Model-Based Systems Engineering (MBSE) Using the Model-Based System Architecture Process (MBSAP)

Dr. Mike Borky
Case for Action (“Burning Platform”)

- **Fundamental challenges:**
  - Systems are becoming increasingly complex, networked, adaptive, programmable, cyber-vulnerable, and expensive
  - Operational capability is increasingly dominated by information, computation, and communications
  - Cost and schedule growth, performance and reliability shortfalls, and premature obsolescence are commonplace
  - Many systems must be sustained and upgraded over long lifetimes
- A more robust and effective System Engineering process is widely viewed as essential for improvement
- Model-Based/Digital Engineering, including MBSE, is the current industry best practice for managing complexity and optimizing delivered capability, especially in information-intensive systems and enterprises

**MBSE provides better tools and processes for executing core SE tasks**

**BUT**

**Transitioning to MBSE is as much a cultural change as a technical one**
Topics

- Nature and Payoffs of MBSE
- Basic Methodology - Model-Based System Architecture Process (MBSAP)
- Case Study – Microwave Oven Model
Some Historical Context

- Prior to and during WW II – origins of Systems Engineering, e.g., B-29 program office to coordinate multiple companies and emerging technologies
- 1950s/1960s – SE methodologies, e.g., requirements analysis and verification, allocated baselines, configuration management, etc. – critical to the success of Project Apollo
- 1960s/1970s – structured programming and the start of Object-Oriented Design (OOD) with ALGOL, PASCAL, LISP, et al. – Ada was DoD’s attempt
- 2000s – Systems Modeling Language (SysML) with steadily improving tool support, also maintained by OMG
Basic Principles of MBSE

- Replace Document-Centric with Model-Centric SE Processes – the model becomes the framework for storing and finding system information
- Establish an integrated tool environment
- Capture, analyze, document, and evolve design under rigorous control with metrics
- Use the model and its artifacts (the architecture) as the basic materials for SE processes and products:
  - Requirements Analysis
  - Performance and Design Specifications
  - Project Planning
  - Performance and Design Trade Studies
  - Configuration Management
  - Integration and Test
- Integrate management, design, logistics, test, specialty engineering, and stakeholders in a collaborative environment
- Support networking and enterprise processes
- Use the model for metrics and governance
- Implement upgrades, tech refresh, and sustainment for the long term

Complex, Often Information-Intensive Systems and Enterprises

Complete, Unambiguous Representation in a Standards-Compliant Architecture Model

Foundation for a Consistent, Controlled, High Quality Systems Engineering Process
### Problems and Payoffs

**LEGACY SE**
- Poor requirements analysis, stability, allocation, V&V
- Limited commonality and reuse
- Integration and test failures
- Late, over-budget, under-performing software
- Expensive, hard to integrate, and hard to upgrade hardware
- Limited interoperability, networking problems
- Closed systems that are hard to modify or upgrade
- Problems in life-cycle sustainment and modernization

**MBSE**
- Unambiguous definition, allocation, traceability of requirements
- Model archives, design patterns
- Strong I&T support, e.g., test cases
- Strong support for agile, object-oriented, DevSecOps, et al.
- Key transition from functional to physical design w/ auto-generated specs, ICDs, et al.
- Support for standards, interface management, performance analysis
- Enforceable modular, loosely coupled architectures → OPEN
- Functional, open architecture supports modular upgrading/technology refreshment
Summary of the MBSAP Methodology

• Capture whatever is known about customer needs/desires/constraints
• Baseline in a requirements management tool

• Transform requirements into a high-level architectural context
• Define system boundary and context, top-level partitioning (Domains), primary behaviors (Use Cases), and primary data content
• Map requirements to Domains and Use Cases
• Begin discovery of both external and internal (Domain) services
• Develop a Conceptual Data Model

• Develop technology-independent functional system design
• Decompose Domains and Use Cases in structural and behavioral diagrams
• Develop a Logical Data Model
• Develop and allocate functional service specs
• Define architectural layering

