JOG SYSTEM ENGINEERING
GRAND SYSTEMS DEVELOPMENT
TRAINING PROGRAM INTRODUCTORY PRESENTATION

THE MODEL, THE TEXTUAL AND GRAPHICAL RAS, AND THE SPECIFICATION – A LOGICAL AND EFFECTIVE PROGRESSION

Presented By
Jeffrey O. Grady
JOG SYSTEM ENGINEERING
(858) 458-0121
jeff@jogse.com

VERSION 14.0 122A-1

Who Is Jeff Grady?

CURRENT POSITION

1993 – PRESENT

Owner, JOG System Engineering
System Engineering Assessment, Consulting, and Education Firm

PRIOR EXPERIENCE

1954 - 1964 U.S. Marine Corps

1964 - 1965 General Precision, Librascope Division
Customer Training Instructor, SUBROC and ASROC ASW Computing Systems

1965 - 1982 Teledyne Ryan Aeronautical
Field Engineer, AQM-34 Series Special Purpose Aircraft Systems
Project Engineer, System Engineer on Unmanned Aircraft Systems

1982 - 1984 General Dynamics Convair Division
System Engineer, Cruise Missile, Advanced Cruise Missile

1984 - 1993 General Dynamics Space Systems Division
Functional Engineering Manager Systems Development Department

FORMAL EDUCATION

SDSU BA Math, UCSD Systems Engineering Certificate,
USC MS Systems Management With Information Systems Certificate

INCOSE Founder, Fellow, ESEP, and First Elected Secretary

AUTHOR System Requirements Analysis (3), System Integration, System Validation and Verification, System Verification, System Engineering Planning and Enterprise Identity, System Engineering Deployment, System Synthesis, System Management

VERSION 14.0 122A-2 © JOG System Engineering

- **Your lecturer has many years of experience as a system engineer and engineering manager as a captive employee in three aerospace firms as well as 14 years of experience as a consultant and trainer in the system engineering field. His authorship includes seven textbooks and desk references in the field with two more being developed by the author and his publisher. One of those books is used in this training course as the textbook.**

The Principal Presentation References

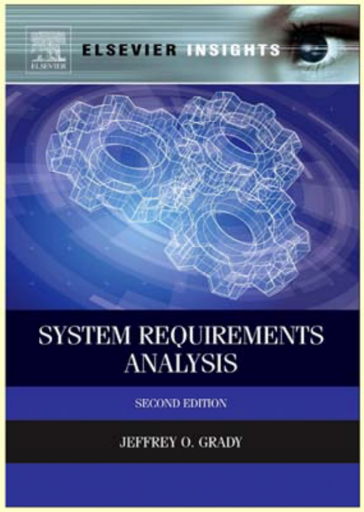
"System Requirements Analysis, 2nd Edition", Jeffrey O. Grady, Elsevier Academic Press, 2014

"The Model, the Textual and Graphical RAS, and the Specification – A Logical and Effective Progression", Jeffrey O. Grady, paper not yet published, 2013

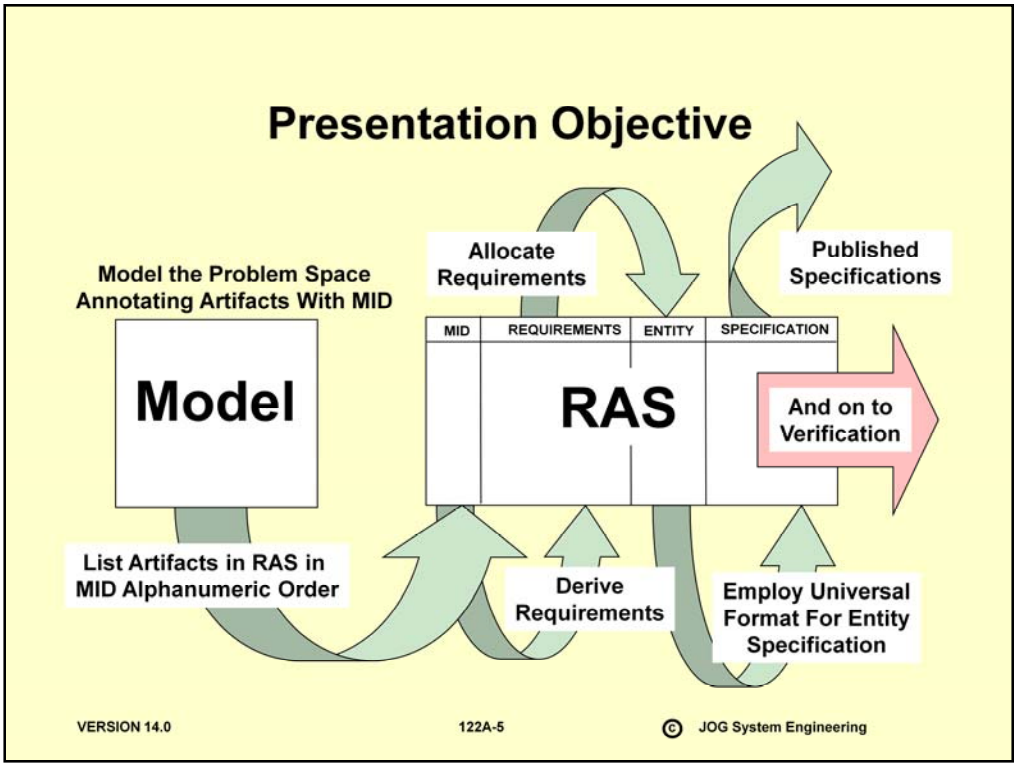
"Universal Architecture Description Framework (UADF)", Jeffrey O. Grady, Systems Engineering, The Journal of The International Council On Systems Engineering, Volume 12 Number 2, Summer 2009 (Best Paper 2009)

"Affordable Requirements Verification", Jeffrey O. Grady, INCOSE Insight, July 2013 (Volume 16, Issue 2)

New System Requirements Analysis Book in E-Book Format



VERSION 14.0 122A-4



- **We should recognize the simple structure of a requirement as stated in a sentence in a specification. The subject identifies the characteristics we wish to control. The verb is a form of the word shall to indicate the imperative nature of satisfying the requirement. Ideally, the sentence would include a numerical value and units related to the subject using one of the following phrases:**

  equal to (with a possible tolerance),

  less than,

  greater than,

  less than or equal to, or

  greater than or equal to.

- **Requirements statements can be stated in an even simpler fashion using primitive statements as noted in purple background on this chart.**
- **The hard work in requirements analysis is not to write requirements sentences, it is to know what to write them about and to know what numerical values to appeal to. Structured analysis is used to answer the first question and good engineering the second.**

What Is a System?

- Collection of product entities intended to achieve a specific function
- Immersed in an environment
- Product and environmental entities inter-related through interfaces
- Product and interface entities clearly defined in a set of specifications where all of the content has been derived though application of a model to the problem space

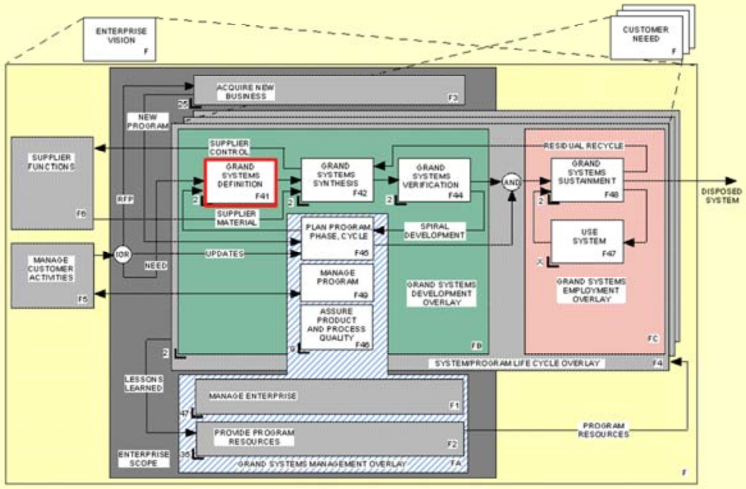
VERSION 14.0 122A-6

Systems Development

- Define the problem to be solved in a set of product and interface entity specifications
- Solve the problem through synthesis in a three-step process
  - Design
  - Procurement
  - Manufacturing
- Determine extent to which entities and the system comply with the content of the specifications through verification
- Manage the program well throughout its development period

