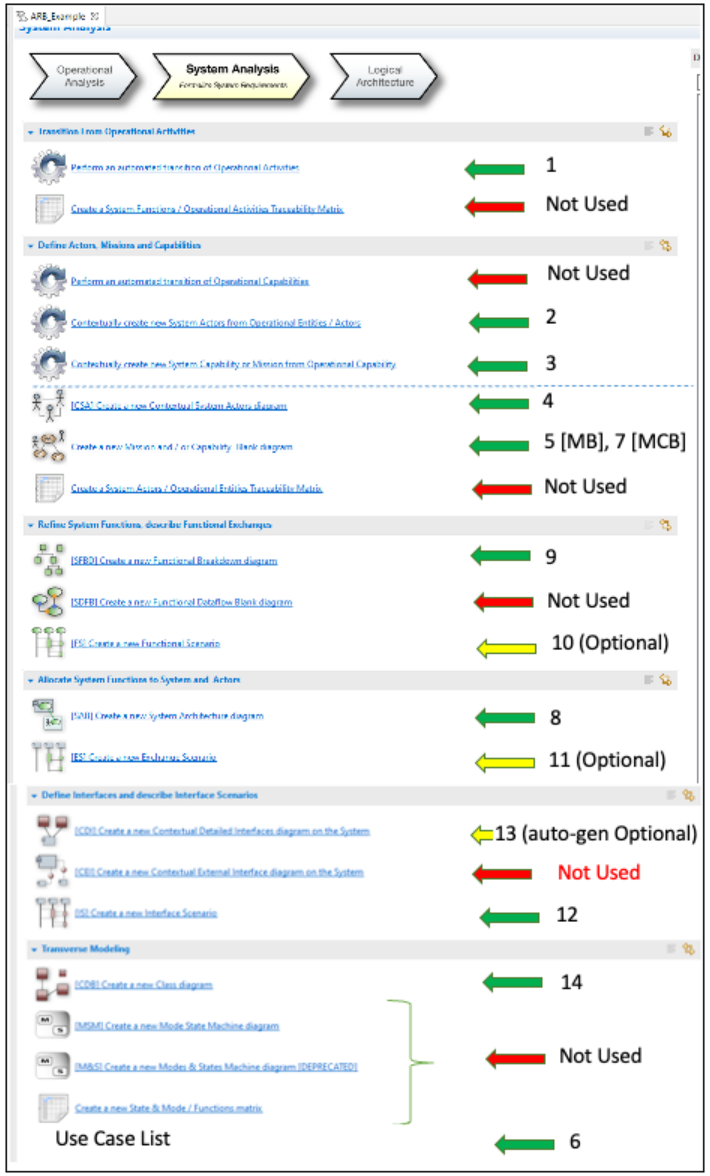
8. General Data Classes and Packages can be generated at the conceptual level based on the interfaces defined in the [OAB]

Acronyms

EO	Electro-Optic	RF	Radio Frequency
FW	Firmware	RTB	Return to Base
GPS	Global Positioning System	SW	Software
MP	Missing Person	UAS	Unmanned Aerial System

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

2. System Analysis (aka Functional Architecture)

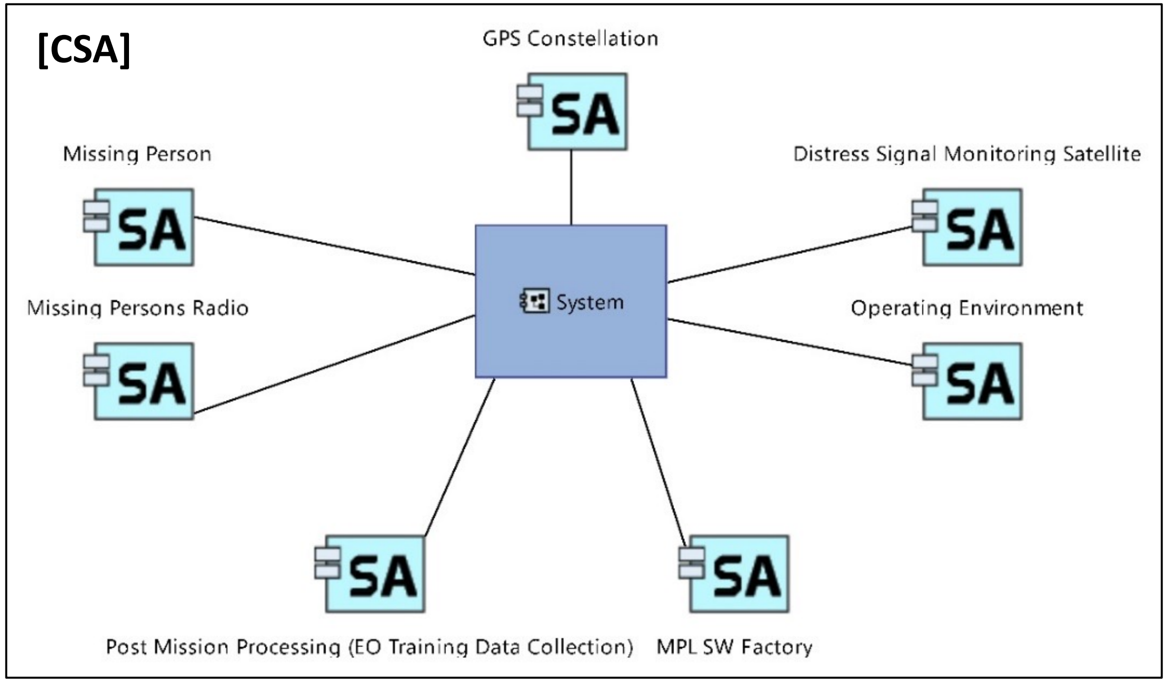


- Imports and Transfers artifacts from Operational Analysis to System Analysis
  - Step 1) Converts on import: Operational Activities → System Function (Level 0)
  - Step 2) Converts on import: Operational Actor/Entities → System Actors/Entities
  - Step 3) Converts on import: Operational Capabilities → System Missions (also transfers Operational Capabilities from ADM Step 1 into ADM step 2)
- System level decisions made via Trades, Analyses, M&S, and multiple system thinking tools (e.g., system dynamics)
- System Use Cases are Defined (System Capabilities in Capella)
- Defines the system boundaries (i.e., Black Box view) and Context diagram to external entities.
- Defines flows in/out of the system and external interfaces
- Data Model is updated into internal/external packages
- Maps System Functions (Level 0) as either internal/external to system
- Internal System Functions are decomposed further into sub functions (Level 1)
- Defines flows between System Functions/Subfunctions
- Creates Functional Architecture

System Analysis - View Acronyms (in order used)

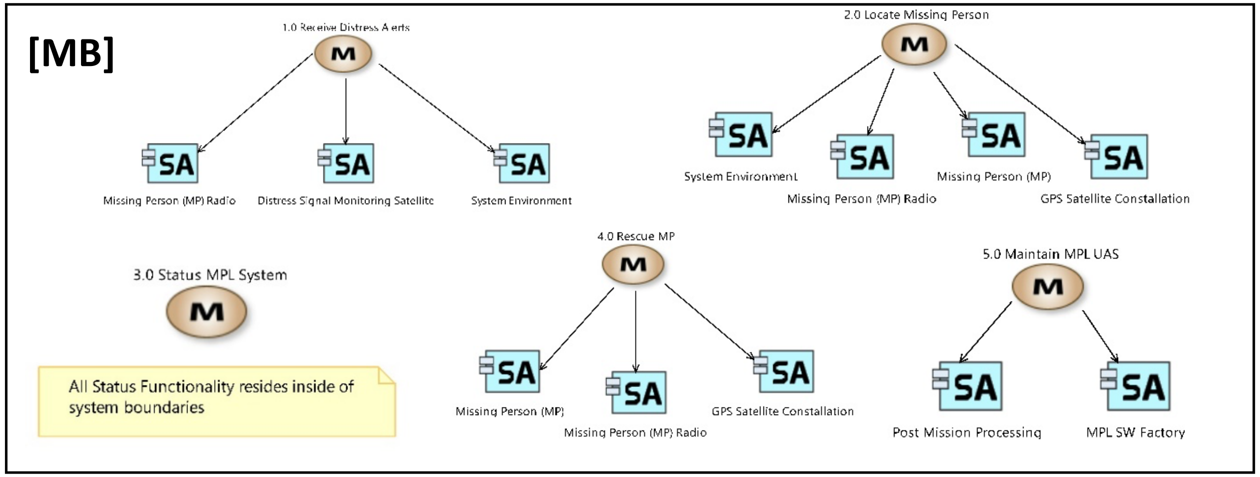
[CSA]	Contextual System Actors Diagram – Partially Auto-generated
[MB]	Mission (Blank) Diagram
[MCB]	Mission Capability (Blank) Diagram (also used for Use Case list supplemental view)
[SAB]	System Architecture (Blank) Diagram
[SFBD]	System Functional Breakdown Diagram – Auto-Generated
[SFCD]	System Function Chain Diagram – Intermediate product for [FS]
[FS]	Functional Scenario – OPTIONAL
[ES]	Exchange Scenario – OPTIONAL
[IS]	Interface Scenario
[CDI]	Contextual Detailed Interface Diagram – OPTIONAL - Auto-Generated
[CDB]	Class Diagram (Blank)

2. Systems Analysis – Views (1 of 5)



4. [CSA] is initially autogenerated with a dark blue “System” block an all previously define/imported System Actors/Entities linked to it (from [OEBD] and [OAB] imported to ADM step 2). This is where the system’s <u>external</u> boundaries get initially defined, by selecting, highlighting and deleting any System Actors/Entities that are <u>in the system</u> (or no longer required). The result will be a simple context diagram of the system showing external Entities in light blue boxes.

5. [MB] will map the System Missions (from [OCB] imported to ADM step 2) to the external System Actors/Entities defined in the [CSA].

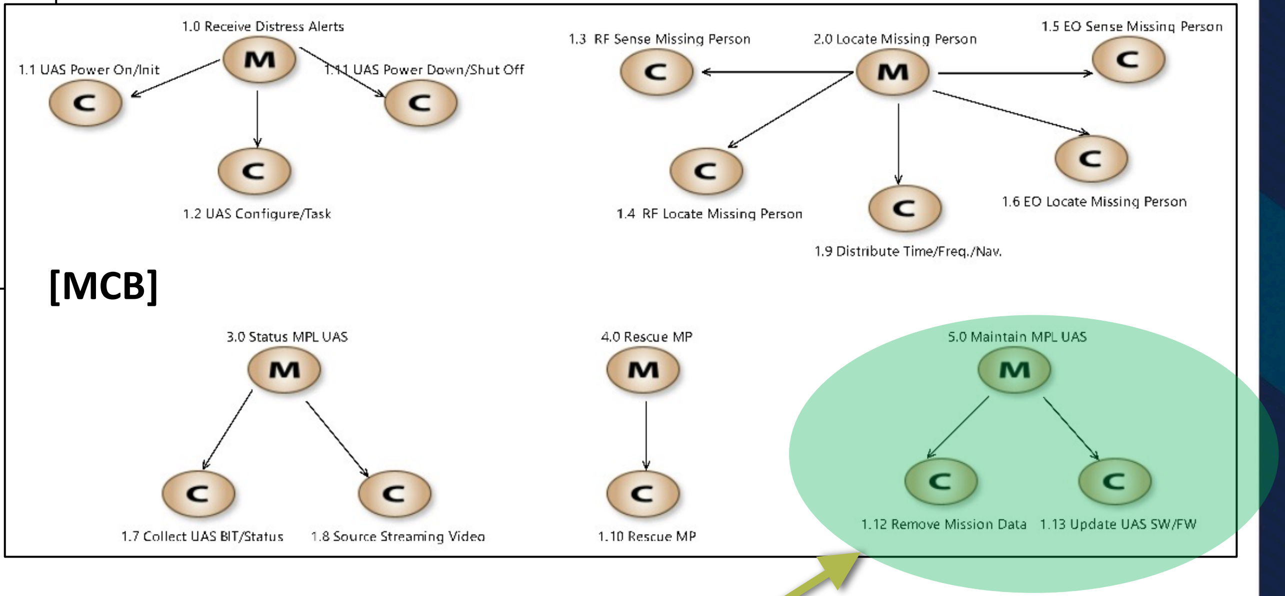


2. Systems Analysis – Views (2 of 5)

System Capabilities required to meet Missions/Operational Capabilities/ ConOpsCONOPs:
1.1 – UAS Power On/Initialization
1.2 – Configure/Task UAS
1.3 - RF Sense Missing Person
1.4 – RF Locate Missing Person
1.5 - EO Sense Missing Person
1.6 – EO Locate Missing Person
1.7 – Collect UAS BIT/Status
1.8 – Receive Streaming Video
1.9 – Distribute Time/Freq./Nav.
1.10 – Remove Mission Data
1.11 - UAS Power Off/Shutdown
1.12 – Update UAS Software
1.13 – Update MPL CC Software
1.14 – Update UGS Software
1.15 – Rescue Missing Person
...

