Requirements Architecting: Object-Oriented To M&S-Driven

Sachin Mehta
Lead, Systems Engineering
Space Superiority & Geospatial Sector
L3Harris Technologies

Problem Space Introduction

- "…the majority of the tools on the market do not cover requirements traceability, and that even fewer provide support for the particular traceability requirements now enforced by DoD STD-2167A [U.S. Department of Defense 1988a]" [Gotel and Finkelstein, 1994][1].
- "…traceability refers to the ability to cross-reference items in the requirements specification with items in the design specification" [Roman, 1985][2].
- All too often, however, there is a lack of bonafide foundational relations between textual statements and architectural design.
- Thus, unique methods to enforce requirements traceability across architecture design specifications will be discussed by:
  - – Orienting the system/mission around aggregate objects.
  - – Exploiting systems modeling and simulation.

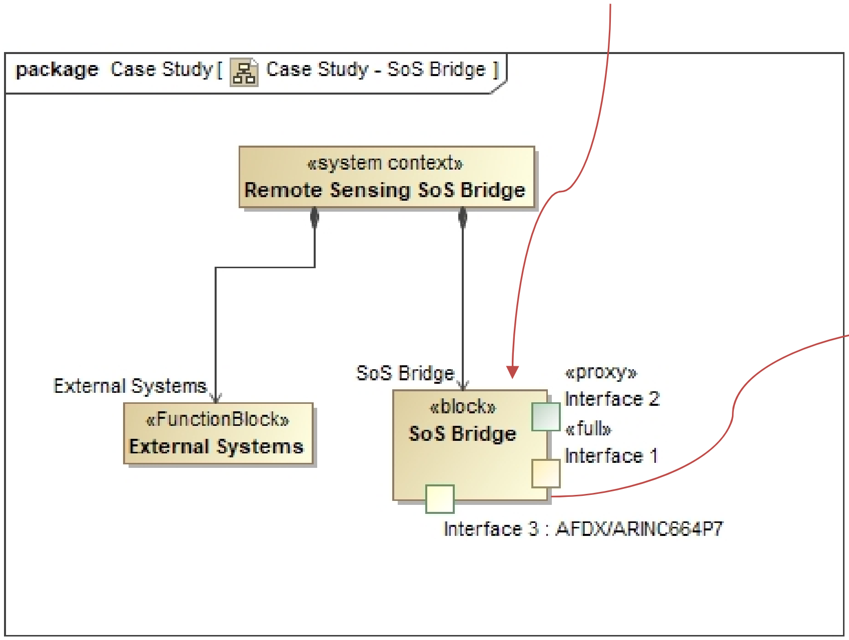
Proposal: Analytical Foundation for Requirements To Architecture

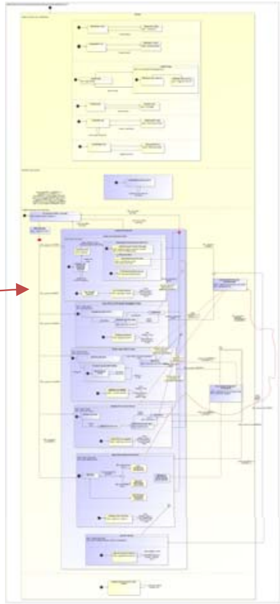
Validated

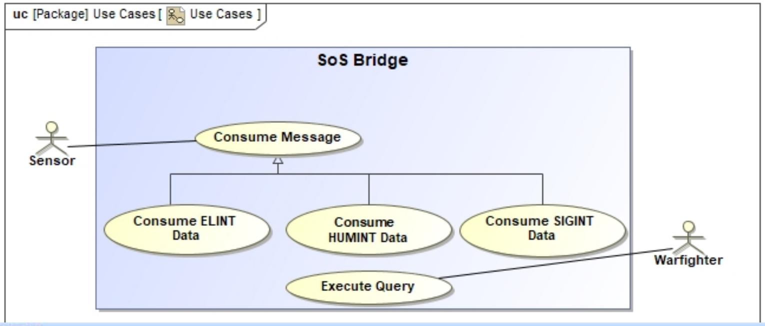
- Desired warfighting mission effects can be realized via exploiting M&S when we need to ensure the "…deliberate planning, analyzing, organizing, and integrating of current and emerging operational and systems capabilities…" [Gold, 2016][3].
- Design specifications can be seamlessly tied across the 'black' and 'white' box perspectives.
- Requirement specifications can be formulated that 'trace to' & revolve around the architectural paradigm.

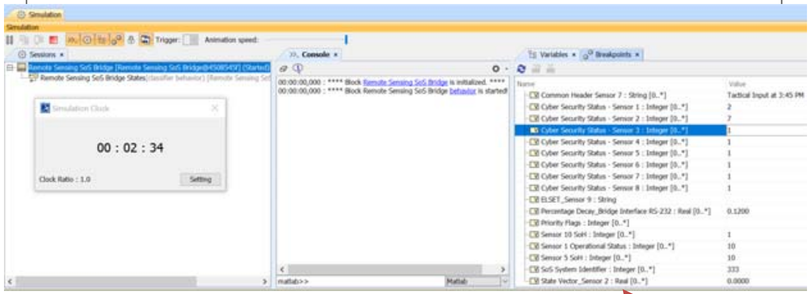
Case Study – Remote Sensing SoS Bridge

- The Statement of Work (SOW) reads the contractor shall develop technical requirements and a system architectural description.
- In many development cycles, there is an “over emphasis on simplistic use case…” [Firesmith, 2007][4] and representation of the system/mission solely by a single object.









Remote Sensing SoS Bridge [Remote Sensing SoS Bridge@45085457] (Started)
Remote Sensing SoS Bridge States(classifier behavior) [Remote Sensing SoS