• Complete trade studies and point design decisions
• Implement logical components with physical products, interfaces, timing, etc.
• Define standards profile
• Develop Physical Data Model and service definitions
MBSAP Mapped to the SE “Vee”

- Depending on the system or other entity, multiple tools may be needed by various disciplines and processes.
- A SysML model is the root of the environment – “Single Source of Technical Truth” (SSTT)
Executables – Simulation at Four Levels of Abstraction Creates a Virtual Prototype

- **Operational** – Requirements, Concept of Operations, Measures of Performance/Effectiveness (MoPs/MoEs)
- **Process/Workflow** – Business Processes/Threads, User Interactions
- **Architecture** – Logical Elements/Behaviors
- **Physical** – Hardware and Software Models
MBSE with the MBSAP Methodology

Operational & Process/Workflow M&S

Approved Need/ Capabilities Statement

CONCEPT EXPLORATION
- Rqmts Definition/Refinement
- Functional/Non-Functional Rqmts
- Performance/Timing Analysis
- Stakeholder Dialog

PERFORMANCE RQMTS
- Requirements Database

CONCEPT DEFINITION
- System/Concept Selection
- Functional Specs Development
  - Refined AoA
  - SEMP
  - WBS
  - MPS

FUNCTIONAL SPECIFICATIONS

INITIAL PHYSICAL VIEWPOINT
- System Design & Test
- Specialty Engineering
- Interface Definition
- Integration & Test Planning

Advanced Development/ Preliminary Design
- Risk Mitigation
- Technology/Subsystem Demos
- Component Rqmts Analysis

DESIGN SPECIFICATIONS

TEST READINESS REVIEW
- Critical Design Review
- Initial Product Specs

All Levels M&S

Initial Physical Viewpoint & Focused Viewpoints

Physical M&S
Example: a Microwave Oven

GE - 1.1 Cu. Ft. Mid-Size Microwave - Stainless steel
Model: JES1145SHSS  SKU: 2026035

4.5 (2,391 Reviews)  216 Answered Questions

Price Match Guarantee
$109.99
Open-Box: from $100.99

Color: Stainless steel

Protect your product

- 2-Year Standard Geek Squad Protection
  $14.99
  About $0.62/mo.

- 4-Year Standard Geek Squad Protection
  $29.99
  About $0.62/mo.

Learn more

Add to Cart

Complete Your Kitchen
# Requirements Table

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Text</th>
<th>Satisfied By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0 Microwave Oven</td>
<td>Overall system requirement.</td>
<td>Microwave Oven</td>
</tr>
<tr>
<td>2</td>
<td>1.2 Control Cooking</td>
<td>The oven shall control timing and power level.</td>
<td>Control Electronics</td>
</tr>
<tr>
<td>3</td>
<td>1.3 Provide Safety Enclosure</td>
<td>The oven shall provide an RF sealed compartment with latching door and rotating tray.</td>
<td>Enclosure</td>
</tr>
<tr>
<td>4</td>
<td>1.1 Generate RF Power</td>
<td>The oven shall produce 750W of RF power at the approved frequency.</td>
<td>RF Power Source</td>
</tr>
</tbody>
</table>
## Requirements Allocation

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Appliance</th>
<th>Control Electro</th>
<th>Low Voltage</th>
<th>Processor</th>
<th>WiFi Transceiver</th>
<th>Cook Energy</th>
<th>Enclosure</th>
<th>Enclosure Conn.</th>
<th>Microwave Oven</th>
<th>p1</th>
<th>Panel</th>
<th>Refrigerator</th>
<th>RF Power Source</th>
<th>WiFi Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- Satisfy
- Satisfy (Impl...
Block Definition Diagram

```
<table>
<thead>
<tr>
<th>Block</th>
<th>Domain</th>
<th>Control Electronics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>processor: Processor [1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>low Voltage Power Supply: Low Voltage Power Supply [1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wifi Transceiver: WiFi Transceiver [1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>InterlockVoltage: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KeySignal: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ControlVoltage: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DisplayValue: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WiFiMessage: WiFi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DisplayRemainingTime(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DisplayPowerLevel(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetPowerLevel(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetTime(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProcessControlPanelSignal(): Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generate5VPower(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send/ReceiveWiFi(): WiFi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block</th>
<th>Domain</th>
<th>RF Power Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ControlVoltage: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GenerateRFPower(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LockDoor(): Void</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block</th>
<th>Domain</th>
<th>Enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>InterlockVoltage: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LockDoor(): Void</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block</th>
<th>Domain</th>
<th>Control Panel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KeySignal: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DisplayDigit: Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SendUserInput(): Void</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DisplayValue(): Void</td>
</tr>
</tbody>
</table>
```
Element Specification for a Block

