JOG SYSTEM ENGINEERING, INC
GRAND SYSTEMS DEVELOPMENT TRAINING PROGRAM
PRESENTATIONS

UNIVERSAL ARCHITECTURE DESCRIPTION FRAMEWORK

Presented By
Jeffrey O. Grady
President JOG System Engineering, Inc.
jeff@jogse.com (858) 458-0121

Who Is Jeff Grady?

CURRENT POSITION

1993-Preset President, JOG System Engineering, Inc.
System Engineering Assessment, Consulting, and Education Firm

PRIOR EXPERIENCE

1954 - 1964 U.S. Marines

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

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

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

1984 - 1993 General Dynamics Space Systems Division
Engineering Manager, System Development

FORMAL EDUCATION

SDSU - BA Math; UCSD - System Engineering Certificate
USC - MS Systems Management With Information Systems Certificate

INCOSE First Elected Secretary, Fellow, Founder, Certified System Engineering Professional

AUTHOR System Requirements Analysis (2), System Verification, System Integration, System Validation and Verification, System Engineering Planning and Enterprise Identity, System Engineering Deployment

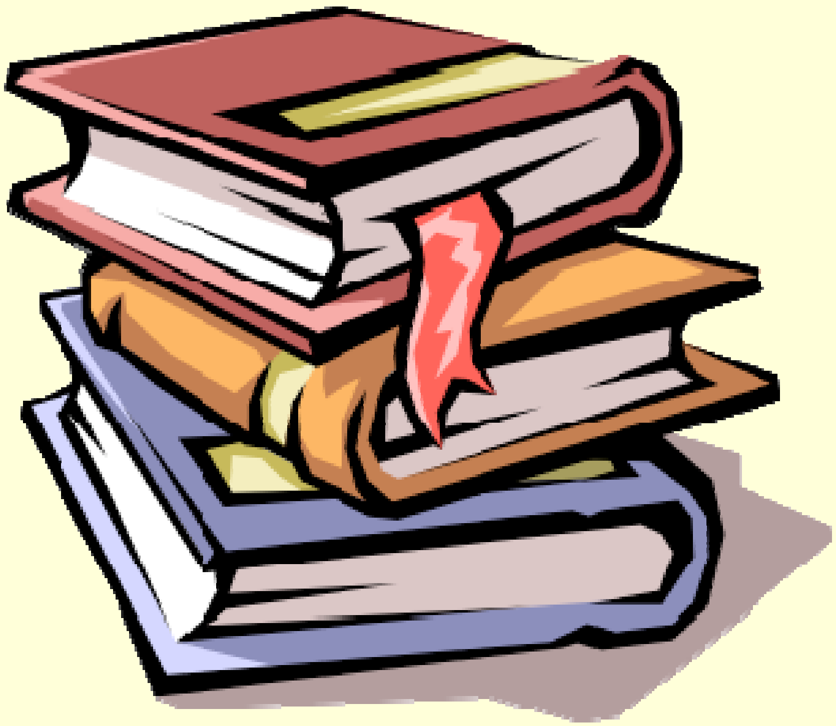
A Proposed Objective and a Means

- We wish to create effective and affordable systems that satisfy our needs.
- An effective way to do this is to follow a three step process within the context of a sound program management infrastructure
  - Define the problem in specifications
  - Solve the problem through synthesis including product design, procurement, and manufacturing
  - Prove that what we created satisfies the requirements that drive the synthesis work – verification
- Simple but not so easy to do

Some Fundamentals In Building Good Performance Specifications

- **A requirement is an essential characteristic appropriate to the development of a design**
- **A good specification captures all of the essential characteristics for a given item with no extraneous content that will drive cost but not value**
- **Synthesis work should be preceded by release of a good performance specification**

To Emphasize!