00 : 02 : 34

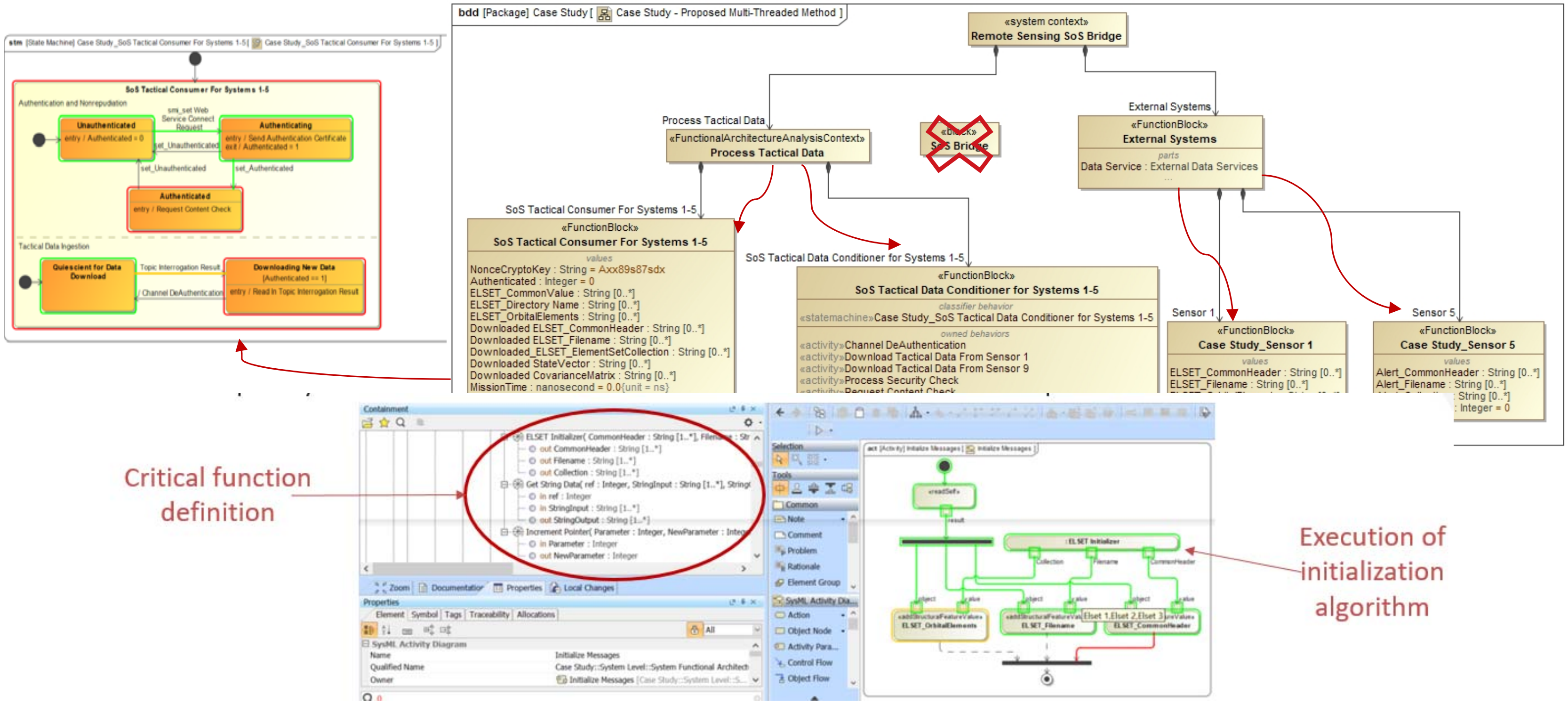
Clock Ratio : 1.0

```
00:00:00,000 : **** Block Remote Sensing SoS Bridge is initialized. ****
00:00:00,000 : **** Block Remote Sensing SoS Bridge behavior is started!
matlab>>
```

Name	Value
Common Header Sensor 7 : String [0..*]	Tactical Input at 3:45 PM
Cyber Security Status - Sensor 1 : Integer [0..*]	2
Cyber Security Status - Sensor 2 : Integer [0..*]	7
Cyber Security Status - Sensor 3 : Integer [0..*]	1
Cyber Security Status - Sensor 4 : Integer [0..*]	1
Cyber Security Status - Sensor 5 : Integer [0..*]	1
Cyber Security Status - Sensor 6 : Integer [0..*]	1
Cyber Security Status - Sensor 7 : Integer [0..*]	1
Cyber Security Status - Sensor 8 : Integer [0..*]	1
ELSET_Sensor 9 : String	
Percentage Decay_Bridge Interface RS-232 : Real [0..*]	0.1200
Priority Flags : Integer [0..*]	
Sensor 10 SoH : Integer [0..*]	1
Sensor 1 Operational Status : Integer [0..*]	10
Sensor 5 SoH : Integer [0..*]	10
SoS System Identifier : Integer [0..*]	333
State Vector_Sensor 2 : Real [0..*]	0.0000

Formulate The Context Using M&S

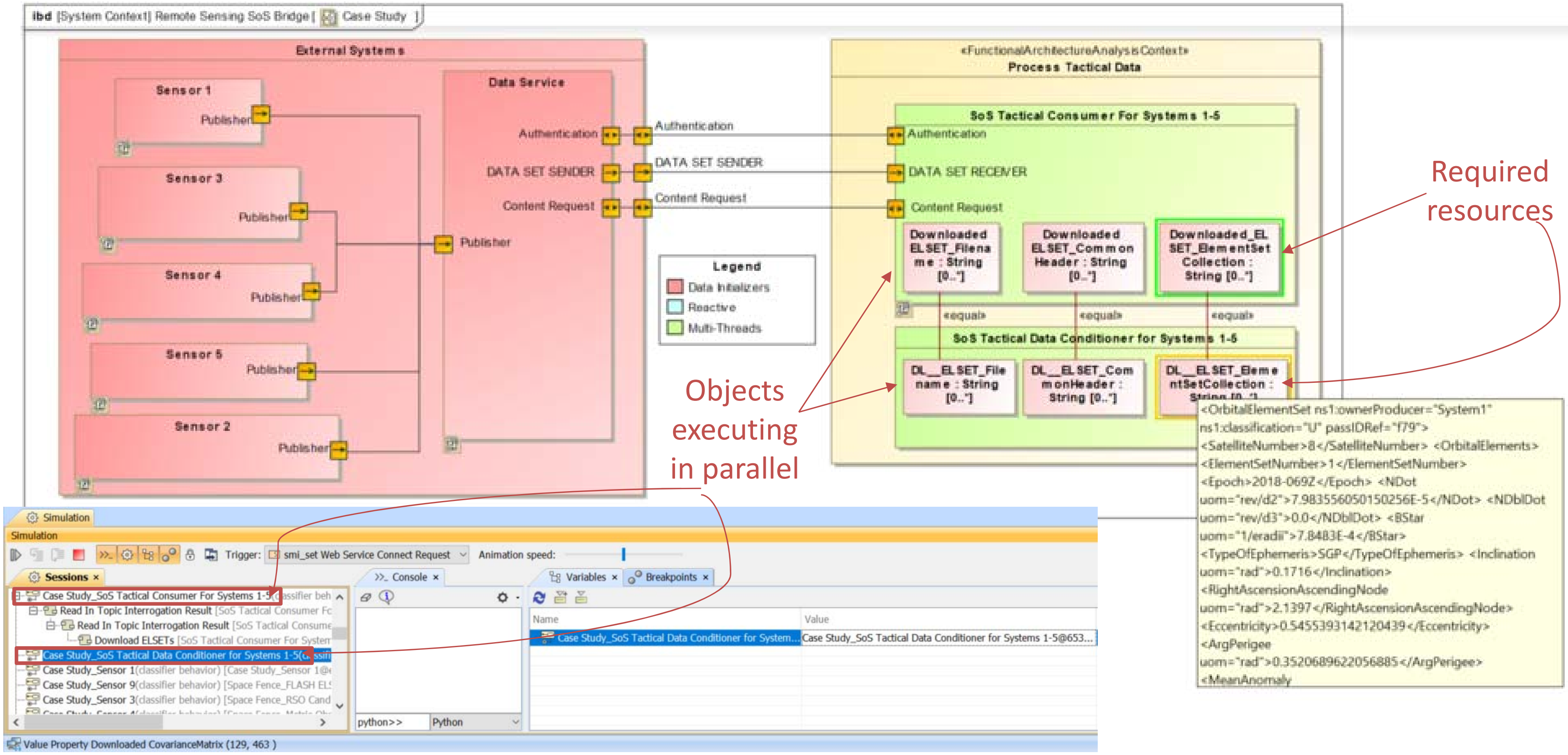
- Decompose the mission context to elaborate functionality, external/internal boundary points, and performance.
- Aggregate behavioral threads and non-functional requirements (NFRs), within each use case, to multiple objects.