VERSION 14.0 122A-7

Enterprise Common Process View of System Life Cycle



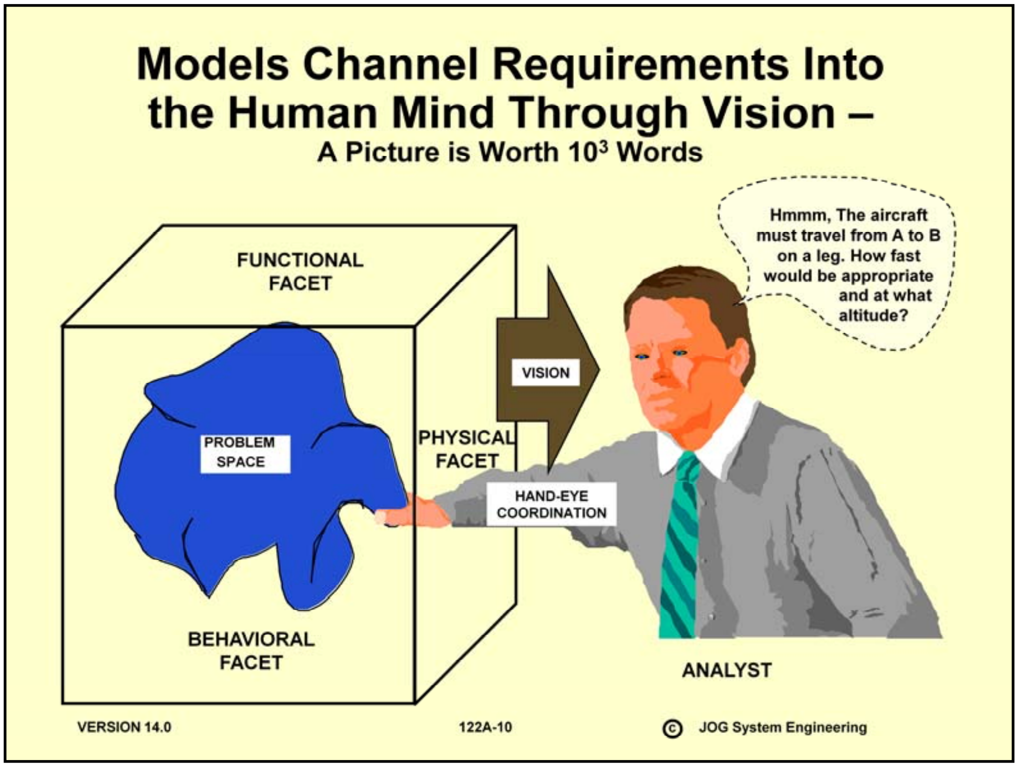
X: REFER TO PROGRAM SYSTEM DEFINITION DOCUMENT FOR EXPANSION

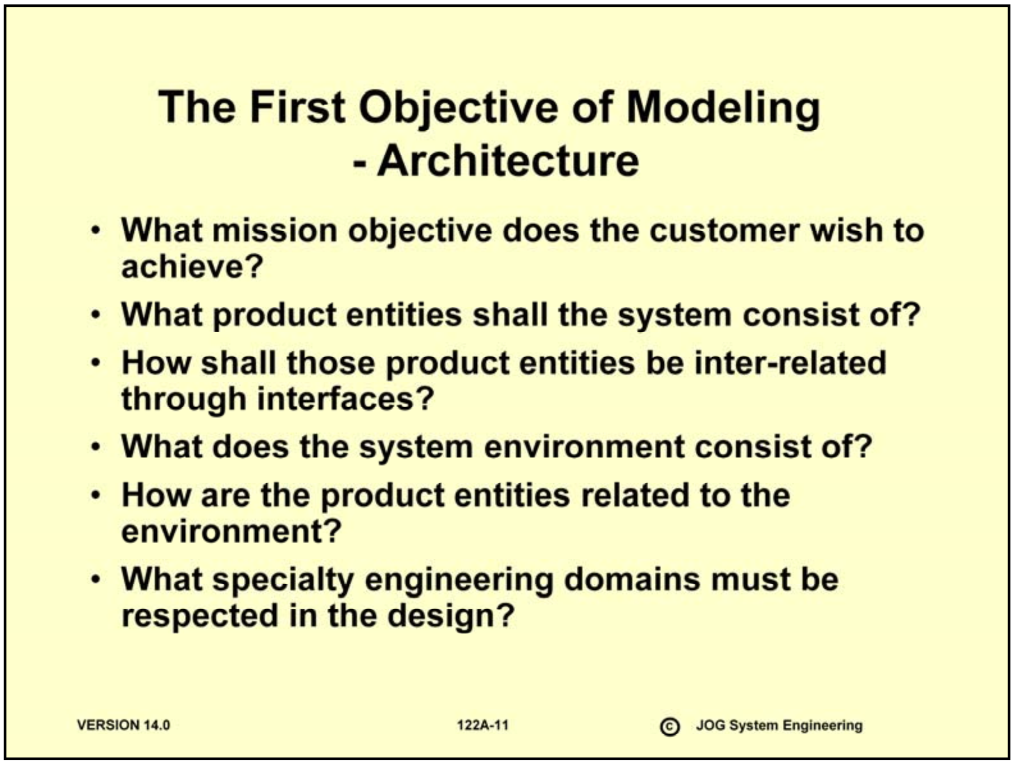
VERSION 14.0 122A-8 © JOG System Engineering

Major Problem on All Programs - Specification Content

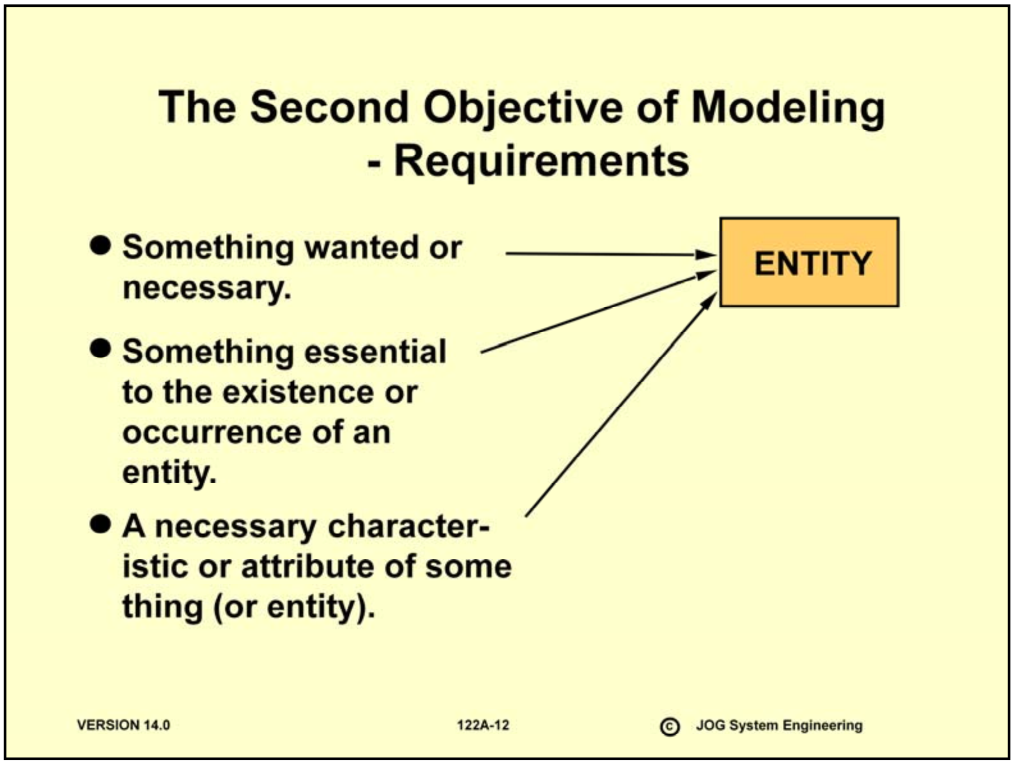
- Each specification contains the essential characteristics its product or interface entity must possess in the form of requirements
- An enterprise should derive the content of all specifications on all programs using a single comprehensive universal architecture description framework (UADF) model
  - Functional
  - MSA-PSARE
  - UML-SysML
  - UPDM maybe
- Adopt the Model-RAS-Specification Sequence using your selected UADF and a template coordinated with it

VERSION 14.0 122A-9

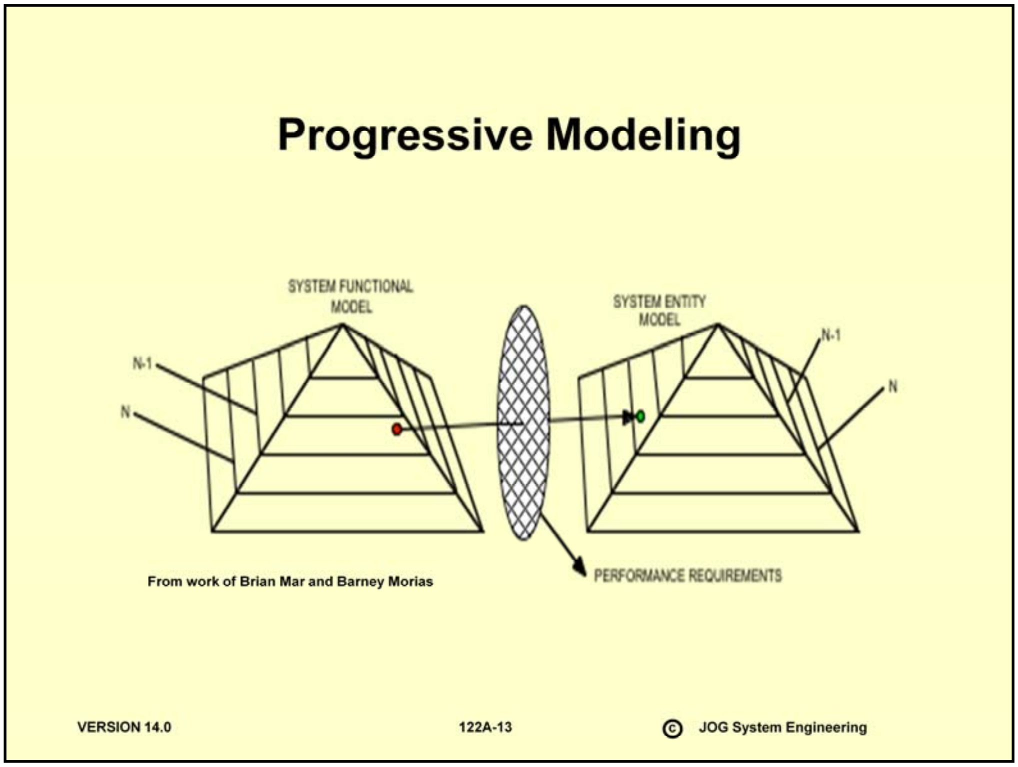

Models Channel Requirements Into the Human Mind Through Vision –
A Picture is Worth 10³ Words
Hmmm, The aircraft must travel from A to B on a leg. How fast would be appropriate and at what altitude?
FUNCTIONAL FACET
VISION
PROBLEM SPACE
PHYSICAL FACET
HAND-EYE COORDINATION
BEHAVIORAL FACET
ANALYST
VERSION 14.0
122A-10
© JOG System Engineering



- The word requirement is defined in the dictionary in these ways.
- It is important to recognize that requirements should include only the essential characteristics and not all of the characteristics of the design solution. The more requirements included in a specification the smaller the solution space available for the design team or designer. The specification for an item should contain the smallest set of requirements that capture the essential characteristics.
- It is possible to define a problem in so much detail that there may not be any solution space at all (null solution space) especially after you consider the small amount of money and time allocated to solve the design problem.



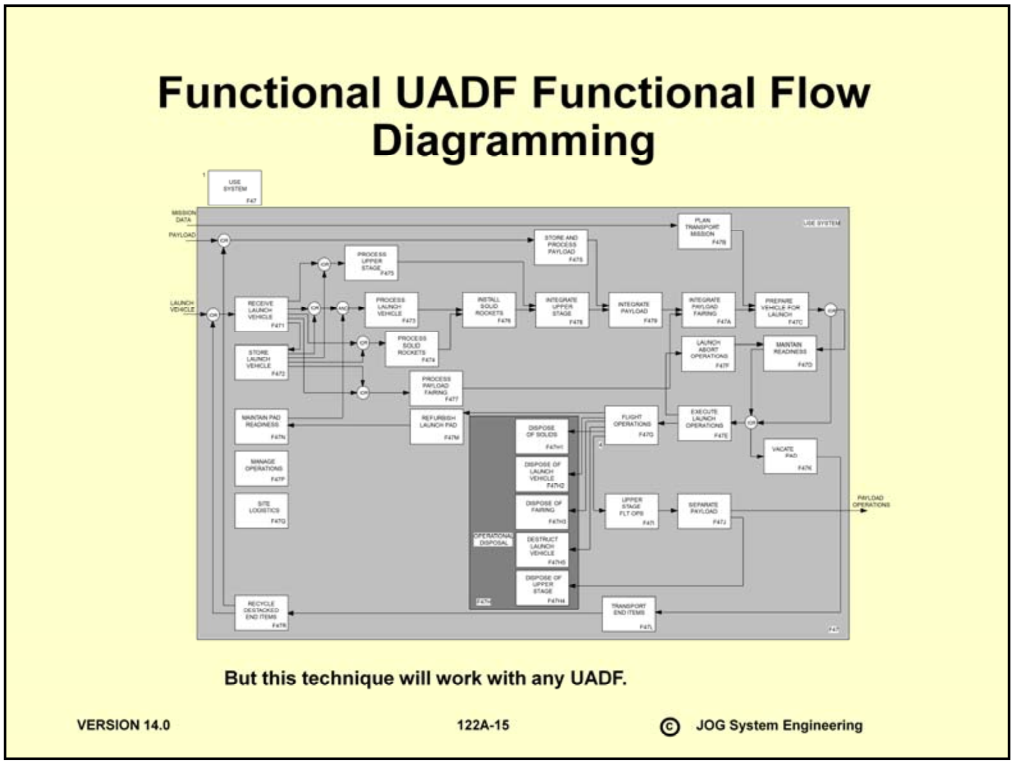
- The word requirement is defined in the dictionary in these ways.
- It is important to recognize that requirements should include only the essential characteristics and not all of the characteristics of the design solution. The more requirements included in a specification the smaller the solution space available for the design team or designer. The specification for an item should contain the smallest set of requirements that capture the essential characteristics.
- It is possible to define a problem in so much detail that there may not be any solution space at all (null solution space) especially after you consider the small amount of money and time allocated to solve the design problem.