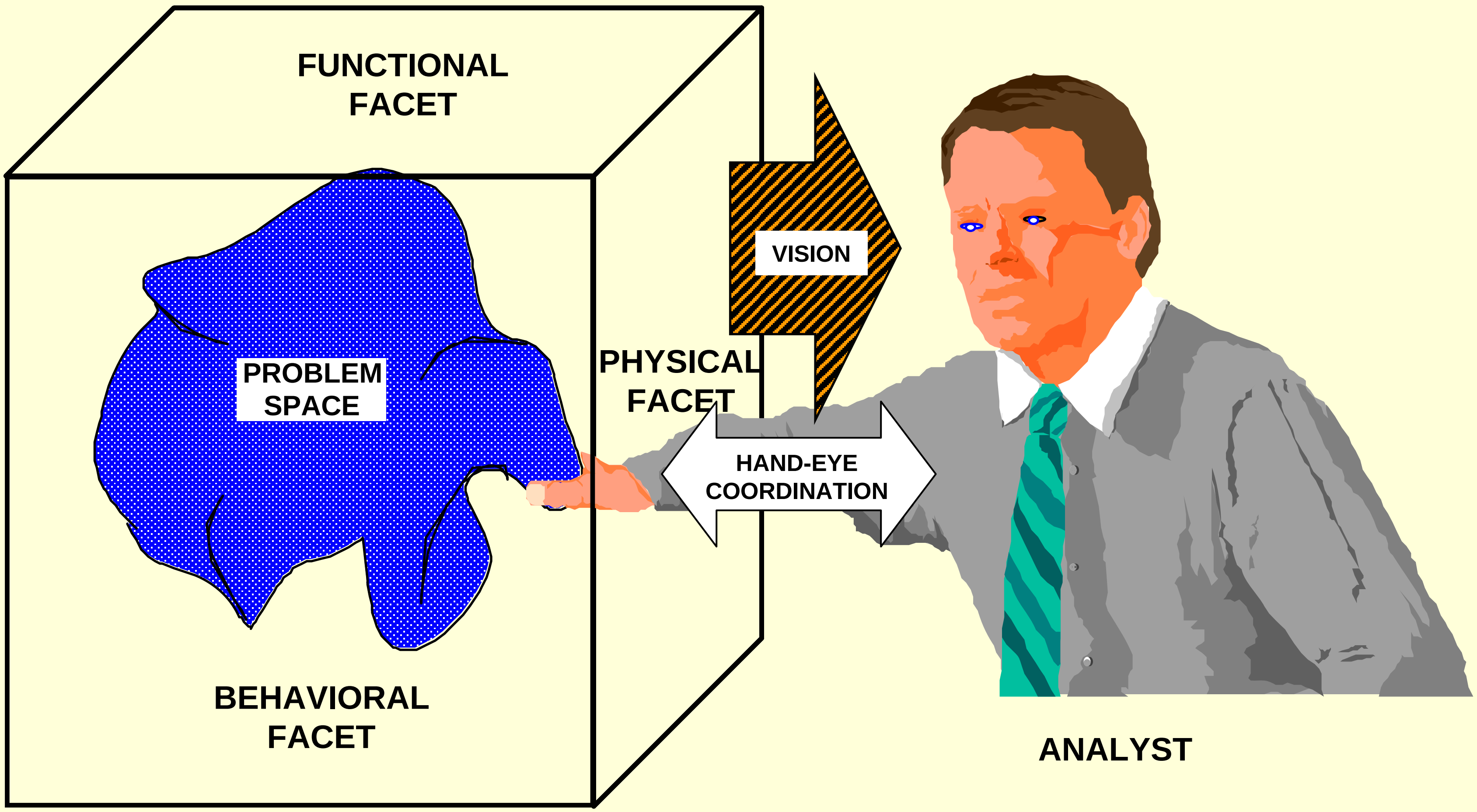
A specification is a document that contains all of the essential character-istics for a given item.

But, how do we identify all of the essential characteristics?