bdd [Package] Case Study [Case Study - Proposed Multi-Threaded Method]
«system context»
Remote Sensing SoS Bridge
Process Tactical Data
«FunctionalArchitectureAnalysisContext»
Process Tactical Data
«block»
SoS Bridge
External Systems
«FunctionBlock»
External Systems
parts
Data Service : External Data Services
SoS Tactical Consumer For Systems 1-5
«FunctionBlock»
SoS Tactical Consumer For Systems 1-5
values
NonceCryptoKey : String = Axx89s87sdx
Authenticated : Integer = 0
ELSET_CommonValue : String [0..*]
ELSET_Directory Name : String [0..*]
ELSET_OrbitalElements : String [0..*]
Downloaded ELSET_CommonHeader : String [0..*]
Downloaded ELSET_Filename : String [0..*]
Downloaded_ELSET_ElementSetCollection : String [0..*]
Downloaded StateVector : String [0..*]
Downloaded CovarianceMatrix : String [0..*]
MissionTime : nanosecond = 0.0{unit = ns}
SoS Tactical Data Conditioner for Systems 1-5
«FunctionBlock»
SoS Tactical Data Conditioner for Systems 1-5
classifier behavior
«statemachine»Case Study_SoS Tactical Data Conditioner for Systems 1-5
owned behaviors
«activity»Channel DeAuthentication
«activity»Download Tactical Data From Sensor 1
«activity»Download Tactical Data From Sensor 9
«activity»Process Security Check
«activity»Request Content Check
Sensor 1
«FunctionBlock»
Case Study_Sensor 1
values
ELSET_CommonHeader : String [0..*]
ELSET_Filename : String [0..*]
Sensor 5
«FunctionBlock»
Case Study_Sensor 5
values
Alert_CommonHeader : String [0..*]
Alert_Filename : String [0..*]
: String [0..*]
: Integer = 0
stm [State Machine] Case Study_SoS Tactical Consumer For Systems 1-5 [Case Study_SoS Tactical Consumer For Systems 1-5]
SoS Tactical Consumer For Systems 1-5
Authentication and Nonrepudiation
Unauthenticated
entry / Authenticated = 0
sm_set Web Service Connect Request
Authenticating
entry / Send Authentication Certificate
exit / Authenticated = 1
set_Unauthenticated
set_Authenticated
Authenticated
entry / Request Content Check
Tactical Data Ingestion
Quiescent for Data Download
Topic Interrogation Result
Downloading New Data
[Authenticated == 1]
entry / Read In Topic Interrogation Result
/ Channel DeAuthentication
Containment
ELSET Initializer(CommonHeader : String [1..*], Filename : String
out CommonHeader : String [1..*]
out Filename : String [1..*]
out Collection : String [1..*]
Get String Data(ref : Integer, StringInput : String [1..*], String
in ref : Integer
in StringInput : String [1..*]
out StringOutput : String [1..*]
Increment Pointer(Parameter : Integer, NewParameter : Integer
in Parameter : Integer
out NewParameter : Integer
Zoom
Documentation
Properties
Local Changes
Properties
Element
Symbol
Tags
Traceability
Allocations
All
SysML Activity Diagram
Name
Initialize Messages
Qualified Name
Case Study::System Level::System Functional Architect
Owner
Initialize Messages [Case Study::System Level::S...
Selection
Tools
Common
Note
Comment
Problem
Rationale
Element Group
SysML Activity Dia...
Action
Object Node
Activity Para...
Control Flow
Object Flow
act [Activity] Initialize Messages [Initialize Messages]
readSelf
result
ELSET Initializer
Collection
Filename
CommonHeader
object
value
ELSET_OrbitalElements
ELSET_Filename
ELSET_CommonHeader

Critical function definition

Execution of initialization algorithm

Exploit M&S To Analyze Capabilities


ibd [System Context] Remote Sensing SoS Bridge [Case Study]
External Systems
Sensor 1
Publisher
Sensor 3
Publisher
Sensor 4
Publisher
Sensor 5
Publisher
Sensor 2
Publisher
Data Service
Authentication
DATA SET SENDER
Content Request
Publisher
Authentication
DATA SET SENDER
Content Request
Legend
🟥 Data Initializers
🟦 Reactive
🟩 Multi-Threads
«FunctionalArchitectureAnalysisContext»
Process Tactical Data
SoS Tactical Consumer For Systems 1-5
Authentication
DATA SET RECEIVER
Content Request
Downloaded ELSET_Filename : String [0..*]
Downloaded ELSET_CommonHeader : String [0..*]
Downloaded_ELSET_ElementSet Collection : String [0..*]
«equal»
«equal»
«equal»
SoS Tactical Data Conditioner for Systems 1-5
DL__ELSET_Filename : String [0..*]
DL__ELSET_CommonHeader : String [0..*]
DL__ELSET_ElementSetCollection : String [0..*]
Simulation
Variables
Breakpoints
Name
Value
Animation speed:

Required resources

Objects executing in parallel

Simulation

Trigger: smi_set Web Service Connect Request

Sessions

- Case Study_SoS Tactical Consumer For Systems 1-5(classifier beh
  - Read In Topic Interrogation Result [SoS Tactical Consumer Fc
    - Read In Topic Interrogation Result [SoS Tactical Consume
      - Download ELSETs [SoS Tactical Consumer For System
- Case Study_SoS Tactical Data Conditioner for Systems 1-5(classifi
- Case Study_Sensor 1(classifier behavior) [Case Study_Sensor 1@e
- Case Study_Sensor 9(classifier behavior) [Space Fence_FLASH ELS
- Case Study_Sensor 3(classifier behavior) [Space Fence_RSO Cand

Console

python>> Python

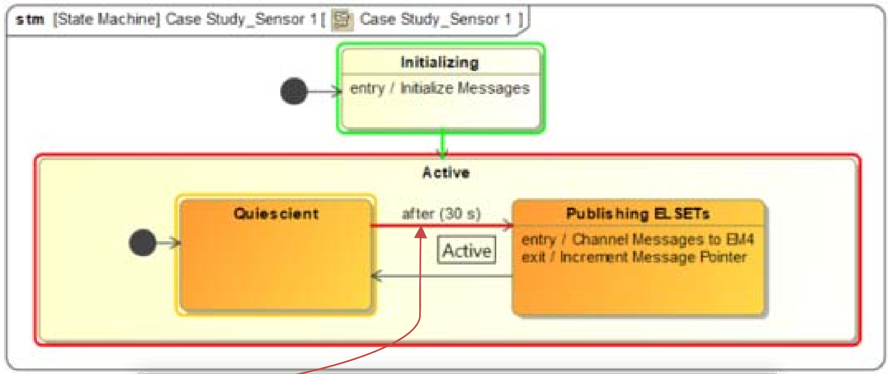
Name	Value
Case Study_SoS Tactical Data Conditioner for System...	Case Study_SoS Tactical Data Conditioner for Systems 1-5@653...