Progressive Modeling
SYSTEM FUNCTIONAL MODEL
SYSTEM ENTITY MODEL
N-1
N
N-1
N
PERFORMANCE REQUIREMENTS
From work of Brian Mar and Barney Morias
VERSION 14.0
122A-13
© JOG System Engineering

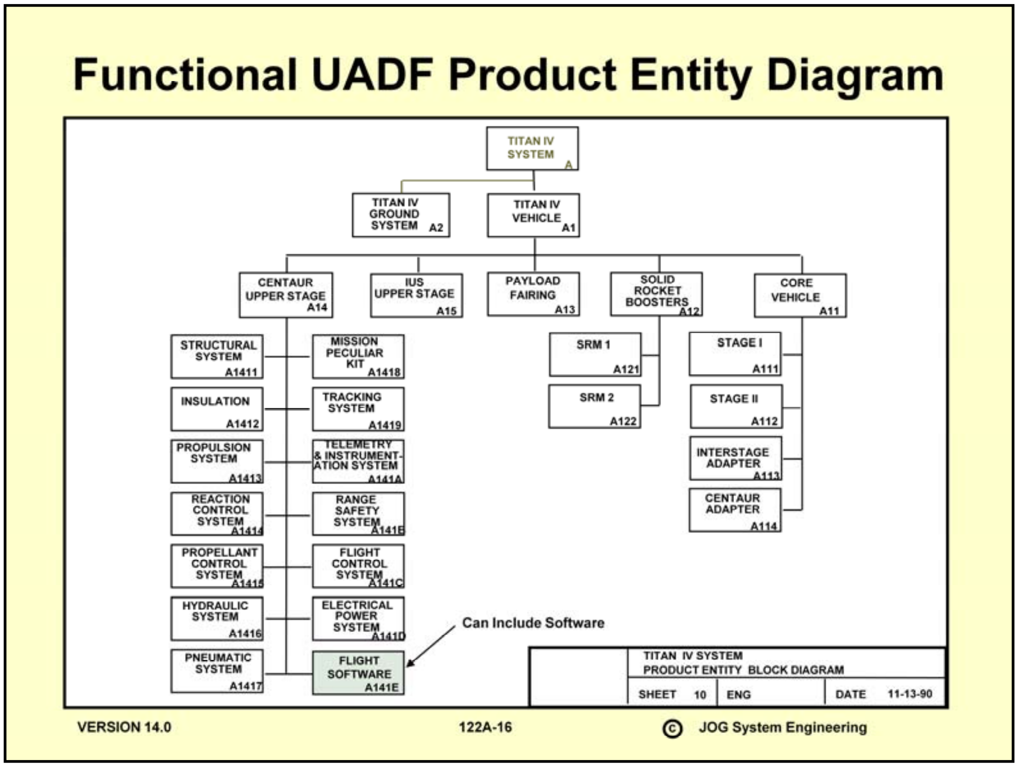
Three UADF Are Available

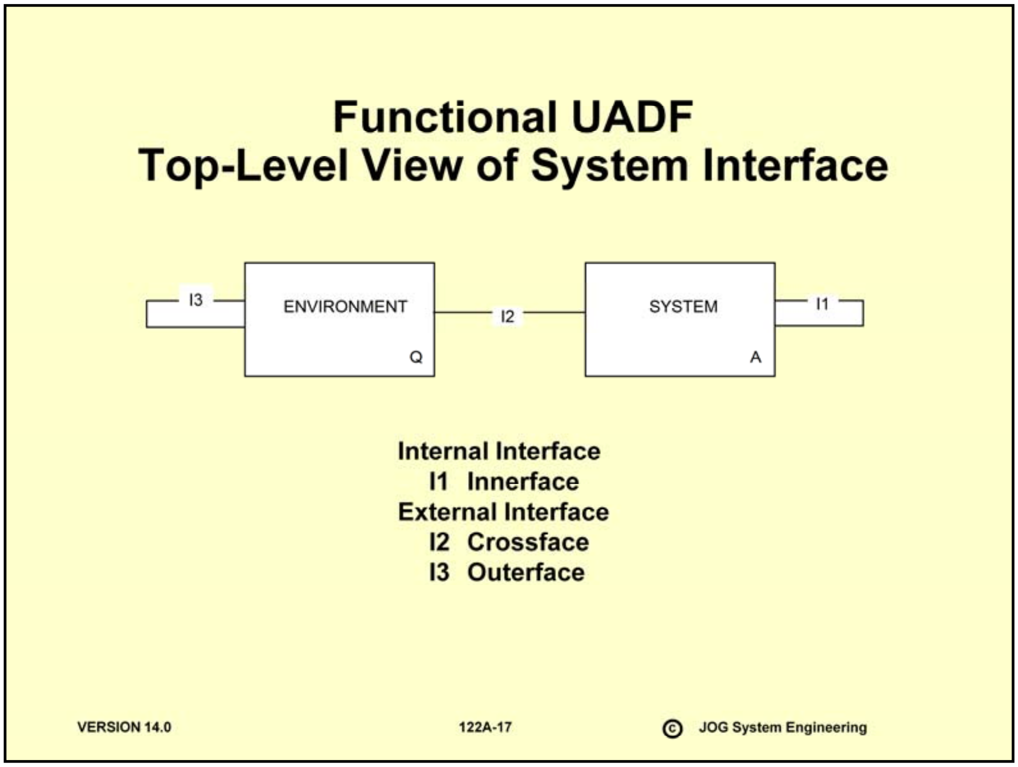
- A UADF is a comprehensive modeling approach in that it matters not how you will implement the solution in HW, SW, or people doing things
- One model is equally effective in HW and SW
- Pick one
  - Functional
  - MSA-PSARE
  - UML-SysML
  - UPDM maybe

VERSION 14.0 122A-14

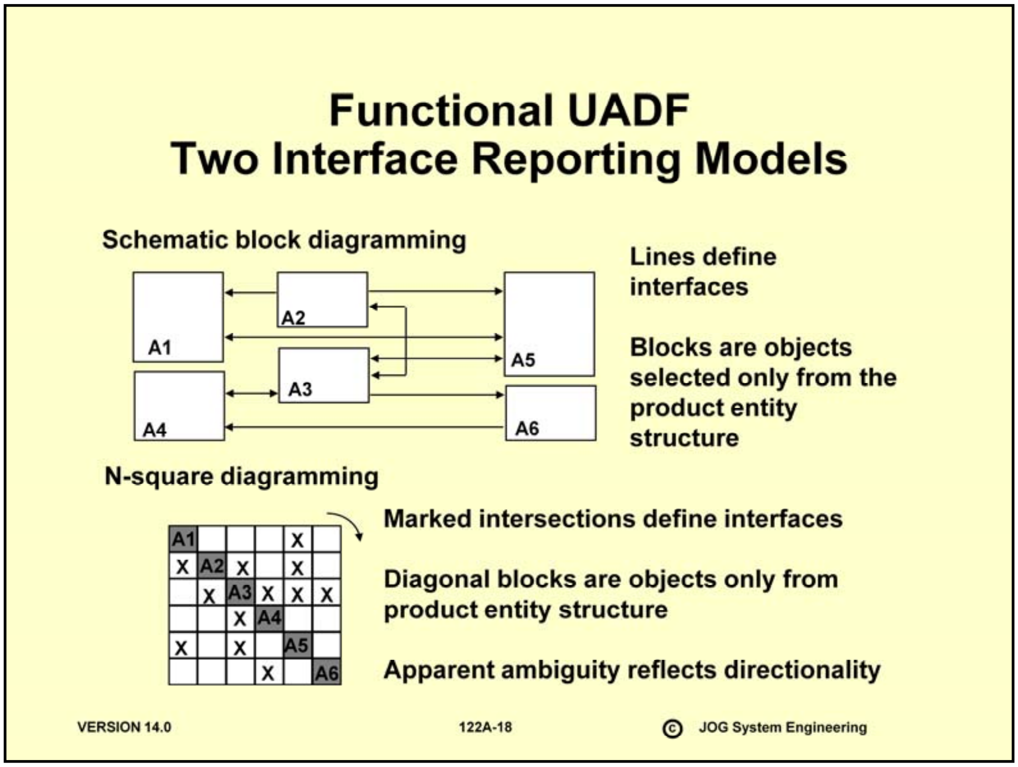


- Requirements appear in specifications. A specification contains all of the requirements appropriate for a single item in a system or the whole system. In this course we will cover how to identify the proper content for a specification.