Specification of Block properties
Specify properties of the selected Block in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.
More Detailed Block Definition Diagram

- **Control Electronics**
  - **Processor**
    - Constraints:
      - Cook Energy: Cook Energy
    - Values:
      - ControlPanelSignal: Integer
      - DisplaySignal: Integer
      - PowerLevel: Power = 750 Watt (unit = Watt)
      - TimeValue: Time = 120 Second (unit = Second)
      - /CookEnergy: Energy (unit = Joule)
      - PartNumber: String = AK7345XP
    - Operations:
      - DisplayRemainingTime(): Void
      - DisplayPowerLevel(): Void
      - SetTime(): Void
      - SetPowerLevel(): Void
      - ProcessControlPanelSignal(): Integer
  - **Low Voltage Power Supply**
    - Operations:
      - GenerateSVPower(): Void
  - **WiFi Transceiver**
    - Values:
      - WiFiMsg: WiFi
Internal Block Diagram
ICD Tables

### Port Table

<table>
<thead>
<tr>
<th>#</th>
<th>Port Name</th>
<th>Port Type</th>
<th>Type Features</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enclosure</td>
<td>EnclosureConnector</td>
<td>LatchIn, LatchOut</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>Panel</td>
<td>Panel</td>
<td>PanelIn, PanelOut</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>RFGenerator</td>
<td>WiFi Interface</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>WiFi</td>
<td>WiFi Interface</td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>

### Connector Table

<table>
<thead>
<tr>
<th>#</th>
<th>Part A</th>
<th>Port A</th>
<th>Item Flow</th>
<th>Port B</th>
<th>Part B</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>processor: Processor [1]</td>
<td>LatchIn</td>
<td></td>
<td>LatchIn</td>
<td>EnclosureConnector</td>
</tr>
<tr>
<td>3</td>
<td>processor: Processor [1]</td>
<td>LatchOut</td>
<td></td>
<td>LatchOut</td>
<td>EnclosureConnector</td>
</tr>
<tr>
<td>6</td>
<td>processor: Processor [1]</td>
<td>RFGenerator</td>
<td>ControlVoltage</td>
<td>RFGenerator</td>
<td>Control Electronics</td>
</tr>
<tr>
<td>7</td>
<td>Control Electronics</td>
<td>WiFi: WiFi Interface</td>
<td></td>
<td>WiFi: WiFi Interface</td>
<td>WiFi Transceiver: WiFi Transceiver [1]</td>
</tr>
<tr>
<td>8</td>
<td>WiFi Transceiver: WiFi Transceiver</td>
<td>WiFi Transceiver</td>
<td>WiFiMsg</td>
<td></td>
<td>processor: Processor [1]</td>
</tr>
<tr>
<td>#</td>
<td>Name</td>
<td>Documentation</td>
<td>Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Appliance</td>
<td>This Domain models the primary processing and electronics resources of the microwave oven.</td>
<td>Type : Enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Manufacturer : Enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Model : String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AC Voltage : voltage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Control Electronics</td>
<td>This Domain models the component used to input and display control parameters.</td>
<td>WiFi : WiFi Interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>processor : Processor [1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>low Voltage Power Supply : Low Voltage Power Supply</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>wifi Transceiver : WiFi Transceiver [1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RFGenerator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Panel : Panel</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Enclosure : EnclosureConnector</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InterlockVoltage : Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KeySignal : Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ControlVoltage : Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Control Panel</td>
<td></td>
<td>KeySignal : Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DisplayDigit : Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Dishwasher</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Enclosure</td>
<td>This Domain models the mechanical housing that provides the cooking Safety Enclosure and mounts the other components of the oven.</td>
<td>InterlockVoltage : Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Low Voltage Power Supply</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conceptual Data Model