6. Use Cases (Called System Capabilities in Capella) are decomposed from the Missions (from [OCB] imported to ADM step 2). They are added into the model as a supplemental view for reference

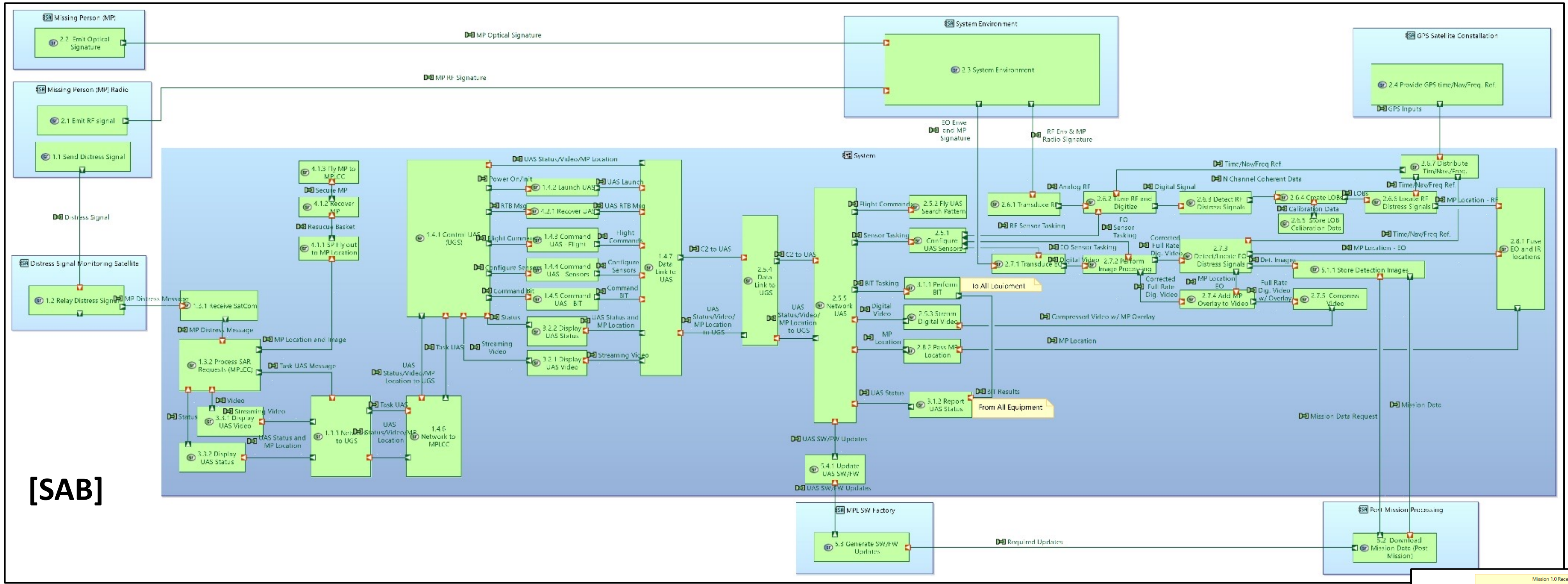
The Use case list should be vetted with the customer and all "illities" POCs to ensure completeness



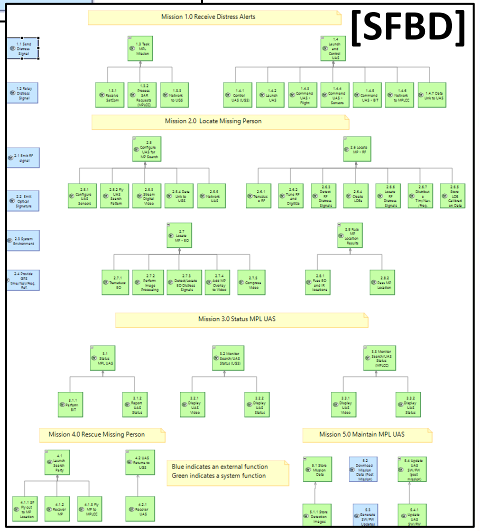
7. A [MCB] is created to show the mapping of the Missions (from [OCB] imported to ADM step 2) to System Capabilities which were created based on the supplemental Use Case list.

Use Cases/System Capabilities must cover "illities", cost, test and other quality functionality

2. Systems Analysis – Views (3 of 5)



[SAB]

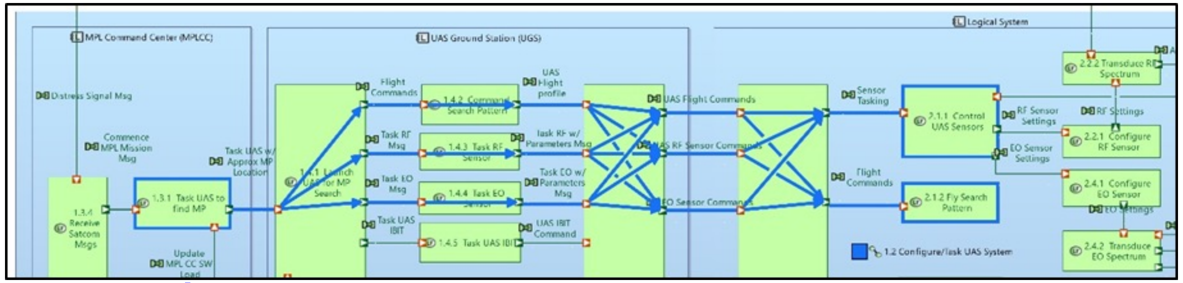


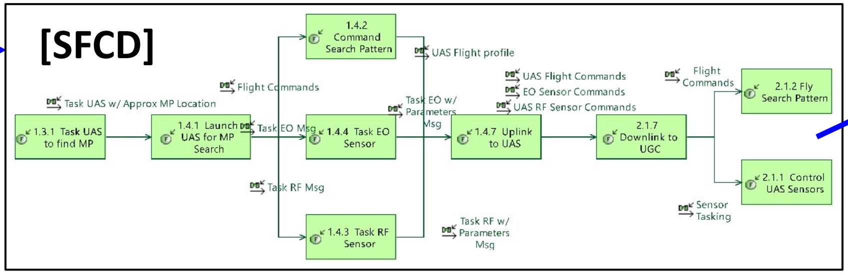
9. [SFBD] is auto-generated from the [SAB]. It imports the top-level System Functions and sub-functions defined in the [SAB]. Notes can be added from the Right Palette tool to identify the parent System Mission for the functionality. Note: Internal functions/sub-functions are green, external ones are light blue.

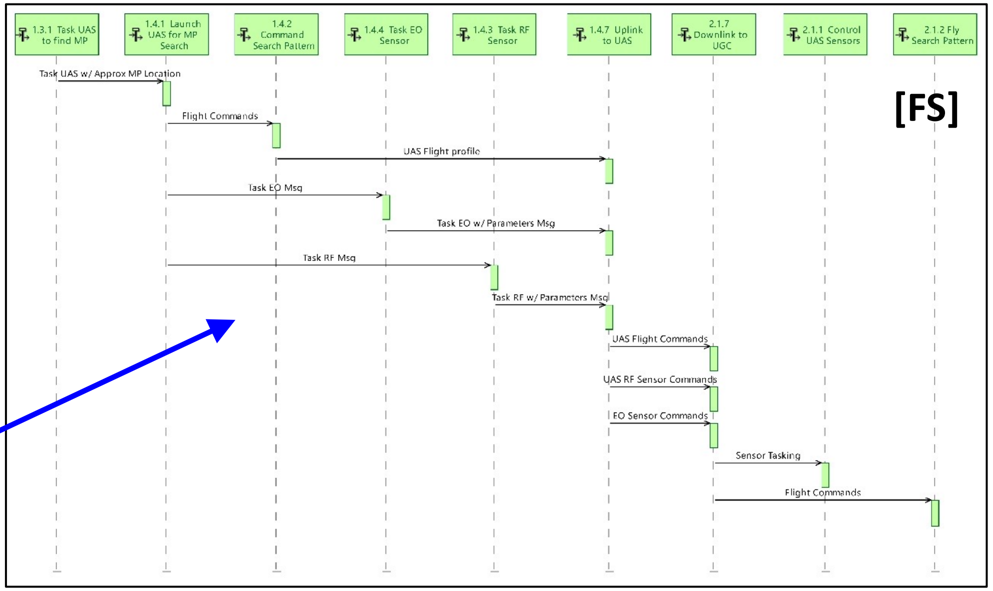
8. [SAB] Imports the System block (dark blue) and external Entity Blocks (light blue) from the [CSA] and spreads them out to allow for internal/external functions and flows to be added. The external top-level System Functions (imported Activities from [OAB] to ADM step 2) are placed in the appropriate External Entity/Actor Block. The internal top-level functions must be decomposed into lower-level system subfunctions before the subfunctions are added into the system block. Data exchanges are added between the internal sub-functions, between the system and external blocks, and between the external blocks (if desired). **Note**: In many Capella examples, the [SDFB] is used to first define the Sub-functions and data exchanges. This was removed for redundancy in this method.

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

2. Systems Analysis – Views (4 of 5)



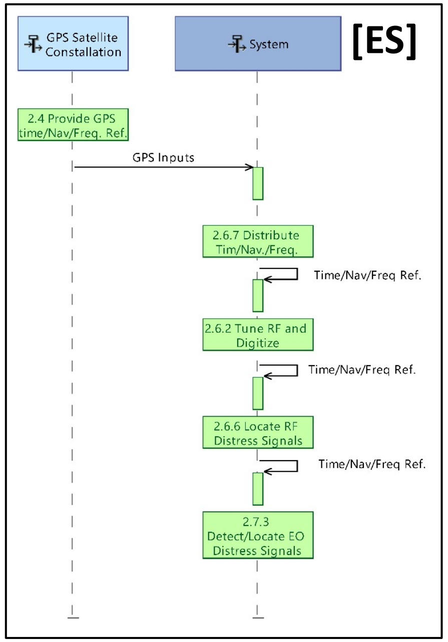




10. Functional threads can be added to applicable portions of the [SAB] by defining the path for each thread (functions and exchanges) and creating an intermediate [SFCD] which extracts the selected thread. The [SFCD] can be selected and transitioned directly into and [FS].

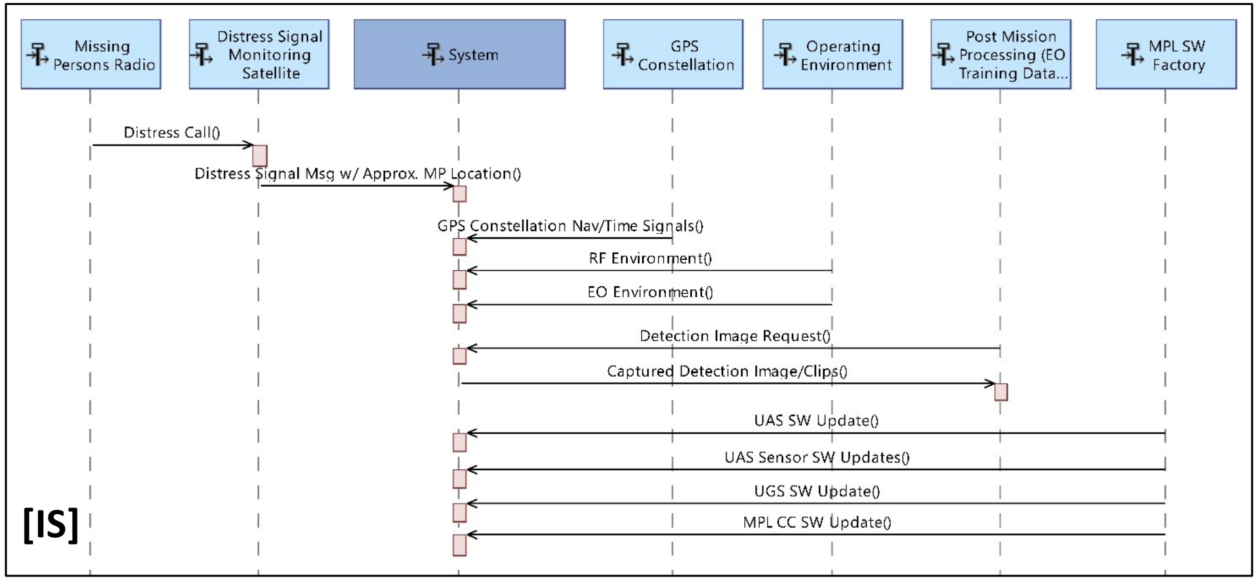
11. [ES] is created from the activity explorer. Since previous artifacts defined all System Elements, they can be pulled in directly from the Right Tool Palette (first system/external entities, then functions and then data exchanges). The tool will only offer adding exchanges between the 2 selected entities