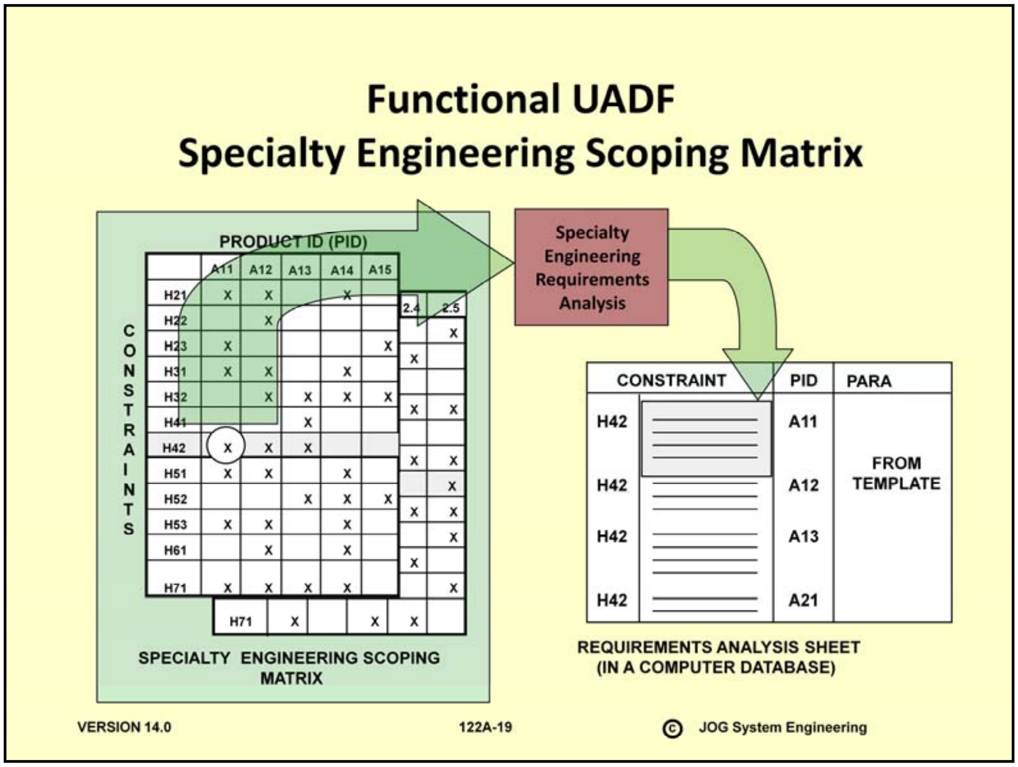

Functional UADF Product Entity Diagram
TITAN IV SYSTEM A
TITAN IV GROUND SYSTEM A2
TITAN IV VEHICLE A1
CENTAUR UPPER STAGE A14
IUS UPPER STAGE A15
PAYLOAD FAIRING A13
SOLID ROCKET BOOSTERS A12
CORE VEHICLE A11
STRUCTURAL SYSTEM A1411
INSULATION A1412
PROPULSION SYSTEM A1413
REACTION CONTROL SYSTEM A1414
PROPELLANT CONTROL SYSTEM A1415
HYDRAULIC SYSTEM A1416
PNEUMATIC SYSTEM A1417
MISSION PECULIAR KIT A1418
TRACKING SYSTEM A1419
TELEMETRY & INSTRUMENT-ATION SYSTEM A141A
RANGE SAFETY SYSTEM A141B
FLIGHT CONTROL SYSTEM A141C
ELECTRICAL POWER SYSTEM A141D
FLIGHT SOFTWARE A141E
Can Include Software
SRM 1 A121
SRM 2 A122
STAGE I A111
STAGE II A112
INTERSTAGE ADAPTER A113
CENTAUR ADAPTER A114
TITAN IV SYSTEM
PRODUCT ENTITY BLOCK DIAGRAM
SHEET 10 ENG DATE 11-13-90
VERSION 14.0
122A-16
© JOG System Engineering



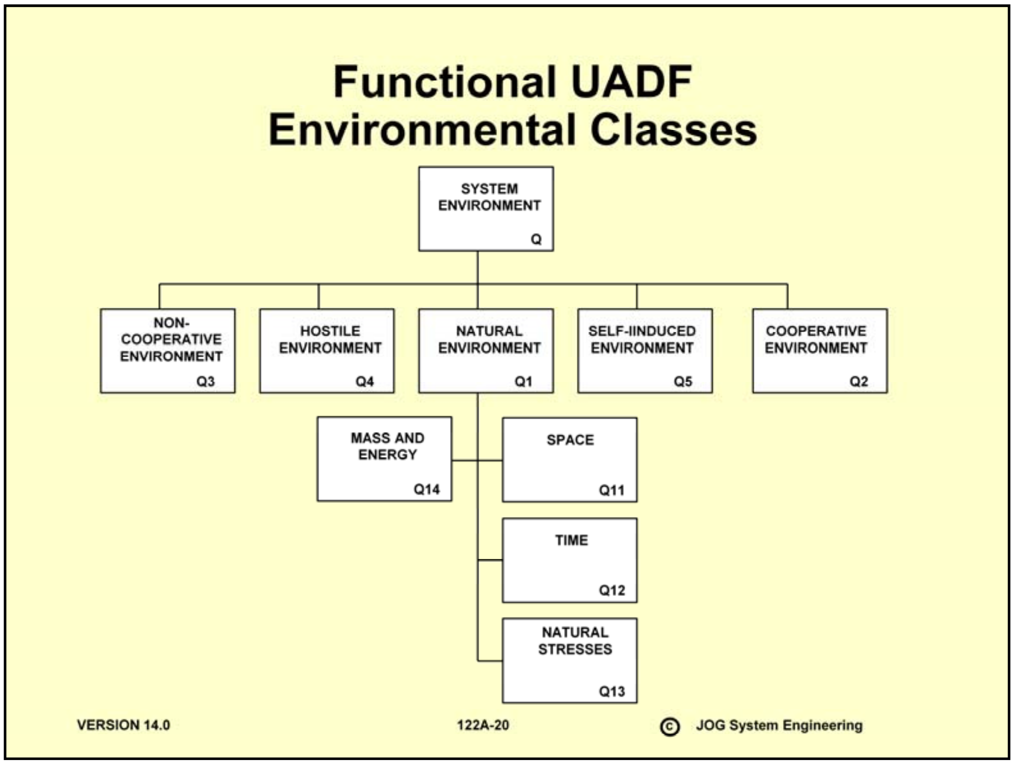
- The requirements analysis process starts at the top, the system, and continues downward as the lower tier entities are defined in the system. Here we see to different views of the system. The pair of block reflects the traditional system engineering view of a system interacting with its environment. The block named system is the top level block on a system product entity diagram that is expanded upon through the use of functional analysis.
- The context diagram view of a system employed in modern structured analysis expresses the interaction of the system with its environment using something called terminators that in UML are examined through use cases.
- In any case, a system is a collection of artifacts that interact within themselves and an environment to achieve planned results. Requirements analysis is an organized way of coming to an understanding about how the system should be constituted and what its requirements should be.



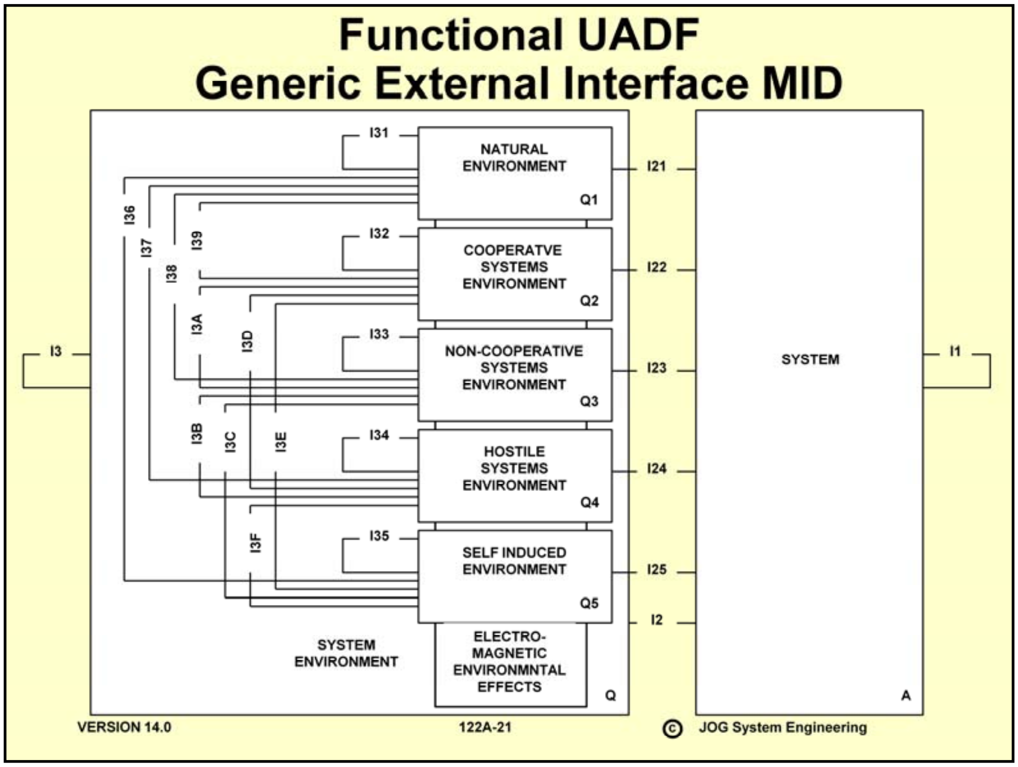
- **Two models have proven useful in identifying needed interfaces. The schematic block diagram, shown at the top, places blocks on the media surface that must be drawn from the product entity structure. These blocks are joined by directed line segments to identify needed interfaces. The engineer doing this work will consider what functionality has been allocated to these entities and base the identification of interfaces upon those allocations.**
- **Alternatively we could use an n-square diagram. The one shown here is offering the identical interface identification as the schematic block diagram. Since there are six objects in this analysis, the value of n in this case is 6. An n-square diagram is useful in all cases where one wants to explore the relationships between n objects and one is interested in the directionality of the relationships. In this case we have marked the clockwise relationship. This diagram is giving us precisely the same story as the schematic block diagram.**



- **A specialty engineering scoping matrix can be used to identify what specialty disciplines must accomplish requirements analysis on what entities. These entities then perform their own modeling work identifying requirements that flow into the RAS.**



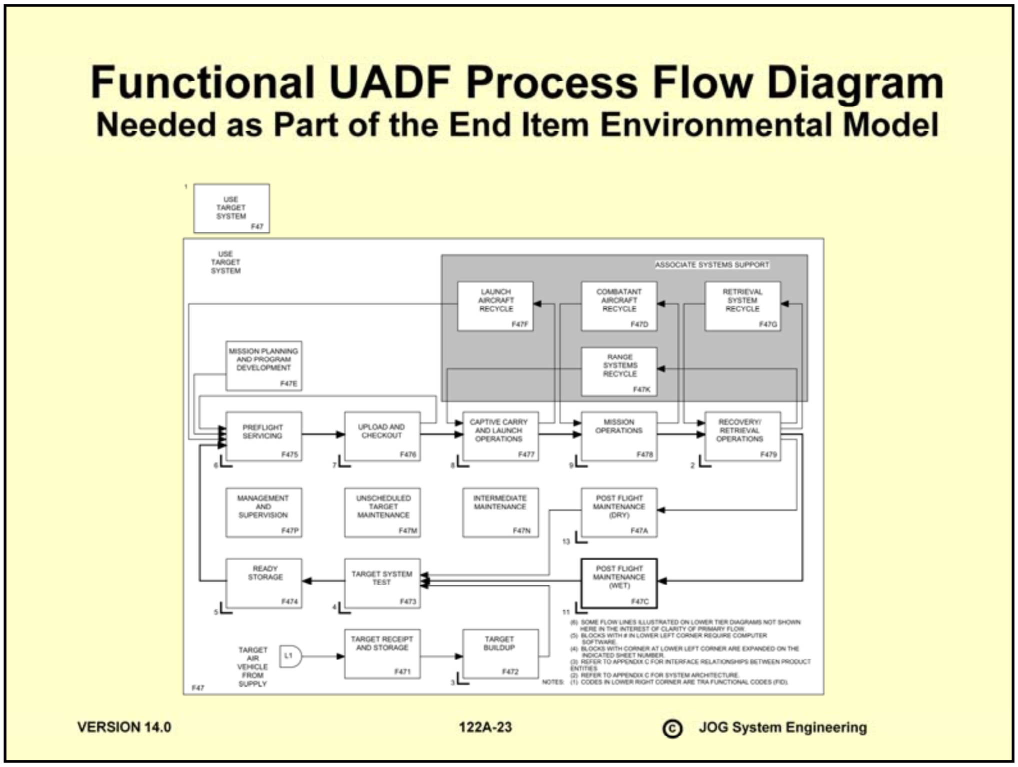
- **But, how do we decide what the right content is? The target is every essential characteristic and nothing else. How can we be sure we have identified every essential characteristic? How can we be sure we have not identified unnecessary content. Just how do we go about identifying the proper content of a specification?**
- **Thus you become aware of one of the two really hard parts of requirements analysis - deciding what to write them about.**