[Diagram showing the Conceptual Data Model with classes DataFoundationClass, ControlData, OperationalData, and MaintenanceData, and their relationships and properties such as values and operations.]
Logical Data Model

```
bdd [Package] e Data [ LDMControlData ]

«block»
«InfoElement»
«Abstract»
ControlData

«block»
«InfoElement»
PowerValue
values
Power : Power = 750 Watt {unit = Watt}

«block»
«InfoElement»
TimeValue
values
Time : Time = 120 Second {unit = Second}
```
Use Cases

1.0 Perform Cooking
- 1.1 Set and Display Power and Time
- 1.2 Control RF Power Generation
- 1.3 Generate RF Power
- 1.4 Operate Safety Interlock

2.0 Perform Maintenance and Repair
- 2.1 Run Diagnostics

Cook
Repairman
**Use Case Spec**

This Use Case models the behavior associated with accepting power and time value inputs via the Control Panel and displaying the current settings.

**Precondition:** Oven is Operational

**Trigger:** Cook presses control pad key

**Postcondition:** Entered values are set in the oven Processor and displayed on the Control Panel

**Data Objects:** Power Level, Time Value

**User Role:** Cook

**Scenario:** See Activity Diagram

**Allocated Requirements:** 1.2 Control Cooking

---

**Comment**

- Owner
- Applied Stereotype
- Owned Element
- Owning Element
- Annotated Element

**To Do**

- Element ID

**Documentation**

- Sync Element

---

Specify properties of the selected Comment in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.
Basic Behavior

act [Activity] Basic Operation [Basic Operation1]

«allocate» Control Panel

Use Case 1.1

Cook Input: Enter Time & Power

: Display Time Value

: Display Power Value

Cook Input: Press Start

Controller computes RF Power Source control voltage and applies it until timer reaches TimeValue.

«allocate» Control Electronics

Use Case 1.1

: Set Time Value

: Set Power Value

«block» InfoElement

TimeValue

«block» InfoElement

: PowerValue

Use Case 1.4

: Send Interlock Voltage

: Lock Door

: Set Interlock Discrete

Door Locked?

[No]

[Yes]

UC 1.2: Control RF Power Source

UC 1.3: Generate RF Power

«allocate» Enclosure

Use Case 1.4

«allocate» RF Power Source
Use Case Scenario

Place Food in Enclosure

UC 1.1 Set and Display Power and Time: 1.1

UC 1.2 Control RF Power Generation

UC 1.4 Operate Safety Interlock

UC 1.3 Generate RF Power

UC 1.4 Operate Safety Interlock

Remove Food
Detailed Use Case Scenario

act [Activity] 1.2 Control RF Power Generation

- in : PowerValue
  - Compute RF Power Source
  - Control Voltage
  - output : ControlVoltage

- in : TimeValue
  - Set Timer

- Apply Control Voltage
  - input : ControlVoltage

- Decrement Timer

- Timer = 0?
  - [Yes] Remove Control Voltage
  - [No]
Sequence Diagram (Interaction)

1. SendTimeValue
2. SendPowerValue
3. SetTimer
4. SendInterlockVoltage
5. LockDoor()
6. SendDoorLockDiscrete
7. StartTimer
8. ComputeControlVoltage
9. SendControlVoltage
10. GenerateRFPower()
11. RemoveInterlockVoltage
12. UnlockDoor

Loop:
[Timer <= TimeValue]
Actor Diagram
Summary

• MBSE is the state-of-the-practice for complex systems and enterprises
  o Development of new systems/enterprises to meet customer needs
  o Sustainment and upgrading of legacy systems
  o Equally valuable for process optimization