Value Property Downloaded CovarianceMatrix (129, 463)

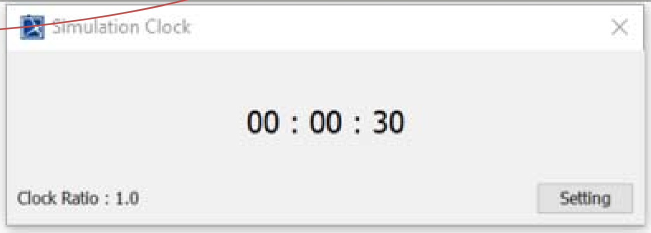
```
<OrbitalElementSet ns1:ownerProducer="System1"
ns1:classification="U" passIDRef="f79">
<SatelliteNumber>8</SatelliteNumber> <OrbitalElements>
<ElementSetNumber>1</ElementSetNumber>
<Epoch>2018-069Z</Epoch> <NDot
uom="rev/d2">7.983556050150256E-5</NDot> <NDblDot
uom="rev/d3">0.0</NDblDot> <BStar
uom="1/eradii">7.8483E-4</BStar>
<TypeOfEphemeris>SGP</TypeOfEphemeris> <Inclination
uom="rad">0.1716</Inclination>
<RightAscensionAscendingNode
uom="rad">2.1397</RightAscensionAscendingNode>
<Eccentricity>0.5455393142120439</Eccentricity>
<ArgPerigee
uom="rad">0.3520689622056885</ArgPerigee>
<MeanAnomaly
```

Exploit Simulation Modeling, ctd.

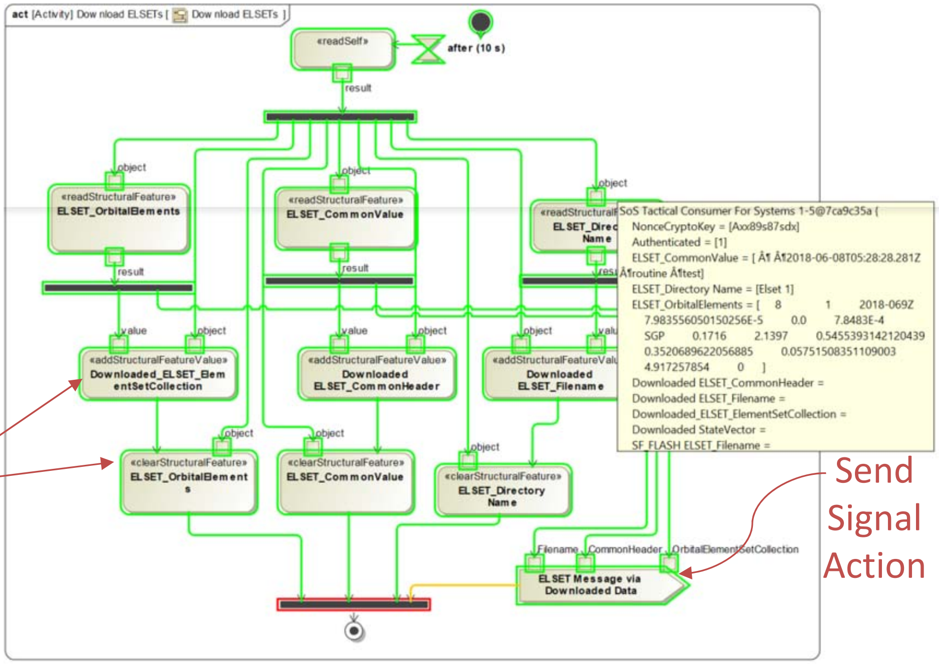
- Utilize stateful and transformational programming, where applicable.
- Robust development environments provide the ability to define behavior, signals, data structures, algorithms, and timing.


stm [State Machine] Case Study_Sensor 1 [Case Study_Sensor 1]
Initializing
entry / Initialize Messages
Active
Quiescent
after (30 s)
Active
Publishing ELSETs
entry / Channel Messages to EM4
exit / Increment Message Pointer

Time Event



Implement fUML and other languages at run-time

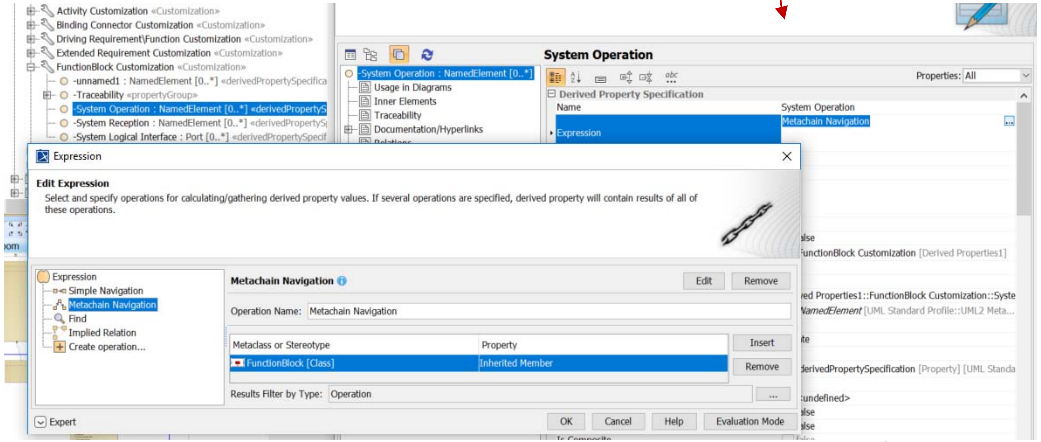


```
SoS Tactical Consumer For Systems 1-5@7ca9c35a (
   NonceCryptoKey = [Axx89s87sdx]
   Authenticated = [1]
   ELSET_CommonValue = [ Â¶ Â¶2018-06-08T05:28:28.281Z
Â¶routine Â¶test]
   ELSET_Directory Name = [Elset 1]
   ELSET_OrbitalElements = [    8          1        2018-069Z
      7.983556050150256E-5       0.0       7.8483E-4
      SGP      0.1716      2.1397      0.5455393142120439
      0.3520689622056885       0.05751508351109003
      4.917257854       0     ]
   Downloaded ELSET_CommonHeader =
   Downloaded ELSET_Filename =
   Downloaded_ELSET_ElementSetCollection =
   Downloaded StateVector =
   SF_FLASH ELSET_Filename =
```

Send Signal Action

Black Box Perspective

- Orient yourself around aggregate objects from the simulation model.
- Where requirements are realized: create operations, create signal receptions, and stereotype interfaces.
- Utilize ‘generalization’, to specify ‘black box’.
- Define integrated queries within development environment.