Functional UADF
Generic External Interface MID
I31
NATURAL
ENVIRONMENT
Q1
I21
I32
COOPERATVE
SYSTEMS
ENVIRONMENT
Q2
I22
I33
NON-COOPERATIVE
SYSTEMS
ENVIRONMENT
Q3
I23
I34
HOSTILE
SYSTEMS
ENVIRONMENT
Q4
I24
I35
SELF INDUCED
ENVIRONMENT
Q5
I25
I2
ELECTRO-
MAGNETIC
ENVIRONMNTAL
EFFECTS
SYSTEM
ENVIRONMENT
Q
I36
I37
I38
I39
I3A
I3B
I3C
I3D
I3E
I3F
I3
SYSTEM
A
I1
VERSION 14.0
122A-21
© JOG System Engineering

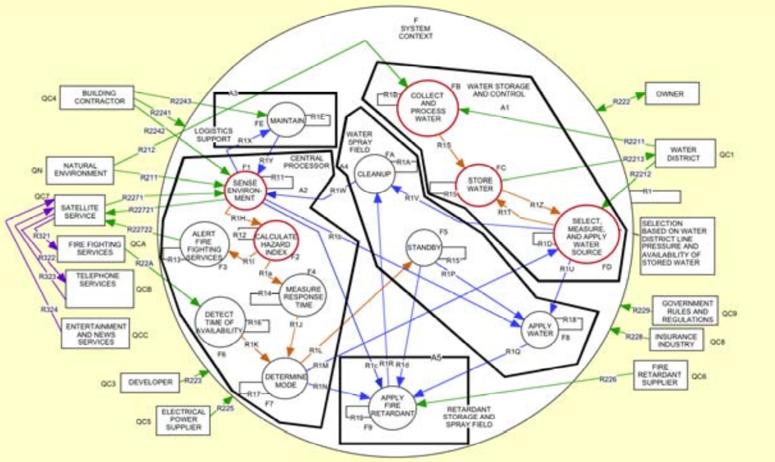
Functional UADF
Three Tier Environmental Modeling

- **System level using integrated union of tailored standards**
- **End item level using three dimensional service use profile**
  - Product entities
  - Environmental stresses
  - Process steps
- **Component level using end item zoning and mapping components to zones**
- **Possible need for an environmental sub system**

VERSION 14.0 122A-22

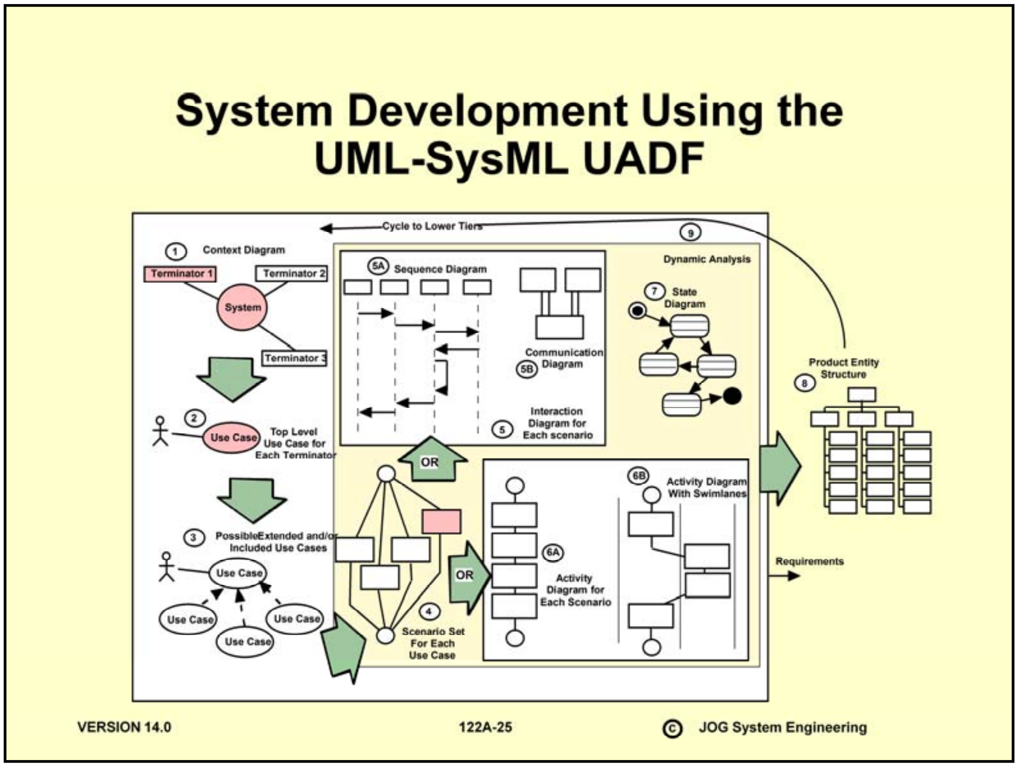

Functional UADF Process Flow Diagram
Needed as Part of the End Item Environmental Model
USE TARGET SYSTEM F47
USE TARGET SYSTEM
ASSOCIATE SYSTEMS SUPPORT
LAUNCH AIRCRAFT RECYCLE F47F
COMBATANT AIRCRAFT RECYCLE F47D
RETRIEVAL SYSTEM RECYCLE F47G
RANGE SYSTEMS RECYCLE F47K
MISSION PLANNING AND PROGRAM DEVELOPMENT F47E
PREFLIGHT SERVICING F475
UPLOAD AND CHECKOUT F476
CAPTIVE CARRY AND LAUNCH OPERATIONS F477
MISSION OPERATIONS F478
RECOVERY/ RETRIEVAL OPERATIONS F479
MANAGEMENT AND SUPERVISION F47P
UNSCHEDULED TARGET MAINTENANCE F47M
INTERMEDIATE MAINTENANCE F47N
POST FLIGHT MAINTENANCE (DRY) F47A
READY STORAGE F474
TARGET SYSTEM TEST F473
POST FLIGHT MAINTENANCE (WET) F47C
TARGET AIR VEHICLE FROM SUPPLY
L1
TARGET RECEIPT AND STORAGE F471
TARGET BUILDUP F472
F47
NOTES: (1) CODES IN LOWER RIGHT CORNER ARE TRA FUNCTIONAL CODES (FID).
(2) REFER TO APPENDIX C FOR SYSTEM ARCHITECTURE.
(3) REFER TO APPENDIX C FOR INTERFACE RELATIONSHIPS BETWEEN PRODUCT ENTITIES
(4) BLOCKS WITH CORNER AT LOWER LEFT CORNER ARE EXPANDED ON THE INDICATED SHEET NUMBER.
(5) BLOCKS WITH # IN LOWER LEFT CORNER REQUIRE COMPUTER SOFTWARE.
(6) SOME FLOW LINES ILLUSTRATED ON LOWER TIER DIAGRAMS NOT SHOWN HERE IN THE INTEREST OF CLARITY OF PRIMARY FLOW.
VERSION 14.0
122A-23
© JOG System Engineering

Systems Development Using the MSA-PSARE UADF

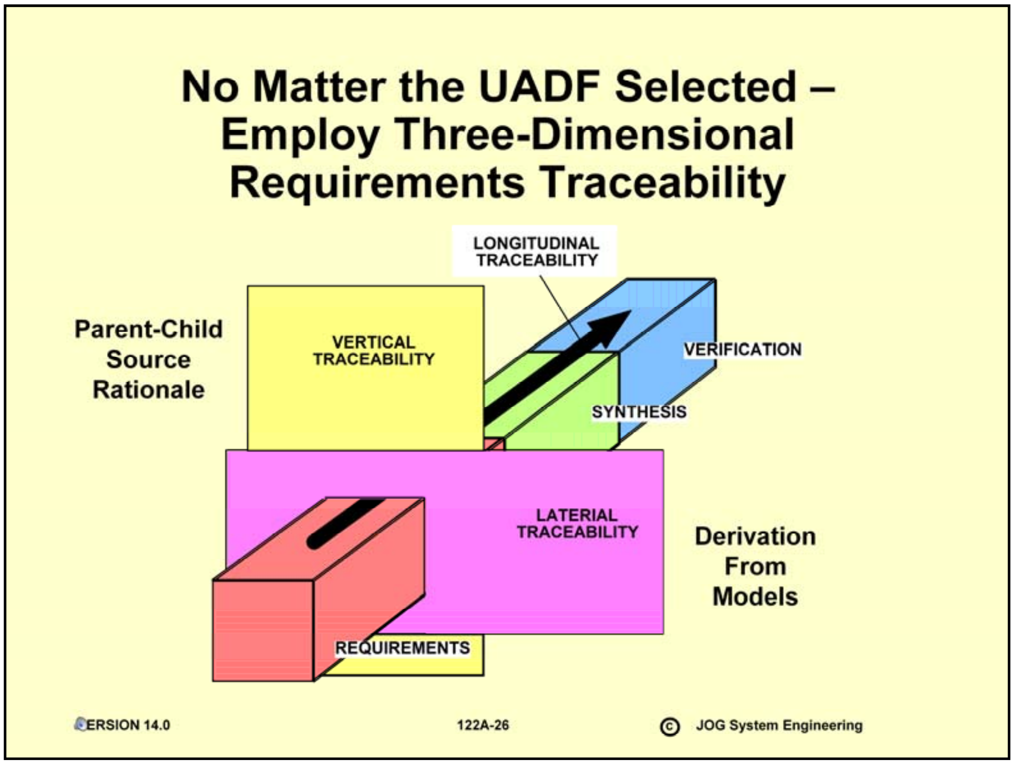


Assign Product Entity MID (A) to Super Bubbles
Assign Interface MID (I) to Functional Relations (R)