Note: [FS] and [ES] diagrams must be linked to an Operational Capability (imported from ADM Step 1) and will show up in the Capability's directory in the Project Explorer



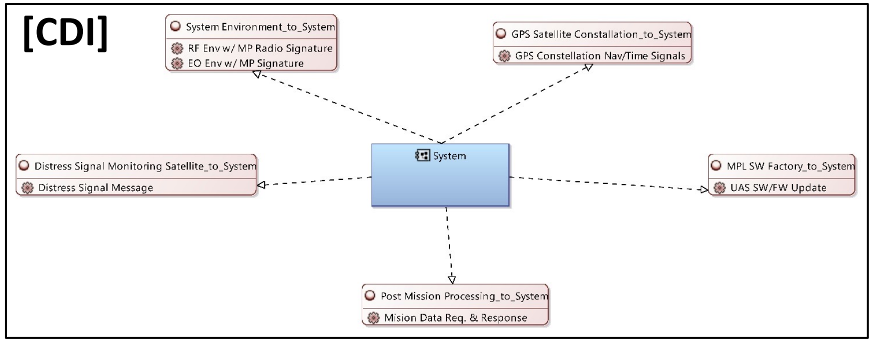
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

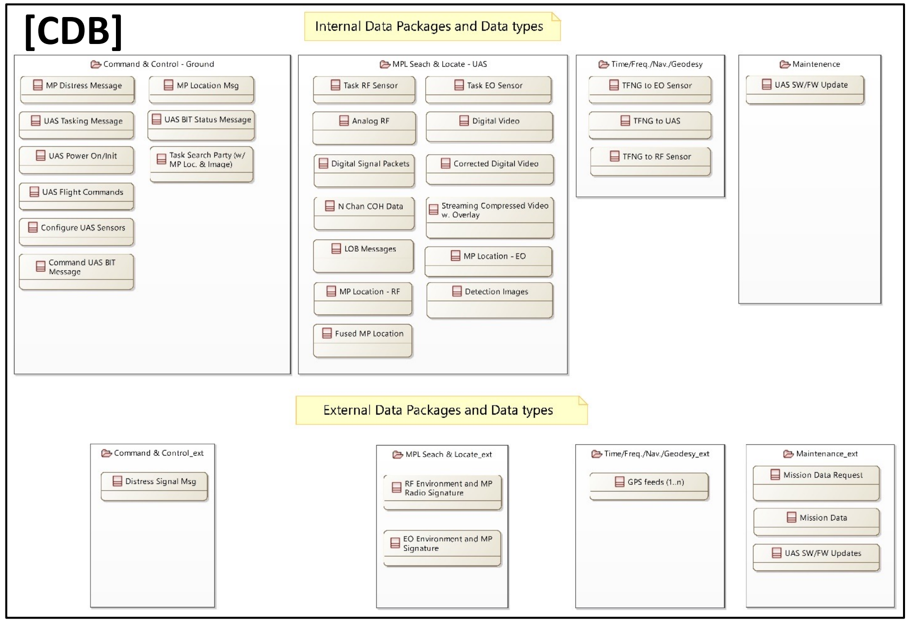
2. Systems Analysis – Views (5 of 5)



12. [IS] is created from the activity explorer. Since previous Internal/External elements were defined in the [CSA] and [SAB], they can be pulled in directly from the Right Tool Palette (first system, then external entities). <u>The EXTERNAL data exchanges can not be pulled from the [SAB] and need to be entered manually.</u> **Note**: The created External interfaces are shown in the interface directory. They are listed individually, and as sorted by interface pairs.

13. [CDI] is auto-generated from the [IS] – Optional

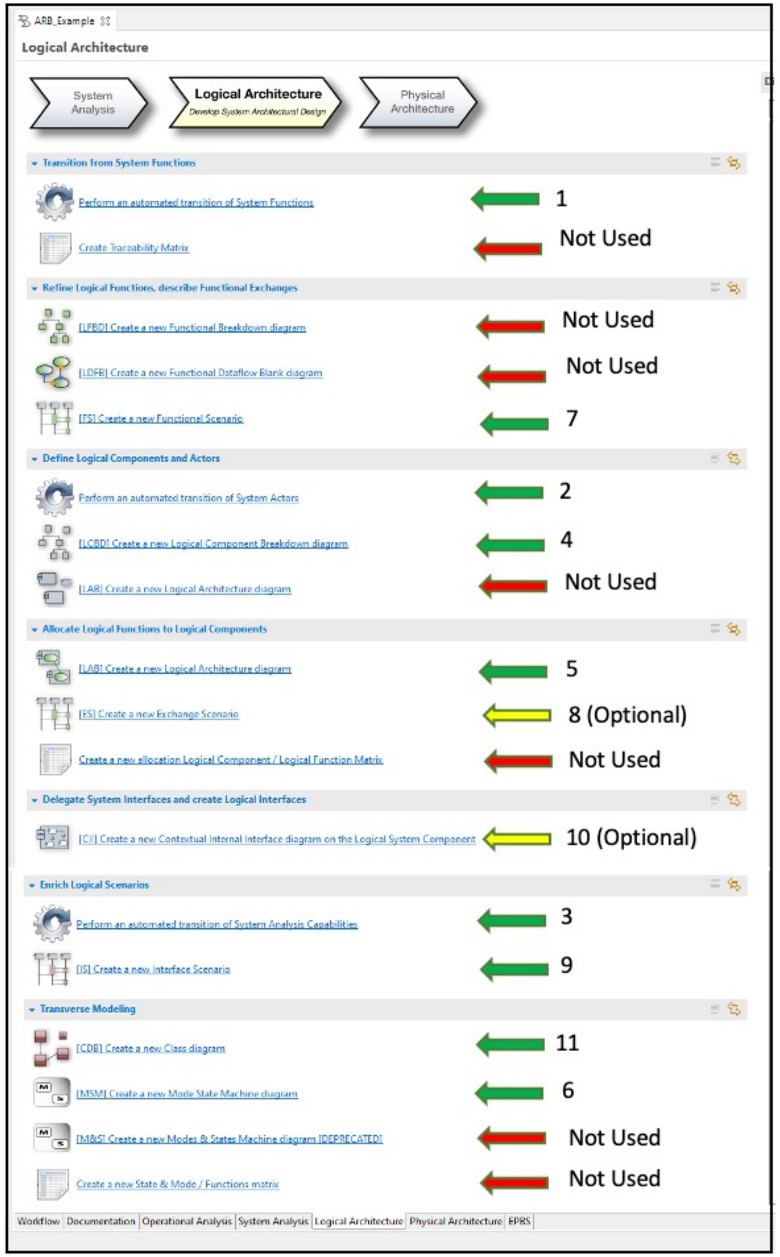




14. General Data Classes and Packages can be sorted into internal and external and internal messages

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

3. Logical Architecture

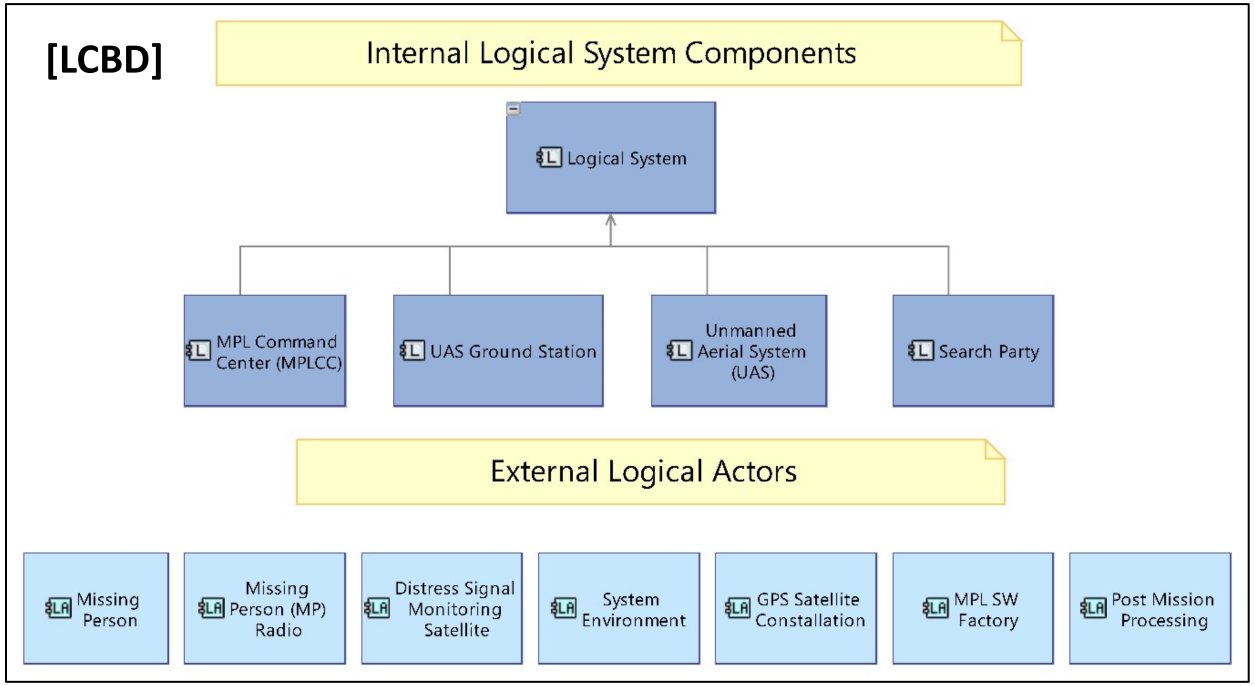


- Imports and Transfers artifacts from System Analysis to Logical Architecture
  - Step 1) Converts on import: System Functions → Logical Function
  - Step 2) Converts on import: External System Actors/Entities → Logical Actors/Entities
  - Step 3) Imports System Capabilities from ADM step 2 into Logical Capabilities in ADM Step 3
- Logical Architecture level decisions made via Trades, Analyses, M&S, and multiple system thinking tools
- Defines the system's logical sub-system and logical component boundaries (i.e. Glass Box view) based on the high-level allocation and aggregation of the previously defined system functions (now logical functions)
  - Allows for derivation of architectural alternatives
- Defines flows between all internal logical subsystems/components and defines the internal interfaces (updates to external interfaces as required.
- Logical Architecture is defined
- Allows for creating process, sequence and activity diagrams to ensure customer ConOps are met and agreed to → SFR

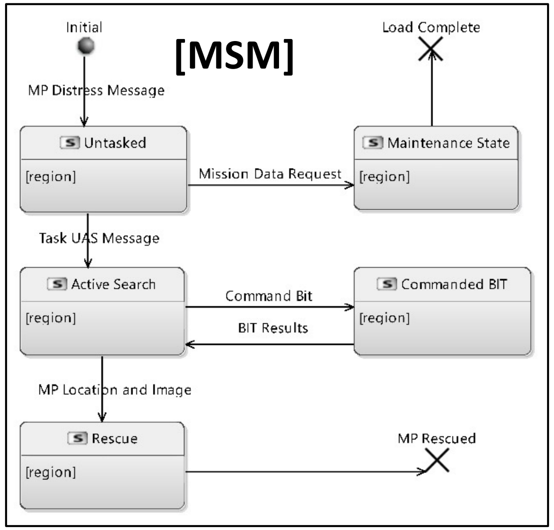
Logical Architecture - View Acronyms (in order used)

[LCBD]	Logical Component Break Down Diagram – Partially Auto-generated
[LAB]	Logical Architecture (Blank) Diagram
[MSM]	Mode State Machine
[LFCD]	Logical Function Chain Diagram – Intermediate product for [FS]
[FS]	Functional Scenario – One for each Use Case/System Capability
[ES]	Exchange Scenario – OPTIONAL
[IS]	Interface Scenario
[CII]	Contextual Internal Interface Diagram – OPTIONAL
[CDB]	Class Diagram (Blank)