Activity Customization «Customization»
Binding Connector Customization «Customization»
Driving Requirement\Function Customization «Customization»
Extended Requirement Customization «Customization»
FunctionBlock Customization «Customization»
-unnamed1 : NamedElement [0..*] «derivedPropertySpecifica
-Traceability «propertyGroup»
-System Operation : NamedElement [0..*] «derivedPropertyS
-System Reception : NamedElement [0..*] «derivedPropertyS
-System Logical Interface : Port [0..*] «derivedPropertySpecif

-System Operation : NamedElement [0..*]
Usage in Diagrams
Inner Elements
Traceability
Documentation/Hyperlinks

System Operation

Properties: All

Derived Property Specification

Name	System Operation
Expression	Metachain Navigation

Expression

Edit Expression

Select and specify operations for calculating/gathering derived property values. If several operations are specified, derived property will contain results of all of these operations.

Expression
Simple Navigation
Metachain Navigation
Find
Implied Relation
Create operation...

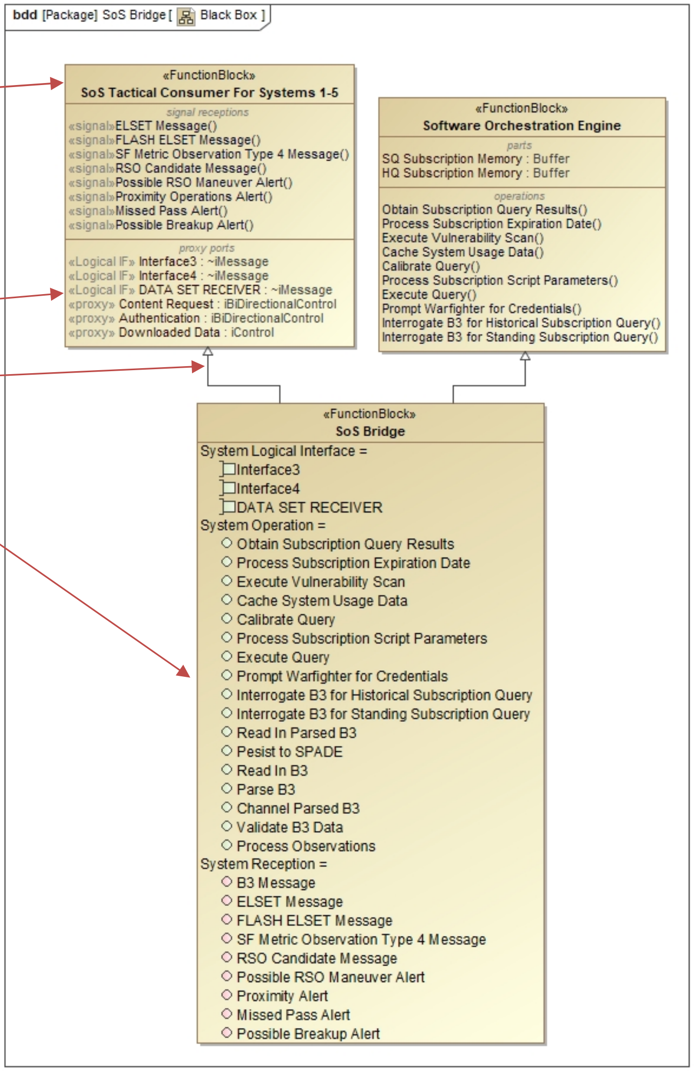
Metachain Navigation

Operation Name: Metachain Navigation

Metaclass or Stereotype	Property
FunctionBlock [Class]	Inherited Member

Results Filter by Type: Operation

alse
unctionBlock Customization [Derived Properties1]
ed Properties1::FunctionBlock Customization::Syste
NamedElement [UML Standard Profile::UML2 Meta...
te
derivedPropertySpecification [Property] [UML Standa
undefined>
alse
alse



bdd [Package] SoS Bridge [Black Box]

«FunctionBlock»
SoS Tactical Consumer For Systems 1-5

signal receptions
«signal»ELSET Message()
«signal»FLASH ELSET Message()
«signal»SF Metric Observation Type 4 Message()
«signal»RSO Candidate Message()
«signal»Possible RSO Maneuver Alert()
«signal»Proximity Operations Alert()
«signal»Missed Pass Alert()
«signal»Possible Breakup Alert()

proxy ports
«Logical IF» Interface3 : ~iMessage
«Logical IF» Interface4 : ~iMessage
«Logical IF» DATA SET RECEIVER : ~iMessage
«proxy» Content Request : iBiDirectionalControl
«proxy» Authentication : iBiDirectionalControl
«proxy» Downloaded Data : iControl

«FunctionBlock»
Software Orchestration Engine

parts
SQ Subscription Memory : Buffer
HQ Subscription Memory : Buffer

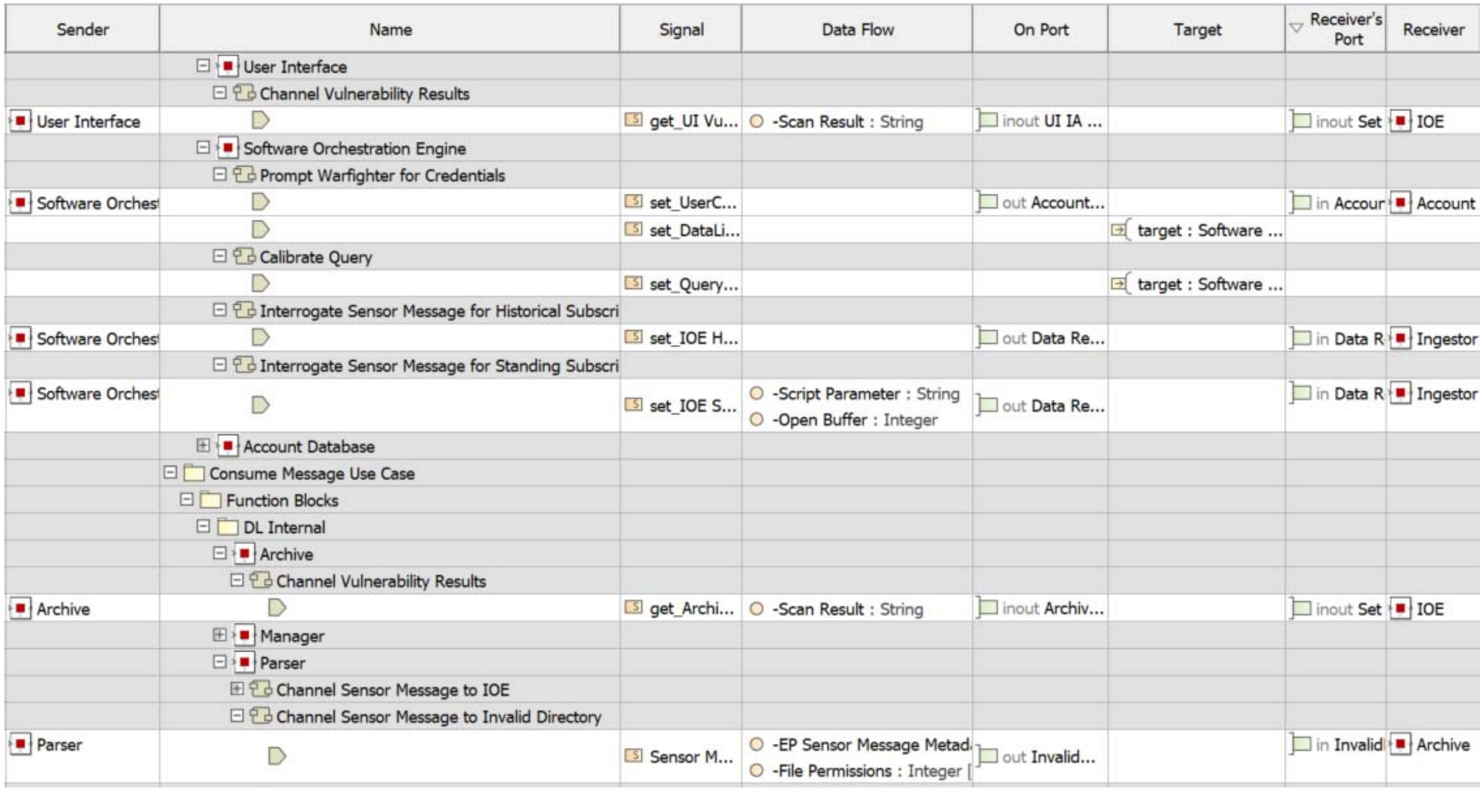
operations
Obtain Subscription Query Results()
Process Subscription Expiration Date()
Execute Vulnerability Scan()
Cache System Usage Data()
Calibrate Query()
Process Subscription Script Parameters()
Execute Query()
Prompt Warfighter for Credentials()
Interrogate B3 for Historical Subscription Query()
Interrogate B3 for Standing Subscription Query()