VERSION 14.0 122A-24



- **The approach encouraged at a little more detailed level starts with the use of a context diagram at the system level even though this is not part of UML it can be used to help organize the use cases. For each context diagram terminator we build a set of use cases (maybe one is sufficient but it could require more than one). For each use case we a build a scenario and keep looping until they are all built. For each scenario we translate it into an activity diagram and integrate the set of activity diagrams if necessary. We the allocate the activities to classifiers and orient the activity diagrams in swim lanes defined by the classifiers. We can now complete the dynamic analysis using some combination of communication, sequence, and state diagrams and decide on packaging into nodes, components, and objects to match the classifiers used in the activity diagram swim lanes and sequence and component diagram classifiers.**
- **We continue to repeat this process for other use cases.**
- **Once this analysis has identified objects we should be able to start writing code based on the requirements exposed in the analysis.**



- **Traceability has the effect of reducing program risk. More traceability results in less risk. How much risk can you stand?**
- **There are actually three kinds of traceability that we should apply, not just vertical or hierarchical parent-child traceability.**
- **Longitudinal traceability encourages that design and verification results are traceable to the requirements - what a concept!**
- **Lateral traceability encourages that the requirements included in specifications were identified as a result of particular structured analysis (modeling) processes.**

Suggested Specification Section 3 Template

TIMID ADVANCE

3.	REQUIREMENTS	3.4	Specialty Engineering Requirements
3.1	Modeling	3.5	Environmental Requirements
3.2	Performance Requirements	3.5.1	Natural Environment
3.3	Interface Requirements	3.5.2	Cooperative Environment
3.3.1	Internal Interfaces (I1)	3.5.3	Non-Cooperative Environment
3.3.2	External Interfaces (I2)	3.5.4	Hostile Environment
3.3.3	Outside Interfaces (I3)	3.5.5	Self-Induced Environment

AGGRESSIVE ADVANCE

3.	REQUIREMENTS	3.3.2.1	Natural Environment
3.1	Modeling	3.3.2.2	Cooperative Systems Environment
3.2	Performance Requirements	3.3.2.3	Non-Cooperative Environment
3.3	Interface Requirements	3.3.2.4	Hostile Environment
3.3.1	Internal Interfaces	3.3.2.5	Self-Induced Environment
3.3.2	External Interfaces	3.4	Specialty Engineering Requirements

Unique Modeling Artifact Identification To Support Lateral Traceability

MID	MEANING	PARA	DEPT	PREFERRED MODEL
A	Product Entity	3.1	331	Product Entity Block Diagram
F	Functionality	3.1	331	Functional Flow Diagramming
H	Specialty Engineering Domain	3.4	331	Specialty Engineering Scoping Matrix
H1	Engineering Domain	3.4.1	3XX	-
H11	Aerodynamics	3.4.1.1	321	Modeling and Simulation
H12	Thermodynamics	3.4.1.2	322	Thermodynamic Analysis
H13	Structural Integrity	3.4.1.3	323	Modeling and Simulation
H14	Structural Statics	3.4.1.4	323	Modeling and Simulation
H15	Structural Dynamics	3.4.1.5	323	Modeling and Simulation
H2	Logistics Domain	3.4.2	341	Functional Flow Diagramming
I	Physical Interface	3.3	331	N-Square Diagram
I1	Internal Interface	3.3.1	331	N-Square Diagram
I2	External Interface	3.3.2	331	N-Square Diagram
I3	Outside Interface	3.2.3	331	N-Square Diagram
J	Functional Interface	NA	331	N-Square Diagram
P	Process	-	-	Process Flow Diagram
Q	Environment	3.5	331	Three Tier Model
Q1	Natural Environment	3.5.1	331	Standards
Q11	Space	3.5.1.1	331	Mission Analysis and Packaging
Q12	Time	3.5.1.2	331	Time Lines
Q13	Natural Stresses	3.5.1.3	331	Standards
Q2	Cooperative Environment	3.3.2	331	N-Square Diagram
Q3	Non-Cooperative Environment	3.3.3	331	Threat Analysis
Q4	Hostile Environment	3.3.4	331	Threat Analysis
Q5	Self-Induced Environment	3.3.5	331	No Specific Model
R	Requirement	3	3XX	-

VERSION 14.0 122A-28

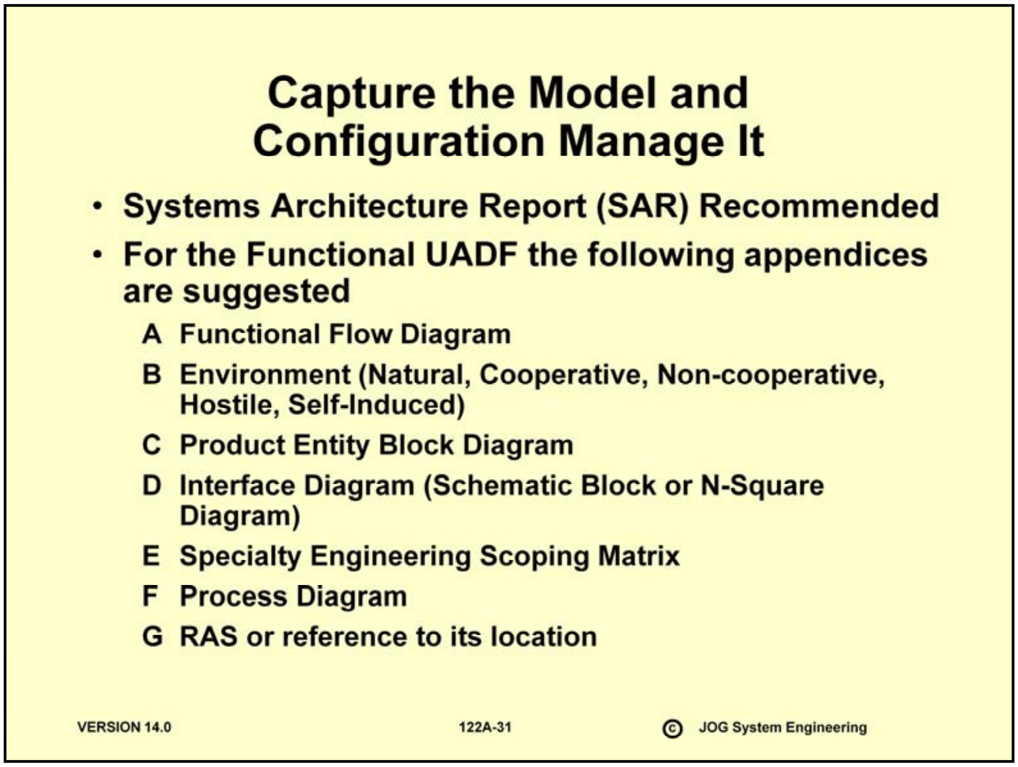
RAS-Complete In Table Form

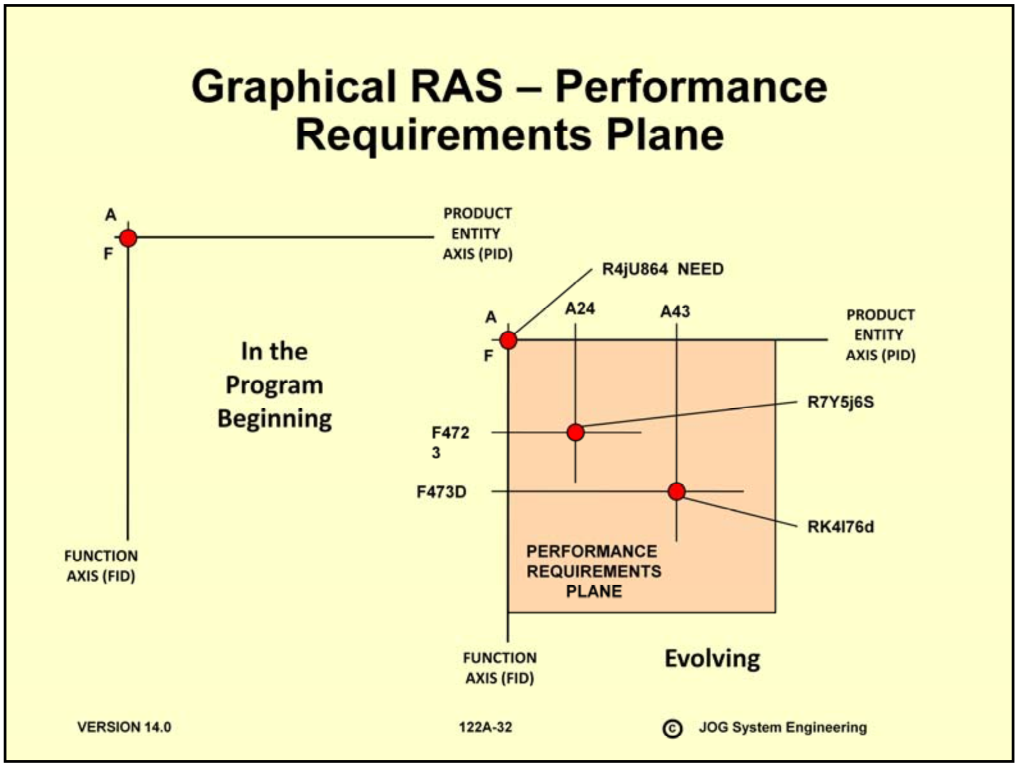
MODEL ENTITY		REQUIREMENT ENTITY		PRODUCT ENTITY		DOCUMENT ENTITY	
MID	MODEL ENTITY NAME	RID	REQUIREMENT	PID	ITEM NAME	PARA	TITLE
F47	Use System			A	Product System		
F471	Deployment Ship Operations			A	Product System		
F4711	Store Array Operationally	RXR67	Storage Volume < 10 ISO Vans	A1	Sensor Subsystem		
H	Specialty Engineering Disciplines			A	Product System		
H11	Reliability	REW34	Failure Rate < 10 x 10-6	A1	Sensor Subsystem	3.1.5	Reliability
H11	Reliability	RG31R	Failure Rate < 3 x 10-6	A11	Cable	3.1.5	Reliability
H11	Reliability	RFYH4	Failure Rate < 5 x 10-6	A12	Sensor Element	3.1.5	Reliability
H11	Reliability	RG8R4	Failure Rate < 2 x 10-6	A13	Pressure Vessel	3.1.5	Reliability
H12	Maintainability	R6GHU	Mean Time to Repair < 0.2 Hours	A1	Sensor Subsystem	3.1.6	Maintainability
H12	Maintainability	RU9R4	Mean Time to Repair < 0.4 Hours	A11	Cable	3.1.6	Maintainability
H12	Maintainability	RJ897	Mean Time to Repair < 0.2 Hours	A12	Sensor Element	3.1.6	Maintainability
H12	Maintainability	R9D7H	Mean Time to Repair < 0.1 Hours	A13	Pressure Vessel	3.1.6	Maintainability
I	System Interface			A	Product System		
I1	Internal Interface			A	Product System		
I11	Sensor Subsystem Innerface			A1			
I181	Aggregate Signal Feed Source Impedance	RE37H	Aggregate Signal Feed Source Impedance= 52 ohms + 2 ohms	A1	Sensor Subsystem		
I181	Aggregate Signal Feed Load Impedance	RE37I	Aggregate Signal Feed Load Impedance= 52 ohms + 2 ohms	A4	Analysis and Reporting Subsystem		
I2	System External Interface			A	Product System		
Q	System Environment			A	Product System		
QH	Hostile Environment			A	Product System		
QI	Self-Induced Environmental Stresses			A	Product System		
QN	Natural Environment			A	Product System		
QN1	Temperature	R6D74	-40 degrees F< Temperature < +140 degrees F	A	Product System		
QX	Non-Cooperative Environmental Stresses			A	Product System		