3. Logical Architecture – View (1 of 4)

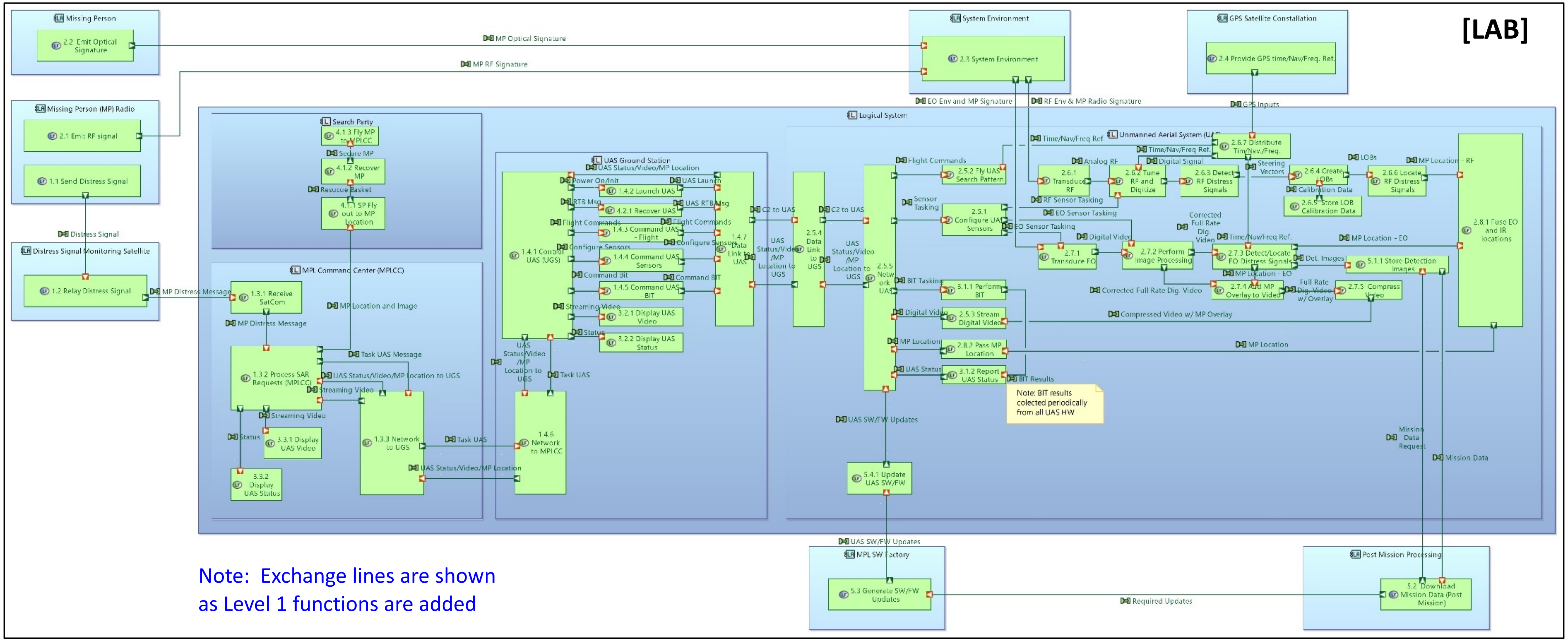


4. [LCBD] is initially autogenerated with a dark blue "Logical System" block. This is where the system internal boundaries get initially defined, by creating internal system elements using the Right Palette tool and linking them to the Logical SystemBlock. The result will be a Logical system breakdown diagram showing internal system entities in dark blue boxes. External logical Actors/Entities will also be shown in light blue boxes. The note tool can be used to organize the blocks.



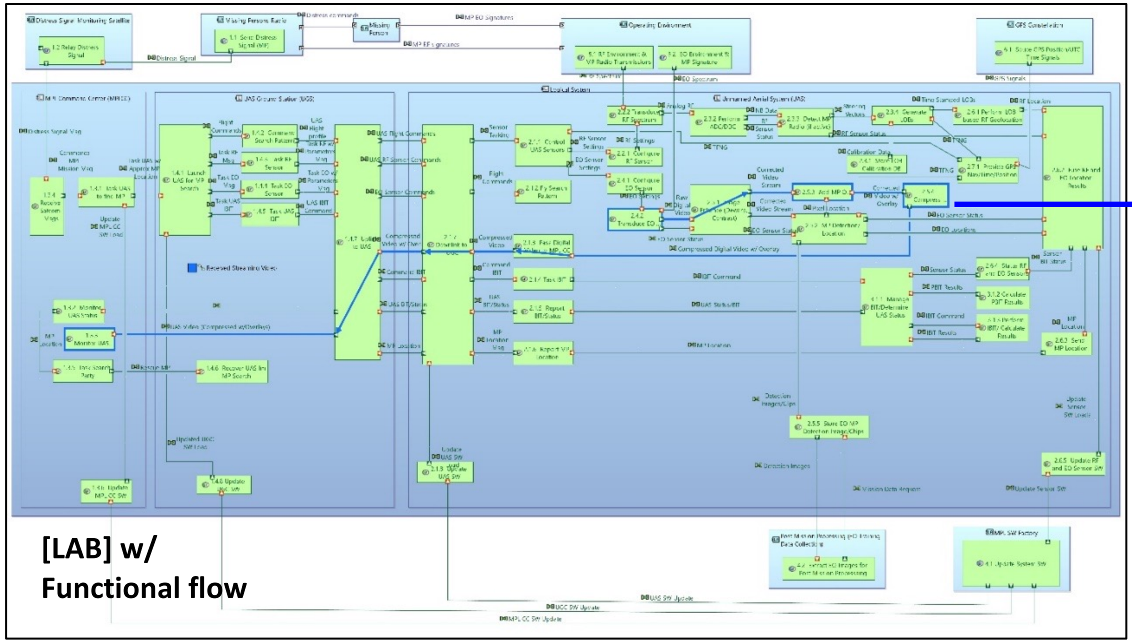
5. [MSM] will define the system state/modes based on the earlier Supplemental views and use cases. Note: Can do at any ADM step.

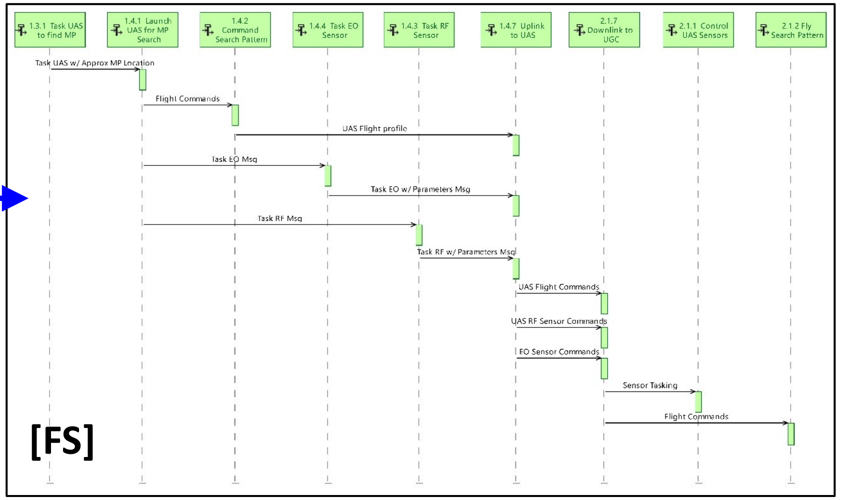
3. Logical Architecture – View (2 of 4)



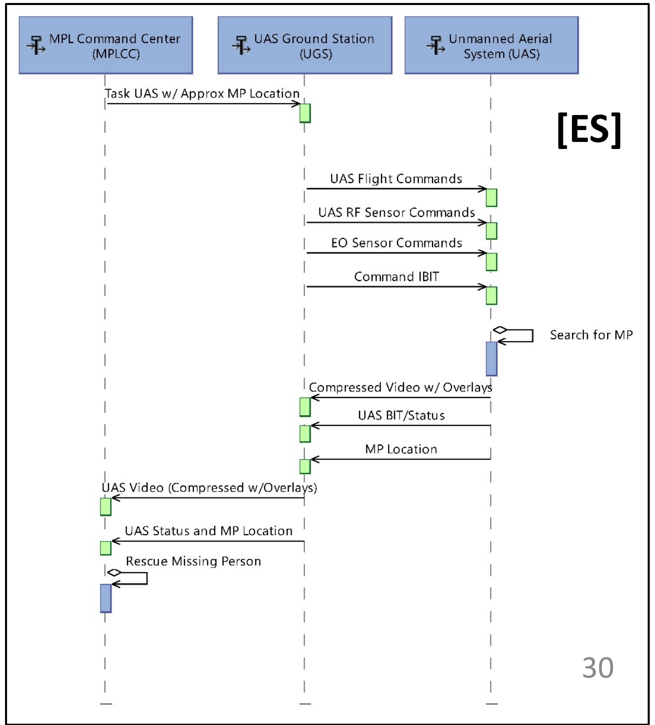
5. [LAB] Imports the System block (dark blue) with defined internal sub-systems and Components, and external Entity Blocks (light blue) from the [LCBD] and reallocates and aggregates the system functions to the internal system elements. Data flows between the internal sub-functions are added/updates as necessary. **Note**: In many Capella examples, the [LDFB] is used to first define the Sub-functions and data exchanges. This was removed for redundancy in this method. Similarly, the auto-generated [LFBD] showing the same functional hierarchy as [SFBD] was not deem required.

3. Logical Architecture – View (3 of 4)





Define all System Capabilities/Threads

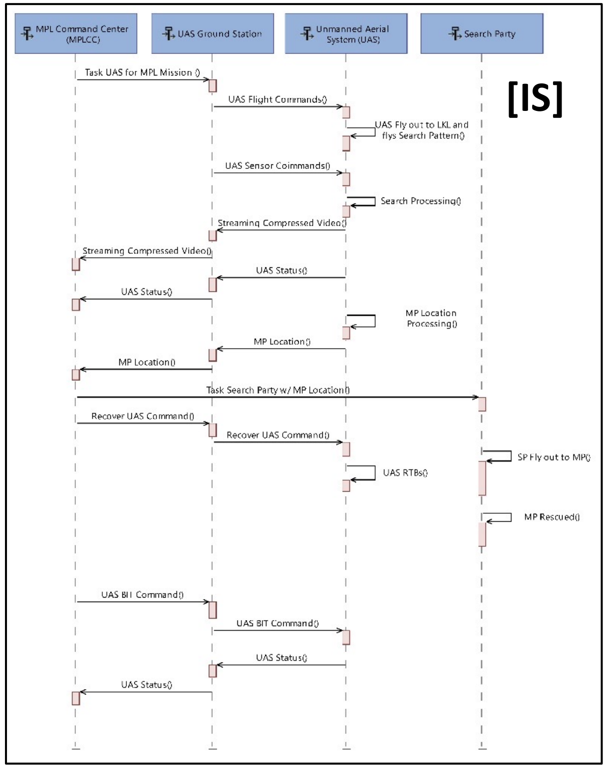


7. Logical threads can be added to applicable portions of the [LAB] by defining the path for each thread (subfunctions and exchanges) and creating an intermediate [SFCD] which extracts the selected thread. The [LFCD] can be selected and transitioned directly into and [FS].