«FunctionBlock»
SoS Bridge

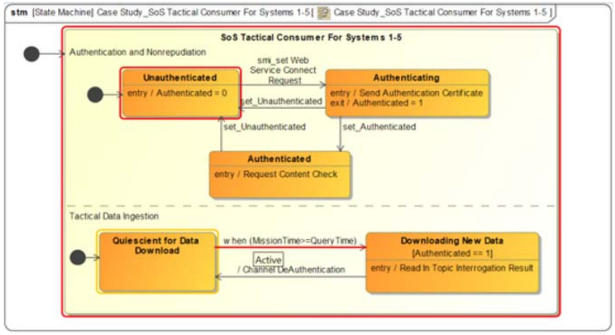
System Logical Interface =
Interface3
Interface4
DATA SET RECEIVER
System Operation =
Obtain Subscription Query Results
Process Subscription Expiration Date
Execute Vulnerability Scan
Cache System Usage Data
Calibrate Query
Process Subscription Script Parameters
Execute Query
Prompt Warfighter for Credentials
Interrogate B3 for Historical Subscription Query
Interrogate B3 for Standing Subscription Query
Read In Parsed B3
Pesist to SPADE
Read In B3
Parse B3
Channel Parsed B3
Validate B3 Data
Process Observations
System Reception =
B3 Message
ELSET Message
FLASH ELSET Message
SF Metric Observation Type 4 Message
RSO Candidate Message
Possible RSO Maneuver Alert
Proximity Alert
Missed Pass Alert
Possible Breakup Alert

White Box Perspective

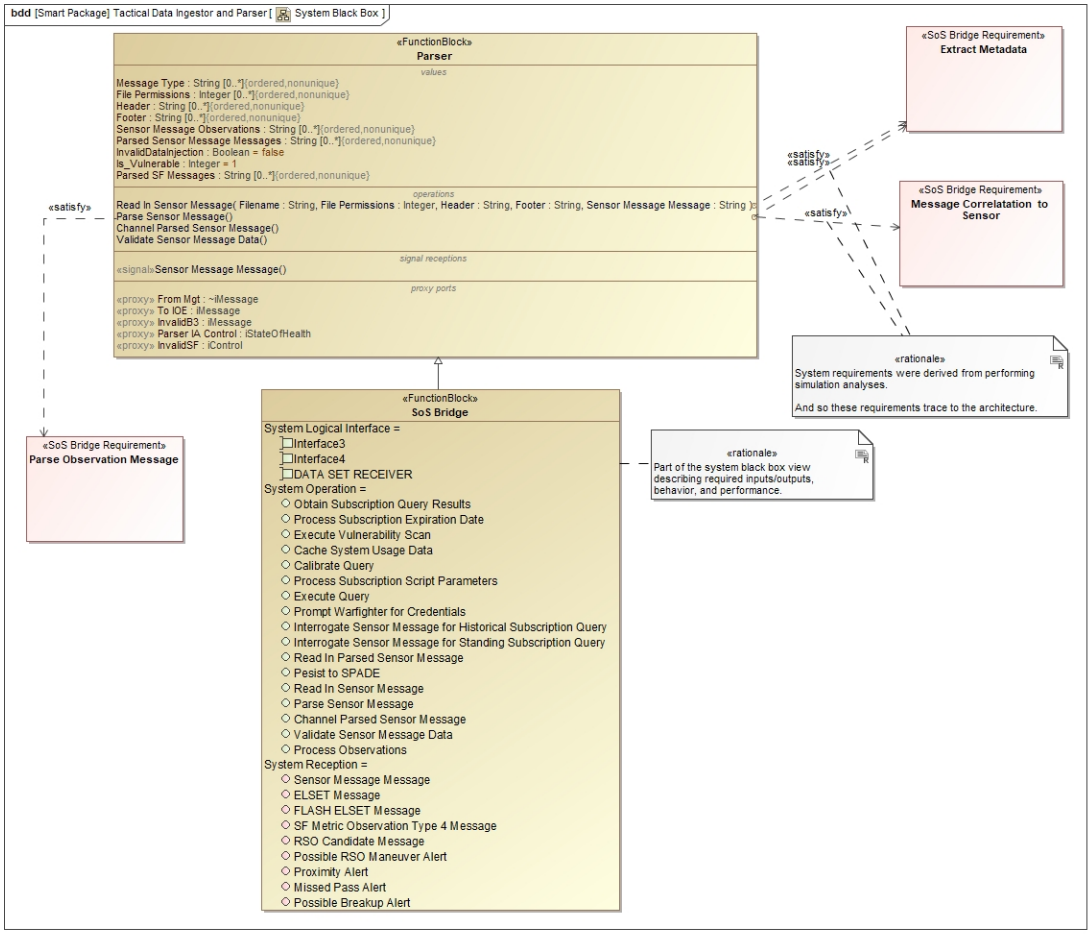
- Detailed design specifications obtained from “non-simplistic” M&S-driven (and object-oriented) analyses.