VERSION 14.0 122A-29 © JOG System Engineering

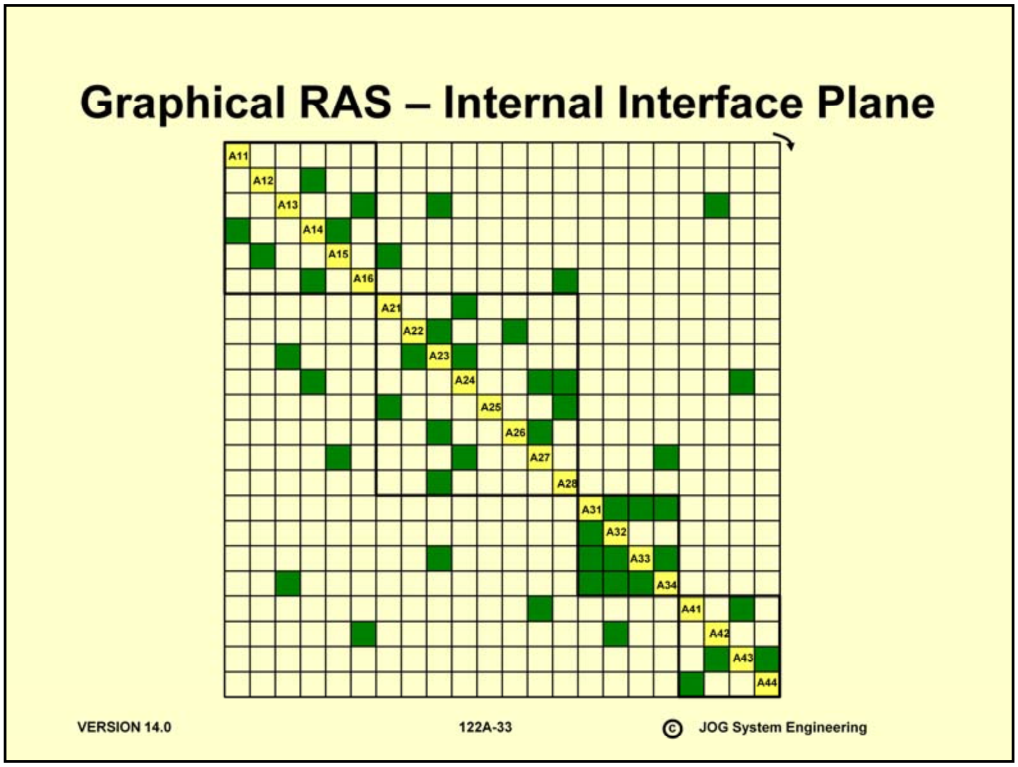
The Requirements Analysis Sheet (RAS)

- Tabular RAS in a computer database from which specifications may be printed is needed on every program
- Graphical RAS will be used in this presentation to explain the content and loading the tabular RAS from models
- In this presentation the functional UADF modeling artifacts are used in building the graphical RAS but the idea is compatible with the other two UADF as well

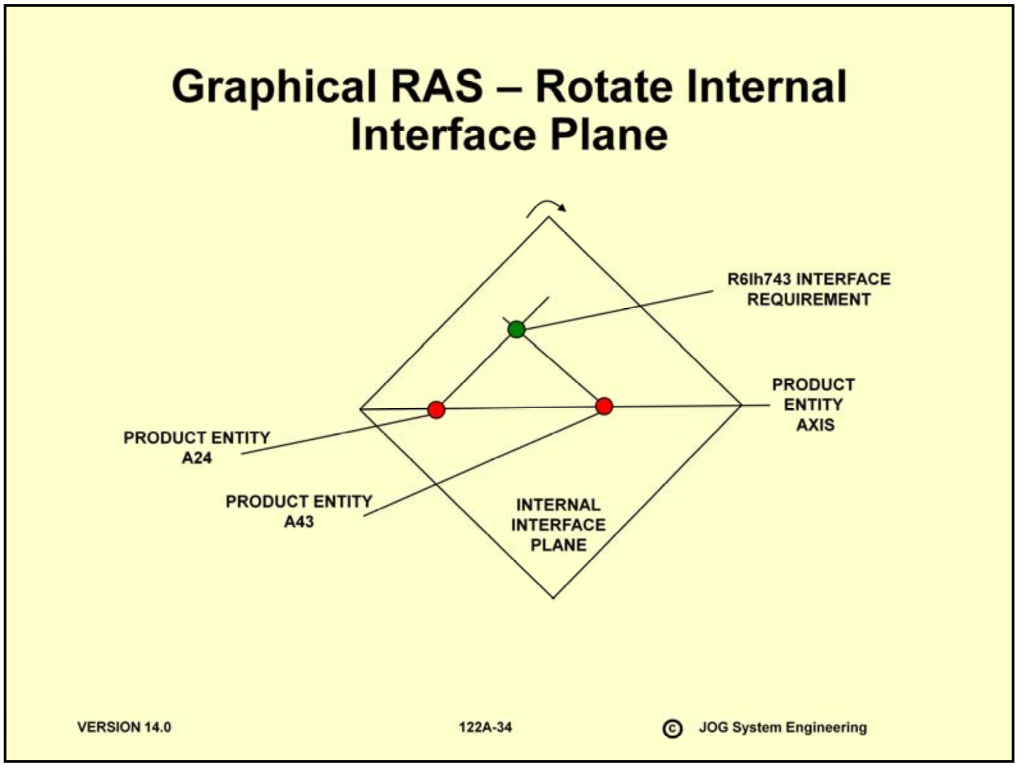

Capture the Model and
Configuration Manage It
• Systems Architecture Report (SAR) Recommended
• For the Functional UADF the following appendices are suggested
A Functional Flow Diagram
B Environment (Natural, Cooperative, Non-cooperative, Hostile, Self-Induced)
C Product Entity Block Diagram
D Interface Diagram (Schematic Block or N-Square Diagram)
E Specialty Engineering Scoping Matrix
F Process Diagram
G RAS or reference to its location
VERSION 14.0
122A-31
© JOG System Engineering



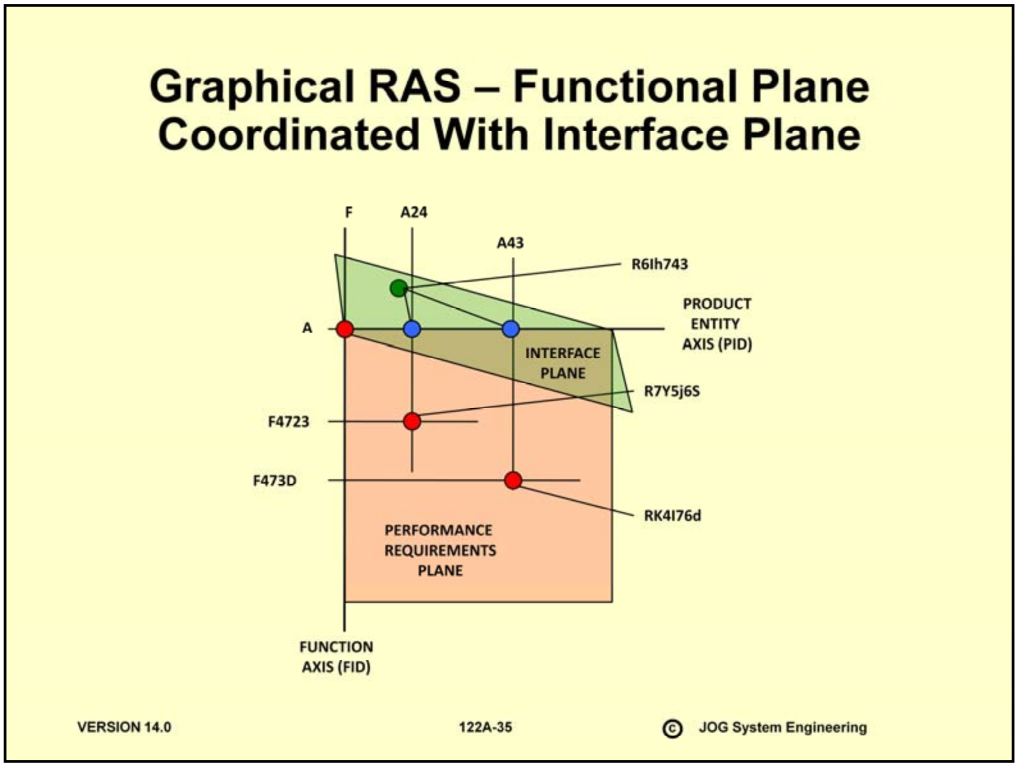
- **We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.**
- **Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.**



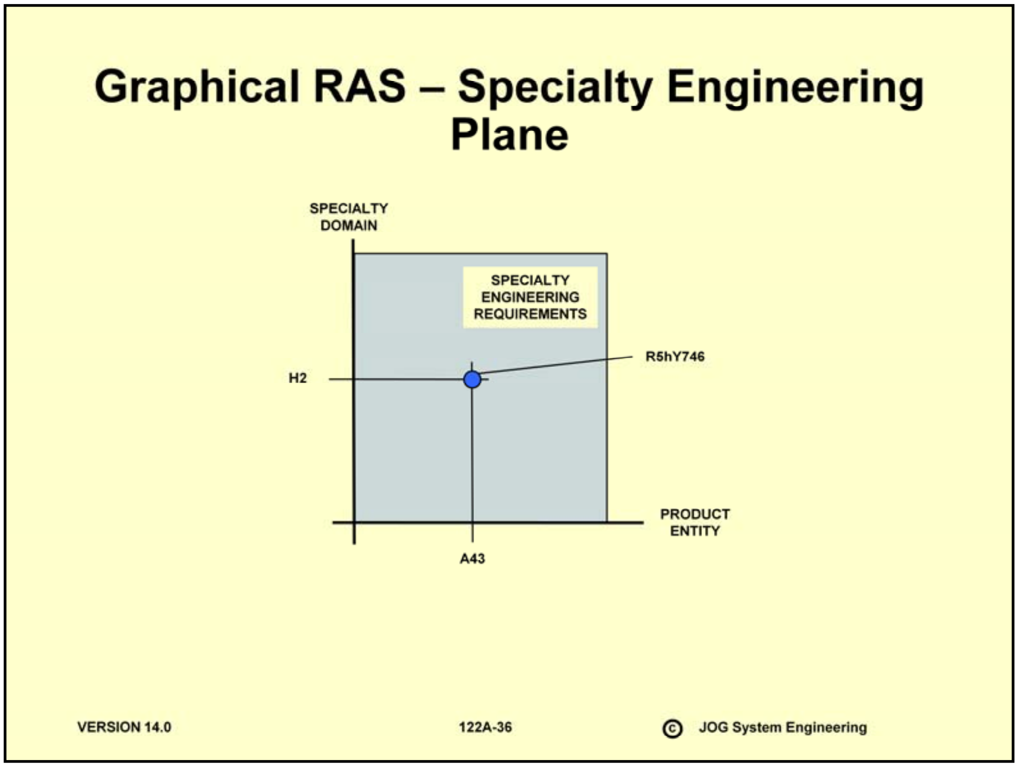
- We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.
- Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.