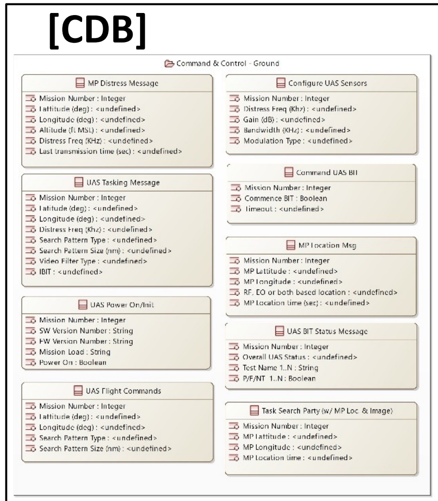
8. [ES] is created from the activity explorer. Since previous artifacts defined all Internal System Elements, they can be pulled in directly from the Right Tool Palette. The tool will only offer adding exchanges between the 2 selected entities

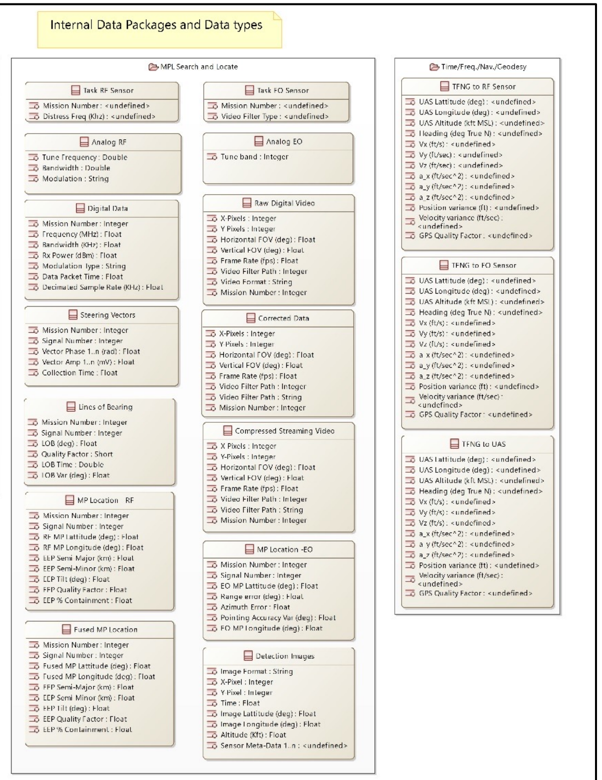
Note: [FS] and [ES] diagrams must be linked to an Operational Capability (imported from ADM Step 2) and will show up in the Capability's directory in the Project Explorer

3. Logical Architecture – View (4 of 4)

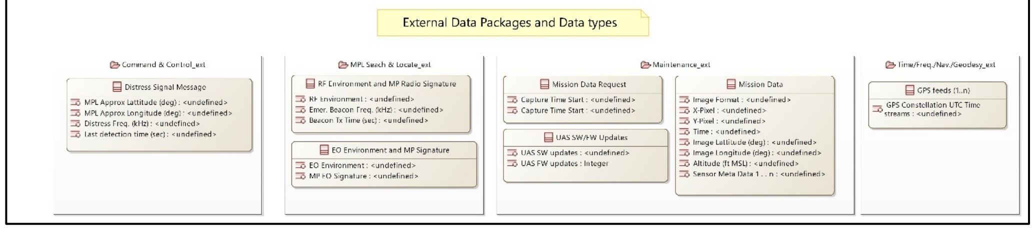


9. [IS] is created from the activity explorer. Since the system subelements were defined in the [LCBD] and [LAB], they can be pulled in directly from the Right Tool). <u>The INTERNAL data exchanges can not be pulled from the [LAB] and need to be entered manually.</u> **Note**: The created Internal interfaces are shown in the interface directory.

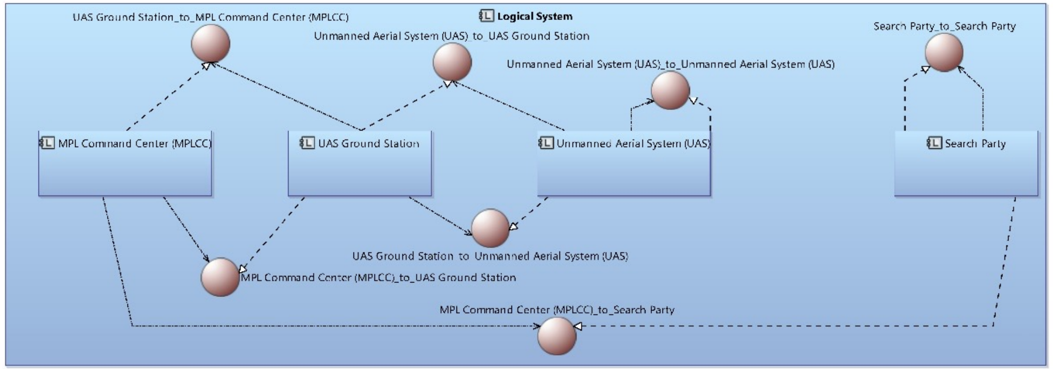




11. Data Classes expanded with properties for external and internal messages

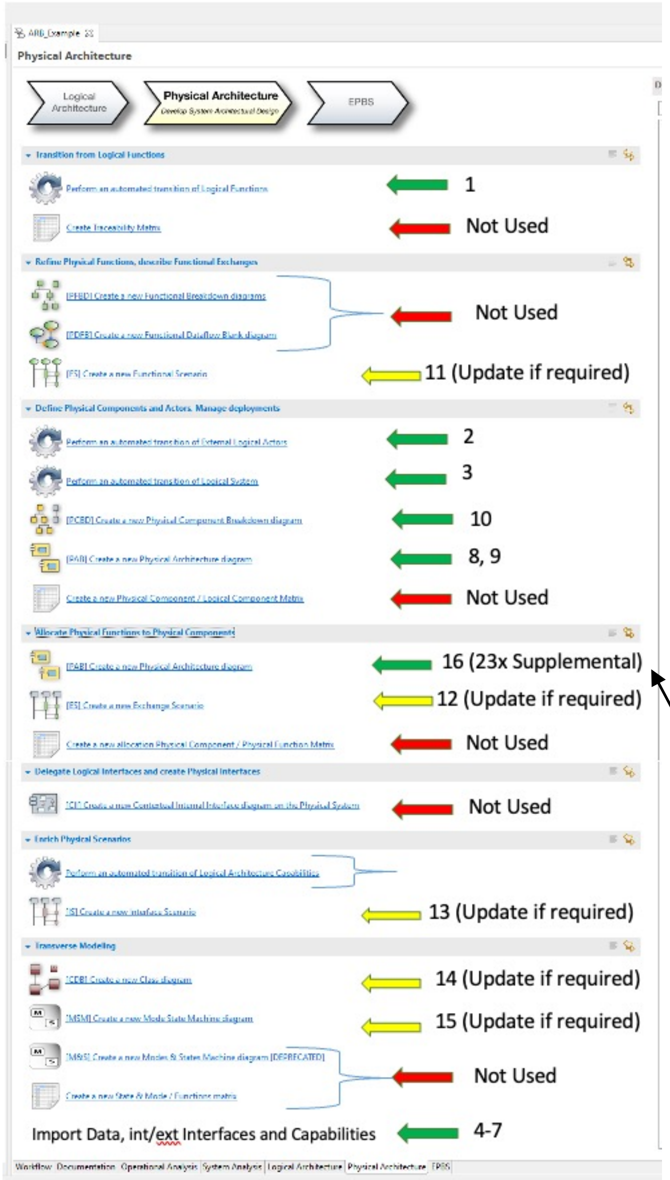


10. [CII] is auto-generated from the [IS] – Optional



THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

4. Physical Arch.



- Imports and Transfers artifacts from Logical Architecture to Physical Architecture
  - Step 1) Converts on import: Logical Function → Physical Function
  - Step 2) Converts on import: Logical Actors/Entities → Physical Actors/Entities
  - Step 3) Converts on import: Logical (internal) System Components → Physical System Components
  - Step 4) Transfers Logical Capabilities from ADM step 2 into Physical Capabilities in ADM Step 3
  - Step 5) Transfers Logical Data model from ADM step 2 to Physical Data Model in ADM Step 3
  - Step 6) Transfers Logical External Interfaces from ADM step 2 to Physical External Interfaces in ADM Step 3
  - Step 7) Transfers Logical Internal Interfaces from ADM step 2 to Physical Internal Interfaces in ADM Step 3
- Physical Architecture level decisions made via Trades, Analyses, M&S, and multiple system thinking tools
- Performs final low-level Allocation to SW/HW/FW, Aggregation of functions, and Partitions all Logical components and associated functions to physical hardware. defines all interfaces and flows in/out of the system (White Box view)
  - Allows for derivation of architectural alternatives and defines the systems properties (especially modularity)
- Defines interaction with system Actors/elements
- Defines final data architecture and external/internal interfaces
- Defines flows between all internal physical elements
- Physical Architecture is defined

"Illities" Related
- 11.1 "illities" POCs
- 11.2 Cyber/Security
- 11.3 Reliability
- 11.4 Maintainability
- 11.5 Availability
- 11.6 Safety
- 11.7 Integrated Logistics Support - Training
- 11.8 Testability
- 11.9 Producibility/Manufacturability/AS-9145
- 11.10 Additive Manufacturing (if applicable)
- 11.11 Affordability
- 11.12 Interoperability

"Illities" Related (cont.)
- 11.13 Mission Planning (if applicable)
- 11.14 Post Mission processing (if applicable)
- 11.15 Mission Simulator (if applicable)
- 11.16 Modular Open System Approach (MOSA)
- 11.17 Flight Certification (if applicable)

Additional Architectural Artifacts
- 11.18 Architectural Trades
- 11.19 Architectural Analyses
- 11.20 Preliminary Requirements Compliance
- 11.21 Architecture Risks
- 11.22 Architecture Opportunities
- 11.23 System Qualification Roadmap

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Physical Architecture - View Acronyms (in order used)

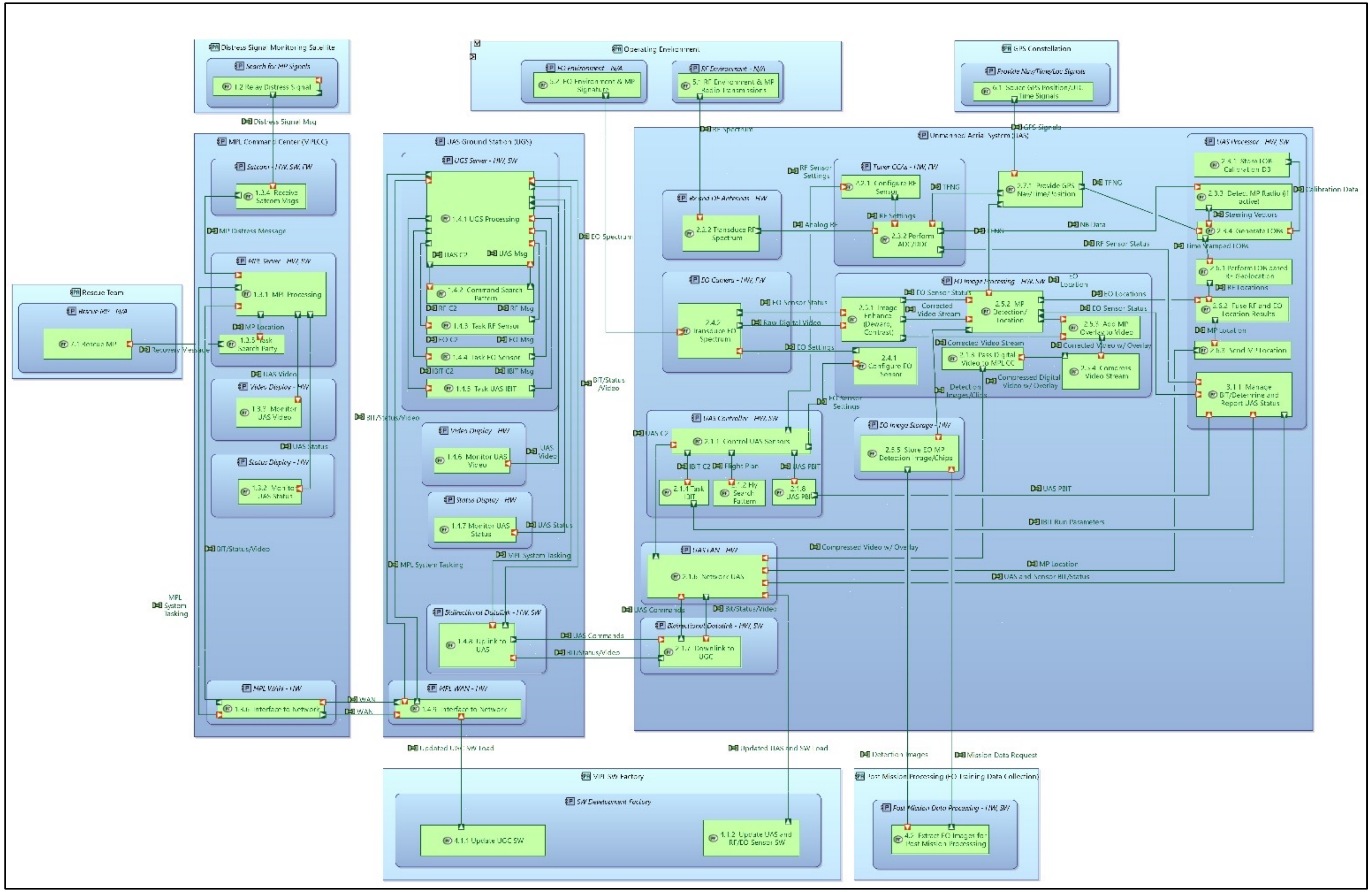
[PAB]	Physical Architecture (Blank) Diagram
[PCBD]	Physical Component Block Diagram

Updated if Necessary and included in the Physical Architecture

[PFCD]	Physical Function Chain Diagram – Intermediate product for [FS]
[FS]	Functional Scenario – OPTIONAL
[ES]	Exchange Scenario – OPTIONAL
[IS]	Interface Scenario (internal)
[IS]	Interface Scenario (external)
[CDB]	Class Diagram (Blank)
[MSM]	Mode State Machine