Sender	Name	Signal	Data Flow	On Port	Target	Receiver's Port	Receiver
	⊟ User Interface						
	⊟ Channel Vulnerability Results						
User Interface		get_UI Vu...	○ -Scan Result : String	inout UI IA ...		inout Set	IOE
	⊟ Software Orchestration Engine						
	⊟ Prompt Warfighter for Credentials						
Software Orches		set_UserC...		out Account...		in Accour	Account
		set_DataLi...			target : Software ...		
	⊟ Calibrate Query						
		set_Query...			target : Software ...		
	⊟ Interrogate Sensor Message for Historical Subscri						
Software Orches		set_IOE H...		out Data Re...		in Data R	Ingestor
	⊟ Interrogate Sensor Message for Standing Subscri						
Software Orches		set_IOE S...	○ -Script Parameter : String ○ -Open Buffer : Integer	out Data Re...		in Data R	Ingestor
	⊞ Account Database						
	⊟ Consume Message Use Case						
	⊟ Function Blocks						
	⊟ DL Internal						
	⊟ Archive						
	⊟ Channel Vulnerability Results						
Archive		get_Archi...	○ -Scan Result : String	inout Archiv...		inout Set	IOE
	⊞ Manager						
	⊟ Parser						
	⊞ Channel Sensor Message to IOE						
	⊟ Channel Sensor Message to Invalid Directory						
Parser		Sensor M...	○ -EP Sensor Message Metad ○ -File Permissions : Integer [	out Invalid...		in Invalid	Archive


stm [State Machine] Case Study_SoS Tactical Consumer For Systems 1-5[Case Study_SoS Tactical Consumer For Systems 1-5]
SoS Tactical Consumer For Systems 1-5
Authentication and Nonrepudiation
Unauthenticated
entry / Authenticated = 0
smi_set Web Service Connect Request
Authenticating
entry / Send Authentication Certificate
exit / Authenticated = 1
set_Unauthenticated
set_Unauthenticated
set_Authenticated
Authenticated
entry / Request Content Check
Tactical Data Ingestion
Quiescient for Data Download
when (MissionTime>=QueryTime)
Active
/ Channel DeAuthentication
Downloading New Data
[Authenticated == 1]
entry / Read In Topic Interrogation Result

Requirements → Architecture



bdd [Smart Package] Tactical Data Ingestor and Parser [System Black Box]

«FunctionBlock»
Parser

values
Message Type : String [0..*]{ordered,nonunique}
File Permissions : Integer [0..*]{ordered,nonunique}
Header : String [0..*]{ordered,nonunique}
Footer : String [0..*]{ordered,nonunique}
Sensor Message Observations : String [0..*]{ordered,nonunique}
Parsed Sensor Message Messages : String [0..*]{ordered,nonunique}
InvalidDataInjection : Boolean = false
Is_Vulnerable : Integer = 1
Parsed SF Messages : String [0..*]{ordered,nonunique}

operations
Read In Sensor Message(Filename : String, File Permissions : Integer, Header : String, Footer : String, Sensor Message Message : String)
Parse Sensor Message()
Channel Parsed Sensor Message()
Validate Sensor Message Data()

signal receptions
«signal»Sensor Message Message()

proxy ports
«proxy» From Mgt : ~iMessage
«proxy» To IOE : iMessage
«proxy» InvalidB3 : iMessage
«proxy» Parser IA Control : iStateOfHealth
«proxy» InvalidSF : iControl

«SoS Bridge Requirement»
Extract Metadata

«SoS Bridge Requirement»
Message Correlatation to Sensor

«SoS Bridge Requirement»
Parse Observation Message

«rationale»
System requirements were derived from performing simulation analyses.

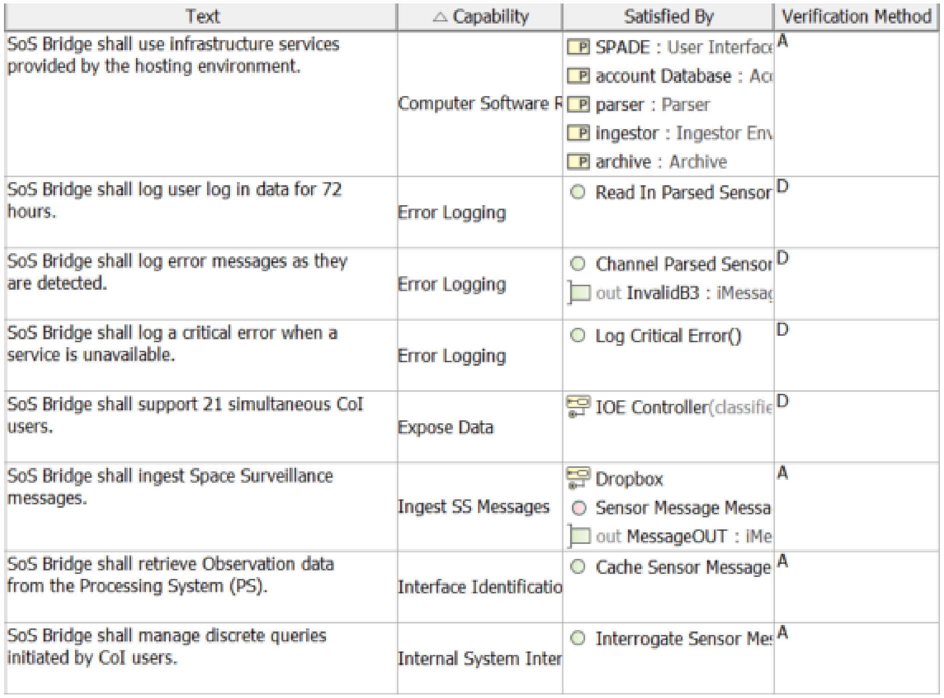
And so these requirements trace to the architecture.

«FunctionBlock»
SoS Bridge

System Logical Interface =
Interface3
Interface4
DATA SET RECEIVER
System Operation =
○ Obtain Subscription Query Results
○ Process Subscription Expiration Date
○ Execute Vulnerability Scan
○ Cache System Usage Data
○ Calibrate Query
○ Process Subscription Script Parameters
○ Execute Query
○ Prompt Warfighter for Credentials
○ Interrogate Sensor Message for Historical Subscription Query
○ Interrogate Sensor Message for Standing Subscription Query
○ Read In Parsed Sensor Message
○ Pesist to SPADE
○ Read In Sensor Message
○ Parse Sensor Message
○ Channel Parsed Sensor Message
○ Validate Sensor Message Data
○ Process Observations
System Reception =
○ Sensor Message Message
○ ELSET Message
○ FLASH ELSET Message
○ SF Metric Observation Type 4 Message
○ RSO Candidate Message
○ Possible RSO Maneuver Alert
○ Proximity Alert
○ Missed Pass Alert
○ Possible Breakup Alert