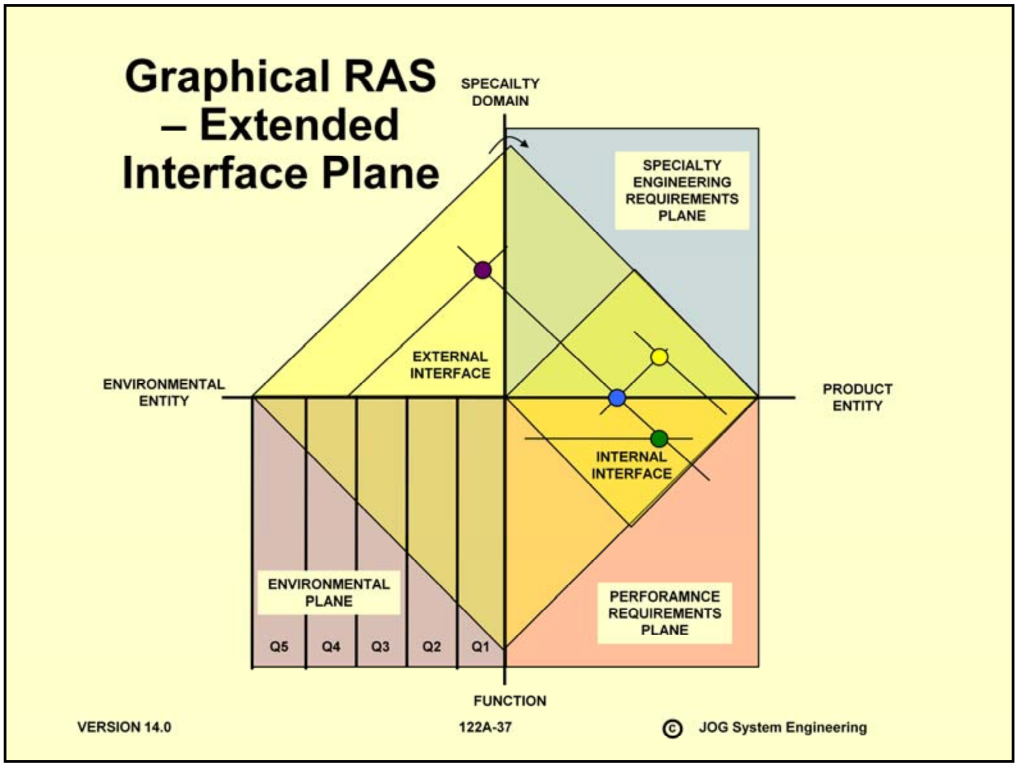
- **We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.**
- **Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.**



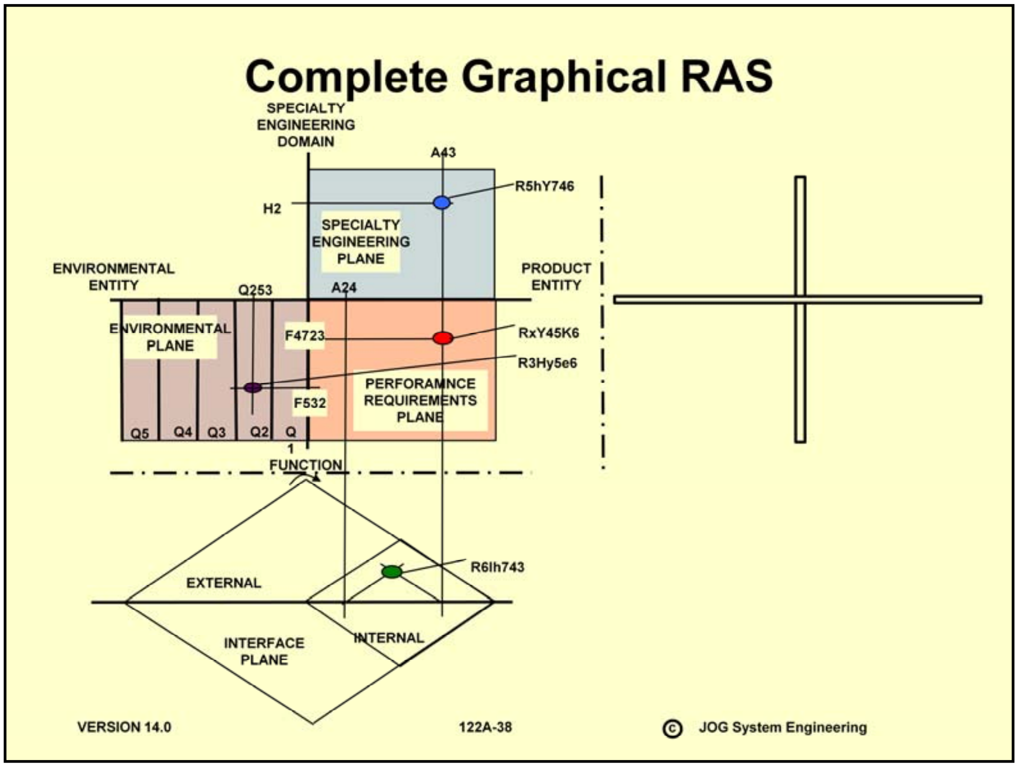
- We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.
- Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.



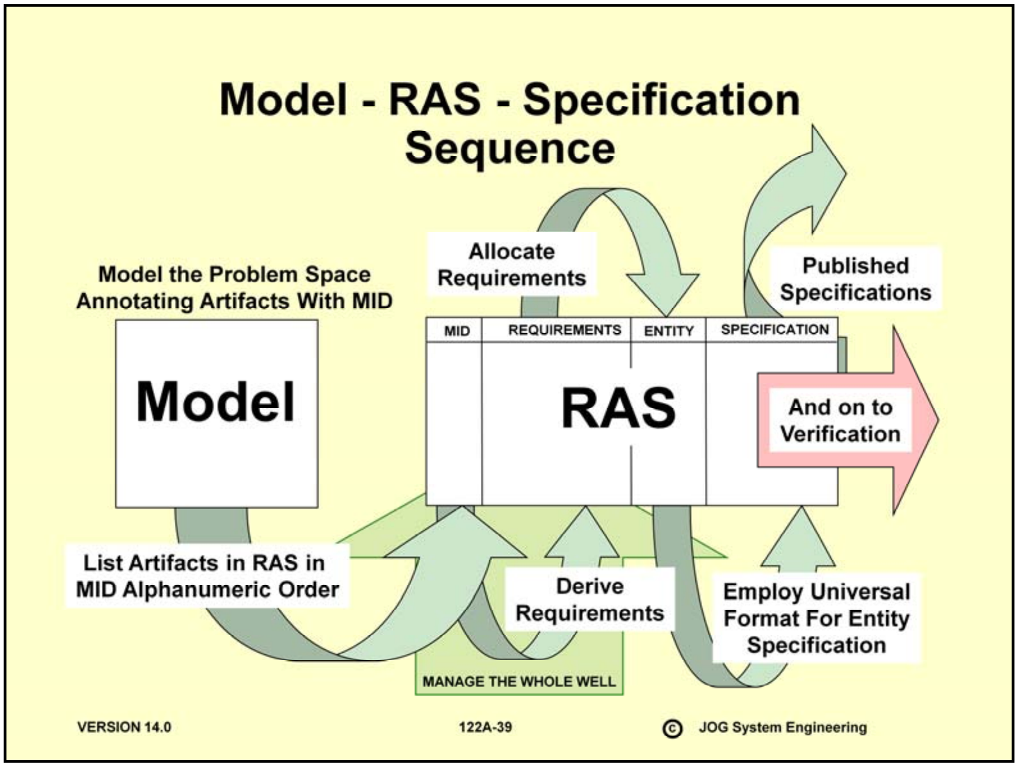
- **We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.**
- **Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.**



- We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.
- Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.



- We will quickly discuss an organized way of accomplishing this work using models from which we derive the structure and characteristics of the system under development and in the process identify the essential characteristics of the system entities that are transformed into full English sentences as requirements and placed in context with a planned template. The specifications thus formed are reviewed and published and become the basis for design and development.
- Most programs do not make an effort to capture the results of their modeling work unless it is captured in a computer tool and the case will be made that we should save these models for future work.



- **We should recognize the simple structure of a requirement as stated in a sentence in a specification. The subject identifies the characteristics we wish to control. The verb is a form of the word shall to indicate the imperative nature of satisfying the requirement. Ideally, the sentence would include a numerical value and units related to the subject using one of the following phrases:**

  equal to (with a possible tolerance),

  less than,

  greater than,

  less than or equal to, or

  greater than or equal to.

- **Requirements statements can be stated in an even simpler fashion using primitive statements as noted in purple background on this chart.**
- **The hard work in requirements analysis is not to write requirements sentences, it is to know what to write them about and to know what numerical values to appeal to. Structured analysis is used to answer the first question and good engineering the second.**

Prescription For the Enterprise That Has Not Yet Reached Perfection

1. Adopt a UADF and insist that all persons doing architecture development and requirements analysis work use it.
2. Adopt a way of uniquely identifying all modeling artifacts from which requirements may be derived.
3. Adopt a means by which personnel may capture modeling and specification content such that they may be configuration managed. There are not any computer tools known to the author that could capture all of the modeling and documentation features covered in the paper but one could build a simple text-oriented database linked to hand drawn or computer application graphics modeling artifacts.
4. Adopt a means for personnel to accomplish modeling work and retention of masters in the formal system baseline documentation.
5. Adopt a set of specification templates coordinated with modeling.
6. Establish a policy such as Table 1 of the supporting text suggests that clearly assigns responsibility for all specification content to personnel from specific functional departments on all programs.
7. Prepare a written document telling how this work is to be done on programs.
8. Train all personnel who have a role in this work in the appropriate parts of it assigned to their functional department.
9. Establish a quality assurance means that will assure that the work is accomplished in accordance with the prepared instructions and contractual requirements on programs.