4. Physical Architecture – View (1 of 3)

Done for each Alternative Architecture



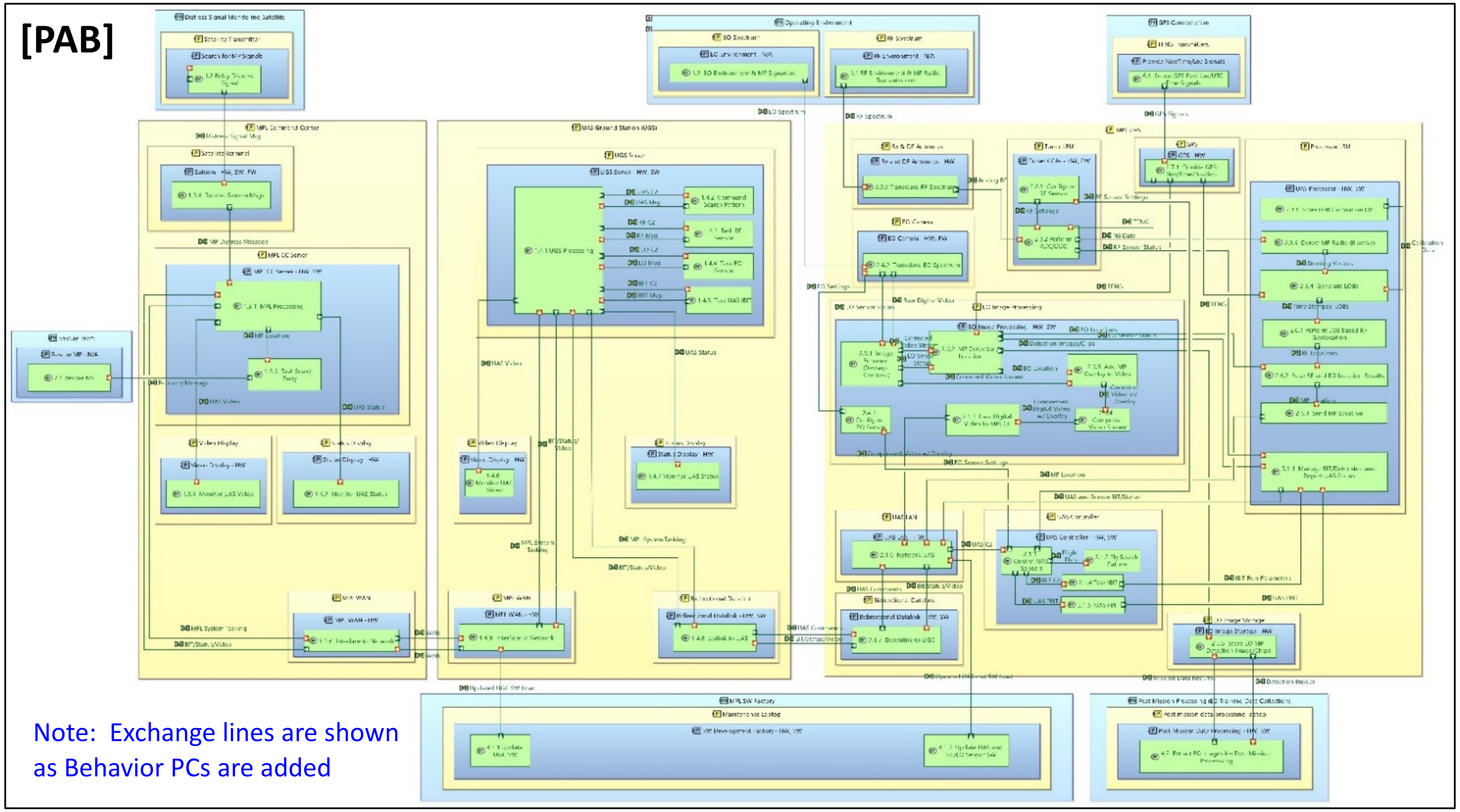
8. [PAB] is created from the activity explorer. The diagram is used to create the lower-level Behavior Physical Components (PCs) that will detail the low-level allocation and aggregation of the functions.

The Level 0 Behavior PCs come form the [LCBD], which are imported into ADM step 4. The Level 1 Behavioral PCs are contained in their associated Level 0 Behavioral PC and define the allocations for all single or aggregated functions they contain

Note: By establishing the Behavior PCs first, they can be pulled into Physical Node Components (with all included functions) easily in the next step.

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

4. Physical Architecture – View (2 of 3)



9. [PAB] is created from the activity explorer. The diagram is used to create the Physical Node Components (PNCs) that detail the low-level partitioning of the Behavior PCs (with aggregated functions) from the last step.

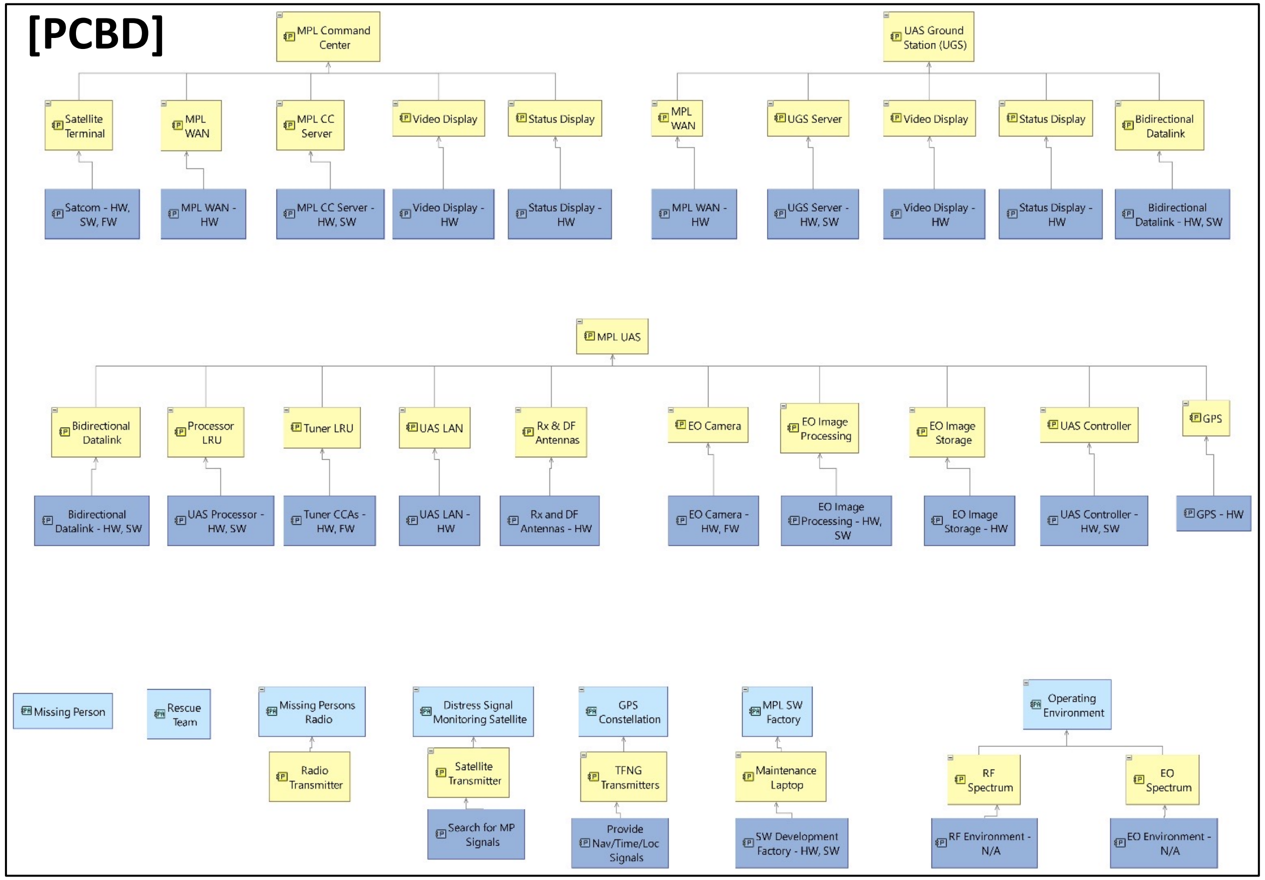
The Level 0/1 PNCs are derived from the Level 0/1 Behavior PCs (Note: modularity heuristics dictate a nominal 1:1 mapping between them)

The physical architecture is then created by placing in the Level 1 Behavioral PCs and then the physical functions. Note: Data flows will be auto-populated as functions get added. Tool will only allow correct placements

Note: Partition are shown in the title of the Level 0 Physical Node Component Blocks

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

4. Physical Architecture – View (3 of 3)



10. [PCBD] is auto-generated from the [PCB]. It imports the top-level (Level 0) Physical Node Components (Yellow blocks) and shows hierarchically their associated Level 1 Physical Node Components (Yellow blocks) and Level 1 Behavior PC (dark Blue blocks)

External entities will also be shown (light Blue blocks) with their associated Level 1 Physical Node Component(s) (Yellow blocks) and Level 1 Behavior PC (dark Blue blocks)

The [PCBD] and directory structure are good way to check the completeness of the model.

Steps 11 – 15 are to update the Data architecture, interface diagrams (internal/external), State Mode diagram as required, or to create [FS] and [PS] diagrams at the [PAB] level if desired.

Physical Architecture - Supplemental Views (step 16)

"Illity" POCs

To ensure a robust architecture is derived the following "illities" inputs will be required:

Domain	POC
Cyber/Security	
Reliability	
Maintainability	
Availability	
Safety	
Integrated Logistics Support - Training	
Testability	
Producibility/Manufacturability/AS-9145	
Additive Manufacturing (if applicable)	
Affordability	
Interoperability	
Mission Planning	
Post Mission processing	
Mission Simulator	
MOSA	
Flight Certification (if applicable)	

Cyber and Security

- This artifact covers all aspects of security including cyber
- Suggested topics include:
  - Identifying critical information and technologies in the system which must be protected
  - Determining required measures to protect the critical information and technologies as specified by security agencies based on system elements
  - Development approach for system HW based security measures (e.g. IDS, IPS)
  - Development approach for system SW (Abuse Cases, Risk Analysis, Static analysis, dynamic analysis, Penetration testing, etc.) and DevSecOps if applicable.
  - Development approach for security related test equipment and associated software (i.e. representative test env.)
  - Security Certification Approach
  - Upgrade/Security patch approach over lifecycle
  - Multi-Level security approach
  - Previous cyber lessons learned
  - Preliminary cyber requirements
  - Cyber/Security POCs comments on baseline architecture/alternatives
  - Cyber/Security artifact timeline

Reliability

- This artifact covers all aspects of the system's availability
- Suggested topics include:
  - Preliminary reliability requirements and predictions for relevant environment (examples below)
    - Mean Time Between Failure (MTBF)
    - Mean Time Between Maintenance (MTBM)
    - Mean Time Between Repair (MTBR)
    - Mean Time Between Critical Failure (MTBCF)
    - Mean Time Between Operational Mission Failure (MTBOMF)
    - Mean Time to Failure (MTTF)
  - Previous reliability lessons learned
  - Reliability growth prediction and assumptions
  - Reliability POC comments on baseline architecture/alternatives
  - Reliability artifact timeline

Maintainability

- This artifact covers all aspects of the system's maintainability including Built In Test (BIT)
- Suggested topics include:
  - Preliminary maintainability requirements and predictions (examples below)
    - Mean Time To Repair (MTTR)
    - Mean Time Between Maintenance (MTBM)
    - Mean Preventative Maintenance Time (MPMT)
    - Mean Active Maintenance Time (MAMT)
    - Mean Down Time (MDT)
    - Max System Installation/Removal Time
    - Percent BIT Fault Detection (Pfd)
    - Percent BIT Fault Isolation (Pfi)
    - Percent BIT False Alarms (Pfa)
  - Previous maintainability lessons learned
  - maintainability requirements
  - Maintainability POC comments on baseline architecture/alternatives
  - Maintainability artifact timeline

Availability

- This artifact covers all aspects of the system's availability. Availability is dependent on both Reliability and Maintainability metrics, as well as the Organizational ConOP (e.g. Maintenance staff, repair equipment, spares, repair manuals, tools, etc.)
- Suggested topics include:
  - Preliminary availability requirements and predictions (examples below)
    - Inherent Availability (Ai) – Only a function of system design (MTBF, MTTR)
    - Operational Availability (Ao) – A function of Design and support (MTBF, MDT)
    - Material Availability (Am) – A function of operational and total deployed system assets
  - Previous availability lessons learned
  - Availability growth prediction and assumptions
  - Availability POC comments on baseline architecture/alternatives
  - Availability artifact timeline