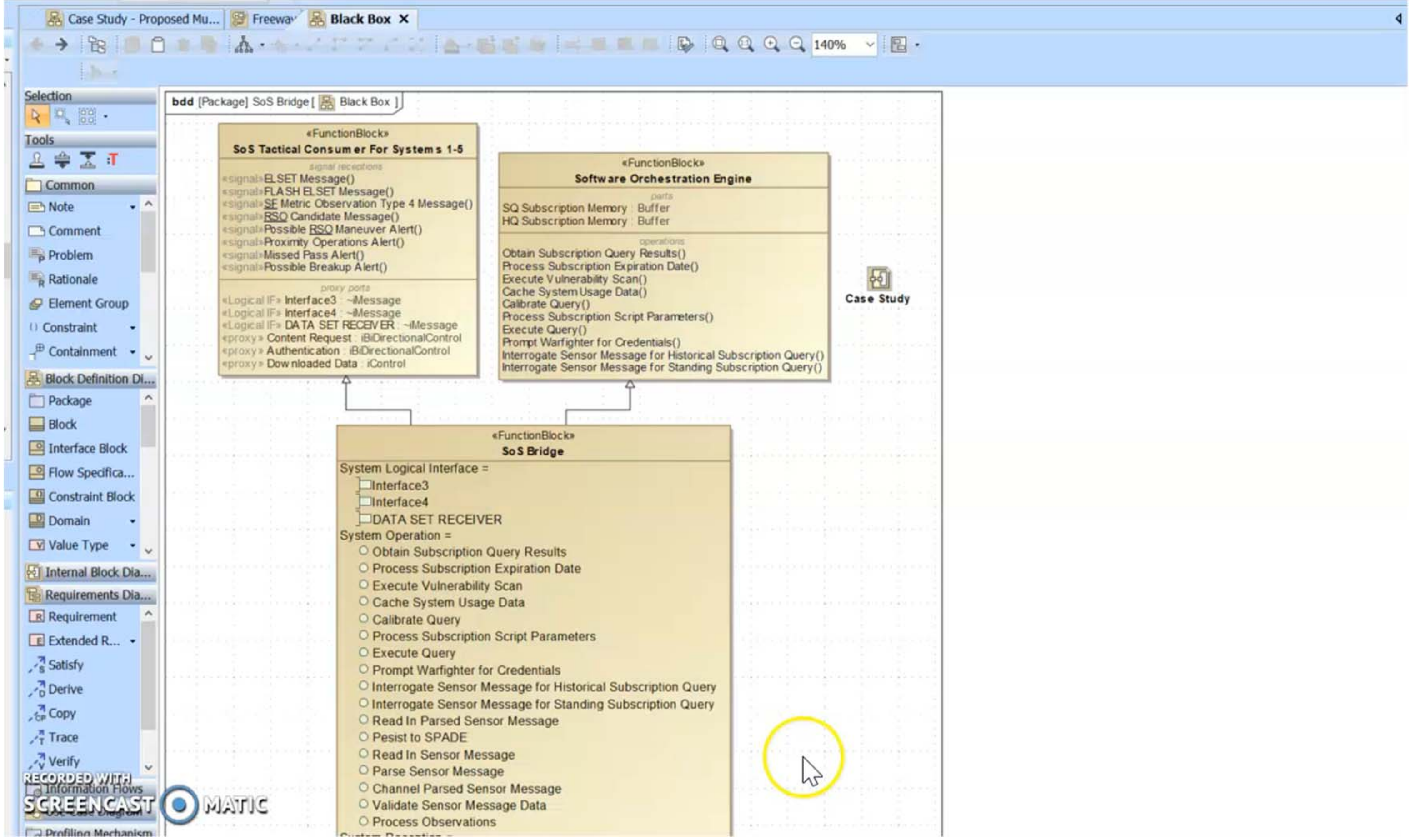
«rationale»
Part of the system black box view describing required inputs/outputs, behavior, and performance.

«satisfy»
«satisfy»
«satisfy»
«satisfy»



Text	△ Capability	Satisfied By	Verification Method
SoS Bridge shall use infrastructure services provided by the hosting environment.	Computer Software R	SPADE : User Interface account Database : Acc parser : Parser ingestor : Ingestor Env archive : Archive	A
SoS Bridge shall log user log in data for 72 hours.	Error Logging	○ Read In Parsed Sensor	D
SoS Bridge shall log error messages as they are detected.	Error Logging	○ Channel Parsed Sensor out InvalidB3 : iMessag	D
SoS Bridge shall log a critical error when a service is unavailable.	Error Logging	○ Log Critical Error()	D
SoS Bridge shall support 21 simultaneous CoI users.	Expose Data	IOE Controller(classifie	D
SoS Bridge shall ingest Space Surveillance messages.	Ingest SS Messages	Dropbox ○ Sensor Message Messa out MessageOUT : iMe	A
SoS Bridge shall retrieve Observation data from the Processing System (PS).	Interface Identificatio	○ Cache Sensor Message	A
SoS Bridge shall manage discrete queries initiated by CoI users.	Internal System Inter	○ Interrogate Sensor Me:	A

Case Study Video Clip



bdd [Package] SoS Bridge [Black Box]

«FunctionBlock»
SoS Tactical Consumer For Systems 1-5

signal receptions
«signal»ELSET Message()
«signal»FLASH ELSET Message()
«signal»SF Metric Observation Type 4 Message()
«signal»RSO Candidate Message()
«signal»Possible RSO Maneuver Alert()
«signal»Proximity Operations Alert()
«signal»Missed Pass Alert()
«signal»Possible Breakup Alert()

proxy ports
«Logical IF» Interface3 : ~iMessage
«Logical IF» Interface4 : ~iMessage
«Logical IF» DATA SET RECEIVER : ~iMessage
«proxy» Content Request : iBiDirectionalControl
«proxy» Authentication : iBiDirectionalControl
«proxy» Downloaded Data : iControl

«FunctionBlock»
Software Orchestration Engine

parts
SQ Subscription Memory : Buffer
HQ Subscription Memory : Buffer

operations
Obtain Subscription Query Results()
Process Subscription Expiration Date()
Execute Vulnerability Scan()
Cache System Usage Data()
Calibrate Query()
Process Subscription Script Parameters()
Execute Query()
Prompt Warfighter for Credentials()
Interrogate Sensor Message for Historical Subscription Query()
Interrogate Sensor Message for Standing Subscription Query()

«FunctionBlock»
SoS Bridge

System Logical Interface =
Interface3
Interface4
DATA SET RECEIVER
System Operation =
○ Obtain Subscription Query Results
○ Process Subscription Expiration Date
○ Execute Vulnerability Scan
○ Cache System Usage Data
○ Calibrate Query
○ Process Subscription Script Parameters
○ Execute Query
○ Prompt Warfighter for Credentials
○ Interrogate Sensor Message for Historical Subscription Query
○ Interrogate Sensor Message for Standing Subscription Query
○ Read In Parsed Sensor Message
○ Pesist to SPADE
○ Read In Sensor Message
○ Parse Sensor Message
○ Channel Parsed Sensor Message
○ Validate Sensor Message Data
○ Process Observations
System Reception =

Conclusion

- Validated operational capability specifications can be realized when orienting the system/mission around multiple objects.

- Requirements can be architected with a stronger analytical foundation, using formal M&S as a driver.

- Design and requirements can be tied together by orienting the system/mission around multiple objects and using formal M&S as a driver.

Thank You!

Sachin Mehta
Lead, Systems Engineering
L3Harris Technologies
Space & Airborne Systems Segment
Space Superiority & Geospatial Sector
Sensor and Applied Defense Solutions Division
Sachin.Mehta@L3Harris.com

References

1. Gotel, Orlena CZ, and C. W. Finkelstein. "An analysis of the requirements traceability problem." Proceedings of IEEE International Conference on Requirements Engineering,1994.
2. Roman, G. C. “A Taxonomy of Current Issues in Requirements Engineering.” COMPUTER, 1985.
3. Gold, R. “Mission Engineering.” Presentation at the 19th NDIA Systems Engineering Conference. October 24-27, Springfield, VA, 2016.
4. Firesmith, D. "Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them." *Journal of Object Technology* 6.1, 2007.