• Payoffs include
  o Support for balanced design to satisfy diverse stakeholder concerns
  o Cost and schedule savings through productivity and quality improvements
  o Better requirements management, configuration control, support for trade studies, support for specialty engineering, etc.
  o Support for initial and long-term system quality and integrity using Quality Attributes

• MBSAP implements the principles of MBSE in a proven methodology and framework
  o Complies with leading industry and Government frameworks and policies
  o Applied successfully in multiple programs

• MBSAP is defined, documented, tool-supported and ready for implementation
Preface

Systems and multiple design advancements have become mainstays of modern development, especially in the fields of engineering and technology. The aim of this edition is to provide a comprehensive guide to modern systems engineering and its applications. It is designed to be a valuable resource for professionals, educators, and students alike.

The thesis is based on the principles of systems engineering, with a focus on the development of complex systems. It covers the latest methodologies and tools used in the industry, as well as case studies that illustrate practical applications.

This Springer imprint is published by the registered company Springer Nature Switzerland AG. The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland.
Textbook Errata

Chapter 1 Exercises
#8 Reference should be to Table 1.2

Chapter 4 Exercises
#5 Reference should be to Figure 4.18
#6 and #7 Reference should be to Figure 4.20

Figure 4.3 should be replaced with the following:
Figure 4.7 should be replaced with the following:

Page 104: the acronym FR should be spelled out as Functional Requirement.

Section 5.5, page 174
Reference to Figure 4.20 should be to Figure 4.22.

Chap 5 Exercises
#6 Reference should be to Figure 5.9.
Figure 5.14 should be replaced with the following:

```
Figure 5.14 should be replaced with the following:
```

```
Figure 5.18 should be replaced with the following:
```

```
Figure 5.18 should be replaced with the following:
```
Figure 7.8 should be replaced with the following:

![Diagram 1]

Figure 7.9 should be replaced with the following:

![Diagram 2]
Figure 7.11 should be replaced with the following:

```
sd[Package]Services[ServiceBehavior]

<<block,:Participant>>
  :ServiceConsumer
    SendServiceRequest()
      Acknowledge()
    SendParameters()
      ParseRequest()

<<block,:Participant>>
  :ServiceProvider
    alt
    [RequestType=Primary]
      ProcessPrimaryProduct()
        SendProduct()  
          Acknowledge() 
        SendProduct()  
    [else]
    ProcessAlternateProduct()  
      SendProduct()  
        Acknowledge() 
```

Figure 7.12 should be replaced with the following:

```
ibd[Package]Services[ParticipantCapabilities]

<<ServicePoint>>
  Port1
  InterfaceDefinition

<<block,Participant>>
  ParticipantA
    <<block,Capability>>
      Capability1
        1
        1
    <<block,Capability>>
      Capability2

<<RequestPoint>>
  Port2
  InterfaceDefinition
```
Figure 8.6 should be replaced with the following:

Chap 11 Exercises
#10 reference should be to Figure 11.11.

Chap 12 Exercises
#6 reference should be to Question 5.

Figure C.18 should be replaced with the following:
Figure C.22 should be replaced with the following:

1 CollectionTaskQueueManager
  values
  CollectionTask: task
  operations
  GetCollectionTask(): task
  ManageQueue(): task

1 SensorCharacteristicsManager
  values
  RadarCharacteristics: data
  SIGINTCharacteristics: data
  EO-IRCharacteristics: data

1 ConstraintManager
  values
  PhysicalConstraint: data
  OperationalConstraint: data

1 TaskSensorPairingAndScheduling
  values
  CollectionTask: task
  TaskStatus: enum
  SensorTask: task
  SensorCharacteristics: data
  Constraint: data
  operations
  ComputePairing(): task
  SendTaskStatus(): void
  SendSensorTask(): void

1 CollectionTasks
  SensorInformation

Constraint Information

Outputs Sensor Task schedule to SensorPlansMgmt
Inputs status of currently active Sensor Tasks from SensorInterface

CollectionPlan
Inputs Collection Tasks from SensorPlansMgmt

SensorInterface

CollectionPlan

<<proxy>>

<<subdomain>>

block [SensorResourceMgmt][SensorResourceMgmt]