Safety

- This artifact covers all aspects of the system's safety.
- Suggested topics include:
  - Preliminary Safety Hazard Assessment
    - Critical Safety items identified
  - Previous safety lessons learned
  - Preliminary safety requirements
  - Safety POC comments on baseline architecture/alternatives
  - Safety artifact timeline

Integrated Logistics Support – Training (ILS-T)

- This artifact covers all aspects of the system's integrated Logistics Support. ILS is dependent on Reliability, Maintainability and Availability metrics, as well as the Organizational ConOP (e.g. Maintenance staff, repair equipment, spares, repair manuals, tools, etc.)
- Suggested topics include:
  - Logistic footprint defined
  - Sparing concept defined
  - Maintenance concept defined (Organization (O-Level), Intermediate (I-Level) and Depot (D Level))
    - Maintenance ConOp Level of Repair Analysis (LORA) Approach
  - System training scope defined
  - Previous ILS-T lessons learned
  - Preliminary ILS-T requirements
  - ILS-T POC comments on baseline architecture/alternatives
  - ILS-T artifact timeline

Testability

- This artifact covers all aspects of the system's testability.
- Suggested topics include:
  - Testability assessment (Includes Development, Production and Sustainment debug and test elements or common elements for all phases)
  - Capital Assets required for system test
  - System test points strategy for factory measurement/BIT
  - Previous testability lessons learned
  - Preliminary testability requirements
  - Test Engineering POC comments on baseline architecture/alternatives
  - Test Engineering artifact timeline

Producibility/Manufacturability/AS-9145

- This artifact covers all aspects of the system's producibility, as well as compliance to the AS-4195 standard.
- Suggested topics include:
  - Defined Leverage/Reuse elements (HW, SW and FW)
    - Architecture and design packages for all leverage/reuse (system and test equipment)
    - MFA, DFMEA and PFMEA for all leverage/reuse (system and test equipment)
  - Manufacturing Feasibility Assessment (MFA)
  - Architecture level DFMEA and risks
  - Architecture level PFMEA and risks
  - Architecture suitability for production
  - Identified factory production assets required
  - Identified factory calibrations/measurements
  - Previous production lessons learned
  - Preliminary production and production test requirements
  - Operations POC comments on baseline architecture/alternatives
  - Operations artifact timeline

Additive Manufacturing (AM) (if applicable)

- This artifact covers all aspects of the system's additive manufacturing
- Suggested topics include:
  - AM uses for early prototyping (fit checks, ergonomics, cable routing, customer feedback, etc.)
  - AM used to address part reduction/consolidation in a multi-piece assembly
  - AM used to help with weight reduction
  - AM used to help with potential thermal/heat transfer issues
  - AM used in brazed assemblies to reduce dependency on braze suppliers
  - Previous AM lessons learned
  - Preliminary AM requirements
  - AM POC comments on baseline architecture/alternatives
  - AM artifact timeline

Affordability

- This artifact covers all aspects of the system's affordability.
- Suggested topics include:
  - Architecture assessment to allocated Design to Unit Production Cost (DTUPC)
    - DTUPC cost allocations to architectural elements
    - Cost break down of architecture (material, labor, calibration, etc.)
    - Cost drivers defined (recurring system and system production)
  - Architecture cost assessed over the system lifecycle
    - Life Cycle Cost (LCC) assumptions defined (e.g. #LRIP, FRP, system lifecycle duration, etc.)
  - Previous affordability lessons learned
  - Preliminary affordability allocations
  - LCC POC comments on baseline architecture/alternatives
  - LCC artifact timeline

Interoperability

- This artifact covers both intra-system and inter-system interoperability
  - Suggested topics include:
    - Use of intra- and inter-system Open Standard Interfaces (HW, SW, FW)
    - Receive over power monitoring (e.g. Limiters, Detectors, ADC Overflow, Lock-outs, Data invalid flag)
    - Manual/Automatic Gain Control (MGC, AGC, NAT)
    - Receive side blanking (e.g. Physical blank lines, Tx Messages, Notch Filters)
    - Transmit side muting (e.g. Physical blank lines, TX override)
    - Receive/transmit interleaving - IB and OOB (Look-Thrus, Rx/Tx priority rules engine)
    - Harmonic mitigation (e.g. Tx filters, Rx Filters)
    - Mixing product mitigation (e.g. Up/Downconversion filtering/mixing approach, component OIP)
    - Interference Cancellation
    - Previous interoperability lessons learned
    - Preliminary interoperability requirements
    - Interoperability POC comments on baseline architecture/alternatives
    - Interoperability artifact timeline

Mix of "illities," Quality Attributes, Verification and Risks to ensure a Robust architecture was defined.

<u>All Domains should be involved from the Conceptual Architecture and on</u>

Physical Architecture - Supplemental Views (step 16)

Mission Planning (if applicable)

- This artifact covers all aspects of the system's mission planning, including generation of operational system files, mission configurable files, and parameters
- Suggested topics include:
  - Mission Planning system (e.g. JMPS, map server)
  - Mission Files (Operational software and mission specific files and parameters)
  - Mission File Media (media used to transfer files, encryption approach, etc.)
  - Previous mission planning lessons learned
  - Preliminary mission planning requirements
    - Route planning
    - Route optimization
    - Threat assessment
    - Preliminary tasking
  - Mission planning POC comments on baseline architecture/alternatives
  - Mission planning artifact timeline

Post Mission Analysis (if applicable)

- This artifact covers all aspects of the system's mission planning, including generation of operational system files, mission configurable files, and parameters
- Suggested topics include:
  - Required Data for post mission analysis
    - BIT/Diagnostics
    - Mission Data
    - Training Data (for ML engines)
  - Post-mission file media (media used to transfer files, encryption approach, etc.)
  - Post-mission analysis system (e.g. JMPS, map server)
  - Previous post mission data analysis lessons learned
  - Preliminary post mission data analysis requirements
    - Signal Processing
    - Image Processing/ML
    - Prognostics metrics generation
  - Post-mission analysis POC comments on baseline architecture/alternatives
  - Post-mission analysis artifact timeline

Simulator (if applicable)

- This artifact covers all aspects of the system's mission simulator, including generation of simulator system files, simulator mission configurable files, and simulator parameters
- Suggested topics include:
  - Assessment for system and simulator commonality (SW, HW, FW)
  - Post-mission File Media (media used to transfer files, encryption approach, etc.)
  - Post-mission Analysis system (e.g. JMPS, map server)
  - Previous simulator lessons learned
  - Preliminary post mission data analysis requirements
    - Signal Processing
    - Image Processing/ML
    - Prognostics metrics generation
  - Simulator POC comments on baseline architecture/alternatives
  - Simulator artifact timeline

MOSA Assessments

- This artifact covers all aspects of the system's MOSA assessment
- Suggested topics include:
  - Business enabling environment description
  - Program Assessment and Rating Tool (PART) and/or Open Architecture Assessment Tool (OAAT) results
    - Business and Technical scoring
    - Areas of improvement
  - Key Open SubSystem (KOSS) Tool results
    - Key interfaces identified
    - Rapidly changing/obsolesce elements identified and mapped against user utility
    - Enhancements to modularity based on findings
    - Key interface mitigations using open standards
      - Technology roadmaps/forecasts
    - Key interface mitigations with no open standard currently available
  - Previous MOSA lessons learned
  - Preliminary MOSA requirements
  - MOSA POC comments on baseline architecture/alternatives
  - MOSA artifact timeline

Flight Certification (if applicable)

- This artifact covers all aspects of security including cyber.
- Suggested topics include:
  - Define ODA and key staff for certification
  - Review of any proposed designs and methods that will be used to show that the system designs and the overall airplane complies with FAA regulations (e.g. DO-160, DO, etc.)
  - Planned ground tests and flight tests to demonstrate that the airplane operates safely
  - An evaluation of the airplane's required maintenance and operational suitability for introduction of the airplane into service
  - Previous Flight Certification lessons learned
  - Preliminary Flight Certification requirements
  - Flight Certification POC comments on baseline architecture/alternatives
  - Flight Certification artifact timeline

Or any other Cert

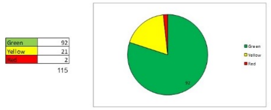
MPL – Trade Studies

Trade	Option 1	Option 2	Selection and Rationale
Single vs. Multiple UASs	Single Large UAs	Multiple Small UASs	**Single Large UAV wa selected**: • Longer loiter time and search range. • High availability and large MTBFA • Greater payload capacity • Can fly higher/Search more area/time
Location Algorithm	Inter-ferometry	Time/ Frequency Direction of Arrival (T/FDOA)	**Interferometry was selected**: • Minimize # antennas installed (2) • Distress signal is narrowband and at a known frequency • Platform will get required transverse bearing spread when flying search profile
UAS Tethering	**Ground Station**	**Satellite**	**The UAVs will be controlled over a ground link:** • The UAS altitude keeps it LOS with the UAS Ground Station • Reduced cost and required payload equipment

MPL - Analyses

Analysis	Method	Tool	Summary Findings
Geo-location Accuracy	Cramer Rao Bound	Matlab	• A Cramer Rao Lower bound tool was created to assess 2 and 3-ship T/FDOA geolocation accuracy obtained as a function of frequency, platform spacing, platform relative heading, platform sensor accuracy and system collection integration time • Model results were found to be consistent with actual measurements/ configurations
Resilience Feature Discovery	Resilience FMECA	Excel	• A FMECA was performed from a resilience attribute point of view • Multiple resilience attributes were assessed • This approach produced some new resilience attributes such as battery power derating and the ability to autonomously fly UAVs back to the base when at low power
UAV Swarm T/FOA Resiliency	Invariant Contracts	• Matlab	• Matlab simulations of the SAR capability were used to quantitatively define the resilience of a 3-ship geolocation swarm when the master is lost, using invariant contracts. • The Recovery Time (Master Lost) is bound as [3.5 13.5] minutes for the stated run parameters. • Recovery time was driven by point at where the Master was lost.

Requirements Compliance



- Yellow requirements – Description and mitigation strategy
- Red requirements – Description and mitigation strategy

Architecture Risks

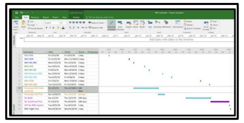
#	Risks	Likelihood (L, M, H)	Impact to Baseline	Mitigation Strategy
1				
2				
3				
4				
5				
6				

Architecture Opportunities

#	Opportunity	Likelihood (L, M, H)	Impact to Baseline	Incorporation Strategy
1				
2				
3				
4				
5				
6				

System Qualification and Test Strategy

- This artifact covers all aspects of security including cyber.
- Suggested topics include:
  - Subsystem test configurations
  - I&T events/roadmap to build up System from Subsystem elements and perform qualification.
  - Location of test events
    - Key test equipment (anechoic chambers, RF ranges, etc.)
  - TRL after each test event
  - Plan for hard to test requirements



Mix of "illities," Quality Attributes, Verification and Risks to ensure a Robust architecture was defined.

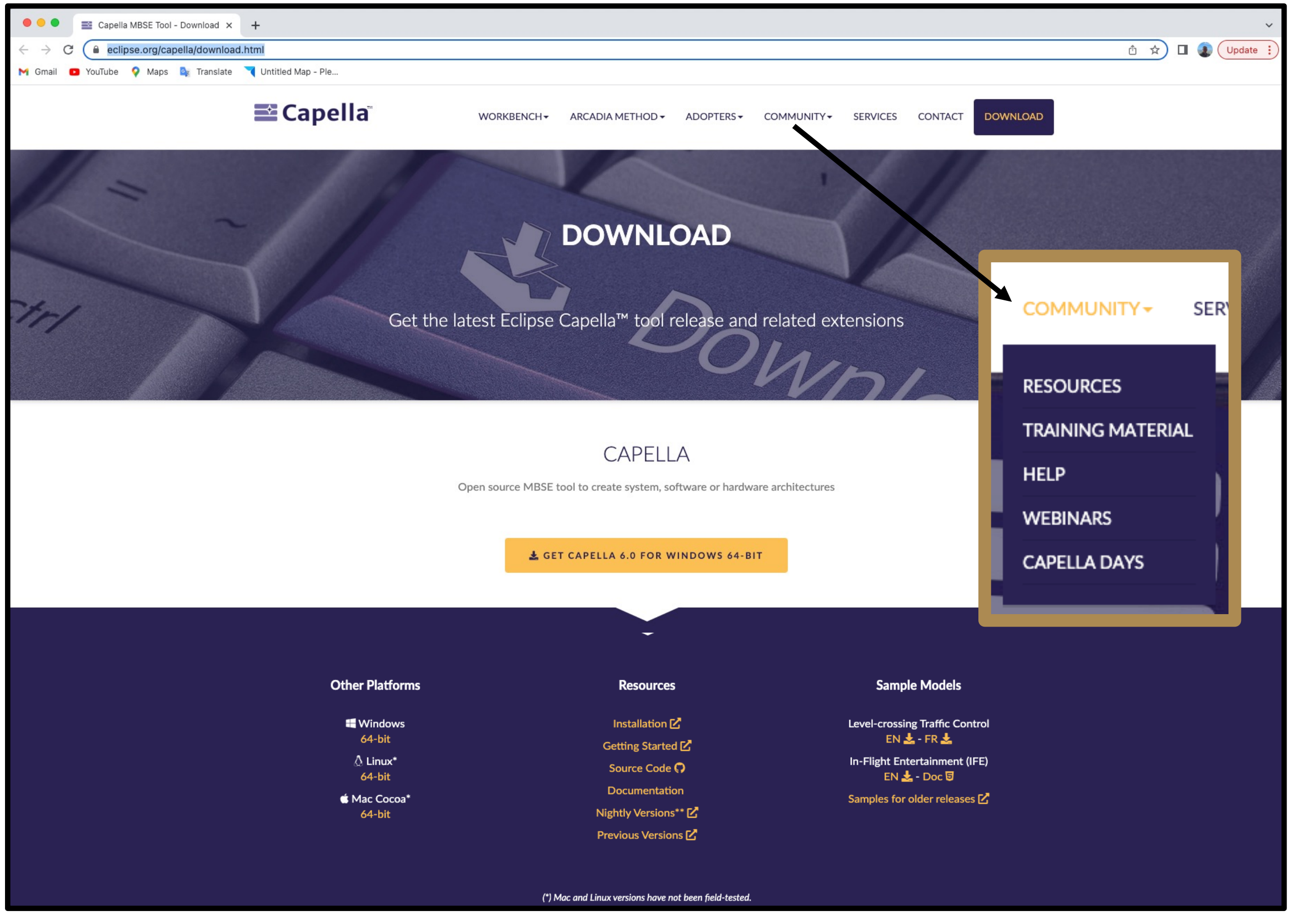
<u>All Domains should be involved from the Conceptual Architecture and on</u>

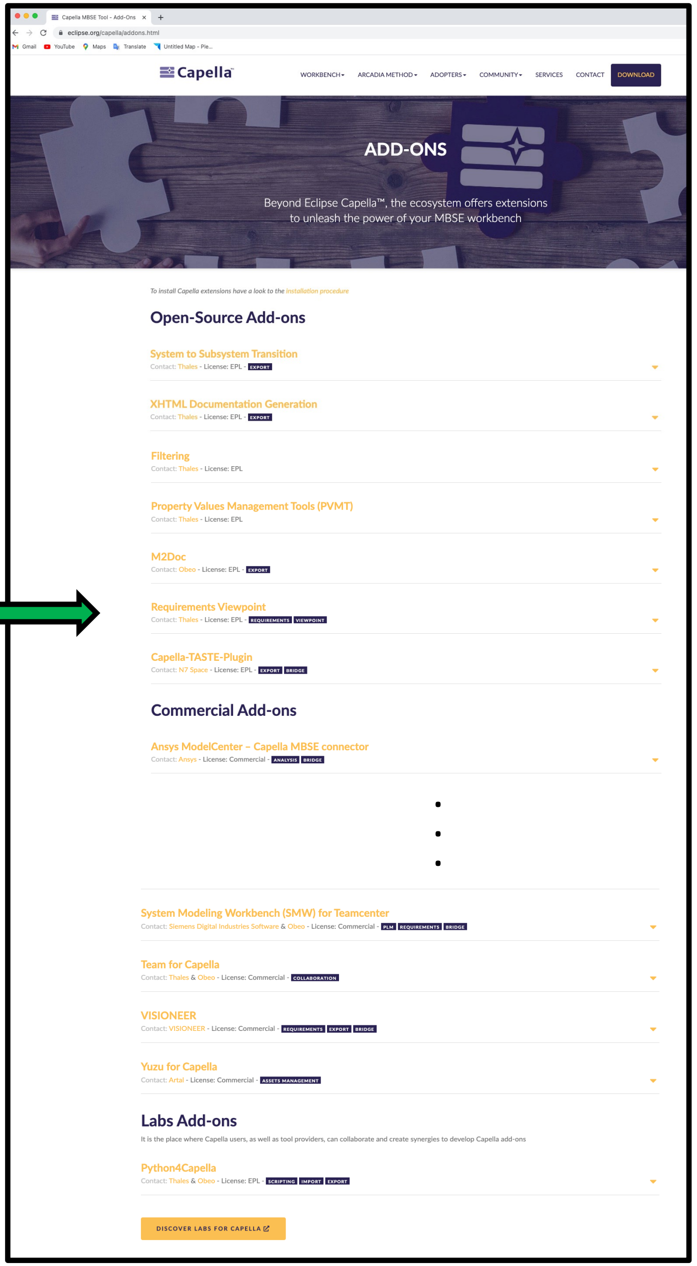
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Capella Demo
(Time Permitting)

Arcadia Capella Download

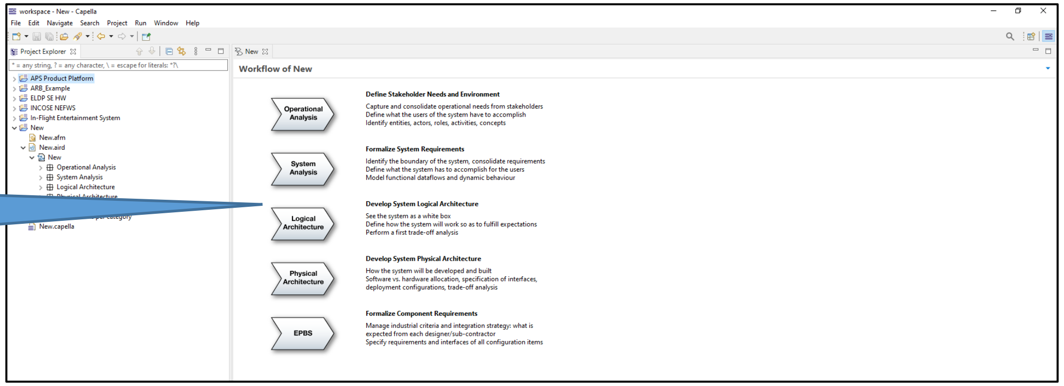
- https://www.eclipse.org/capella/download.html





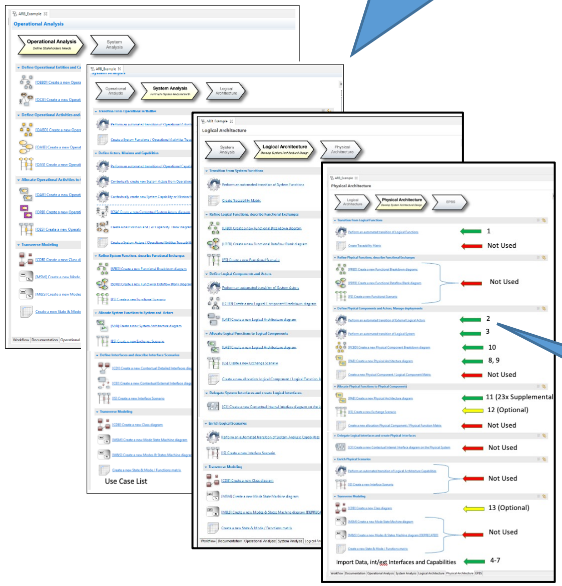
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Arcadia Capella Demos



Activity Explorer guides users though the 4 step ADM and facilitates transferring data between each layer.

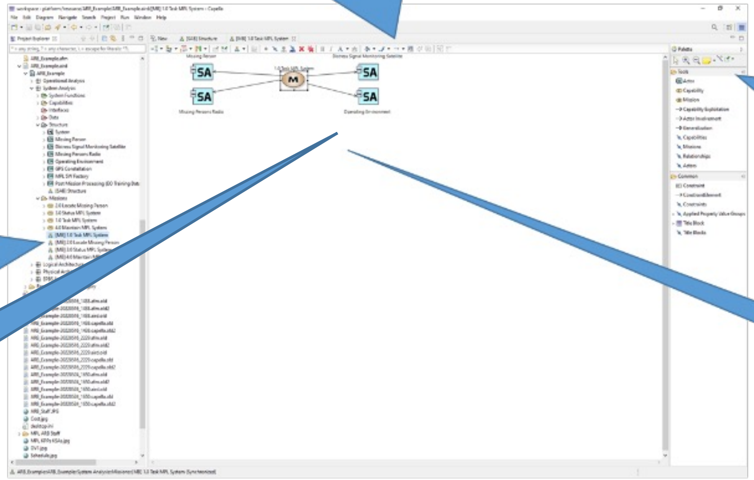
Selecting any layer brings up the methodology for each ADM step.



Main Menu – Artifact creation utilities

Right Palette - Create new artifact elements, or import existing model elements elements

Project Explorer – Directory of each architecture layer and artifacts in each layer

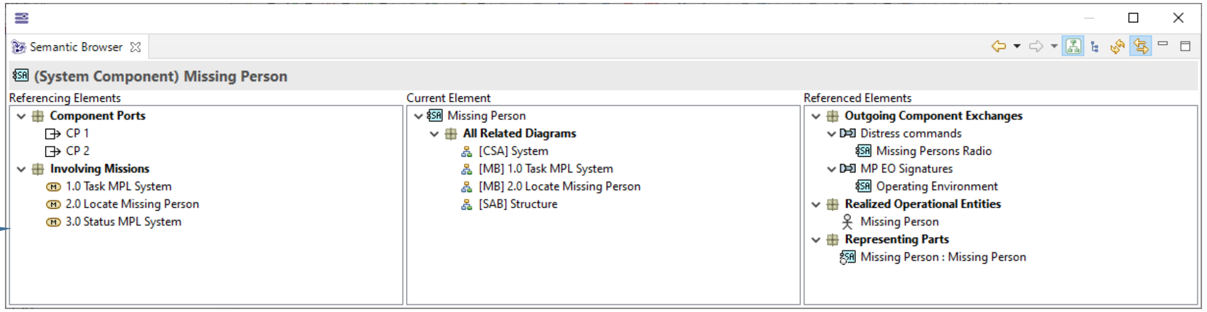


Artifact Tab – new tab is brought up when selected from methodology artifact list

Artifact Tab – Use as workspace to create a new artifact

The semantic browser provides detailed information on artifact elements: Port definition, Linked Missions, Views artifact is in, interactions with other elements.

Excellent debugging tool



Arcadia Capella Familiarization Demo

- Overview of Capella structure
- Artifacts from each of the 4 ADM steps
  - Functional decomposition Operational Capabilities to Physical Functions
  - General element addition and line tool
  - Architecture outputs from each ADM step [OAB], [SAB], [LAB], [PAB]
- Features
  - Ease of generating Functional Scenarios [FS] (a.k.a. Sequence Diagrams)
  - Tool ability to only allow valid choices when placing flow lines
  - Capturing Documents to *.jpgs
  - Auto generation of hierarchical documents

Conclusion

Proposed Method - Conclusions

- **ARCADIA Capella provides an excellent core to model base architecting, since it was created for that purpose**
  - Strongly Covers the 4 steps of 15288:2015
  - Has requirements and design/development add-ons
  - Everything Downloadable via Eclipse Website (True open source)
  - Many papers on using Capella outputs to "Illities" Models → MBE
- **Defining the views and order views are performed further reduce modeling time without giving up fidelity**
- **Supplemental views are needed to provide a greater context of the system and project it is to be used in**
- **Small adaptation curve to use Capella**
  - Many Youtube videos, in tool examples, examples, training materials and forums to support the use of Arcadia Capella
  - Tool logic allows importing artifacts between layers
  - Tool logic has robust error checking
- **Advances by Siemens to bridge Capella to a robust MBE environment (System Modeling Workbench) with production, HW, FW, SW, etc.** (demonstrated at INCOSE IS2022) https://www.plm.automation.siemens.com/global/en/our-story/newsroom/system-modeling-workbench-teamcenter/43935
- **Transportability with SysML is in the works to allow interoperability between model types**

ARCADIA Capella is a high-fidelity Architecture synthesis methodology and tool. It can be directly used to synthesize a robust architecture and then moved (for now) to other models front ends, much the way other models tie into high fidelity tools (e.g. modeling and simulation, physics models, etc.) on the back end.

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Questions