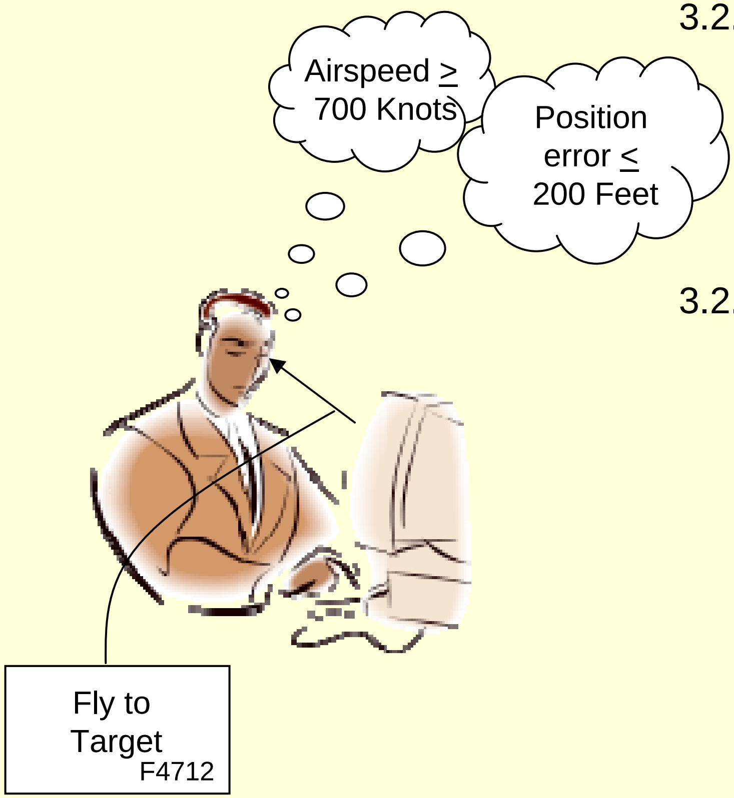
Writing Requirements is not Difficult

- The hard job is
  - Knowing what to write them about and
  - Determining numerical values that should be in them
- Thus we use models to gain insight into the essential characteristics
  - The models are composed of simple graphics
  - Model symbols (lines, blocks, bubbles, ...) relate to requirements that are derived from the model
  - The models encourage completeness and avoidance of unnecessary content
  - Models focus our human thought processes
- Good values requires good domain engineering skills

We Apply Models For Good Reasons



Deriving Performance Requirements



3.2.1.1 Aircraft shall be capable of flight at an airspeed ≥ 700 knots.

3.2.1.2 Position error at an end of leg shall be less than or equal to 200 feet in along track and cross track directions.

Bran Selic's Model Characteristics

- The use of **abstraction** to emphasize important aspects while removing irrelevant ones.
- Expressed in a form that is really **understandable** by observers.
- **Fully** and **accurately** represents the modeled system.
- **Predictive** such that it can be used to derive correct conclusions about the modeled system.
- **Inexpensive** meaning it is much cheaper to construct and study than simply building and observing the modeled system.

Architecture for Systems In Development

In DoDAF an Architecture Description Consists of:

- **A point in time**
- **A defined component**
- **Component parts**
- **What the parts do**
- **How the parts relate to each other**
- **The rules and constraints under which the parts function**

In this Discussion Architecture Is All of Those Things Plus -

- It can be described using a <u>comprehensive</u> model of the system covering product entities of which the system must consist and the relationships that must exist between them, its functionality, and its behavior.
- DoDAF uses 26 views to describe an architecture
- What the system must do, what it must consist of to accomplish those things, and how it must behave in doing so.
- The basis from which appropriate requirements are derived.

But Which Models?
System and Hardware Models

- **Traditional structured analysis**
  - **Functional analysis**
    - **Functional flow diagramming**
    - **Enhanced functional flow diagramming as used in CORE**
    - **Behavioral diagramming, derived from IPO, as used in RDD-100**
    - **IDEF 0, derived from SADT**
    - **Process flow analysis**
    - **Hierarchical functional analysis**
    - **FRAT (Mar and Morais)**
  - **State diagramming**
  - **Specialty engineering scoping and discipline-specific modeling**
  - **Three-tier environmental requirements construct**
  - **Product entity structure**
  - **Requirements analysis sheet**
- **SysML**

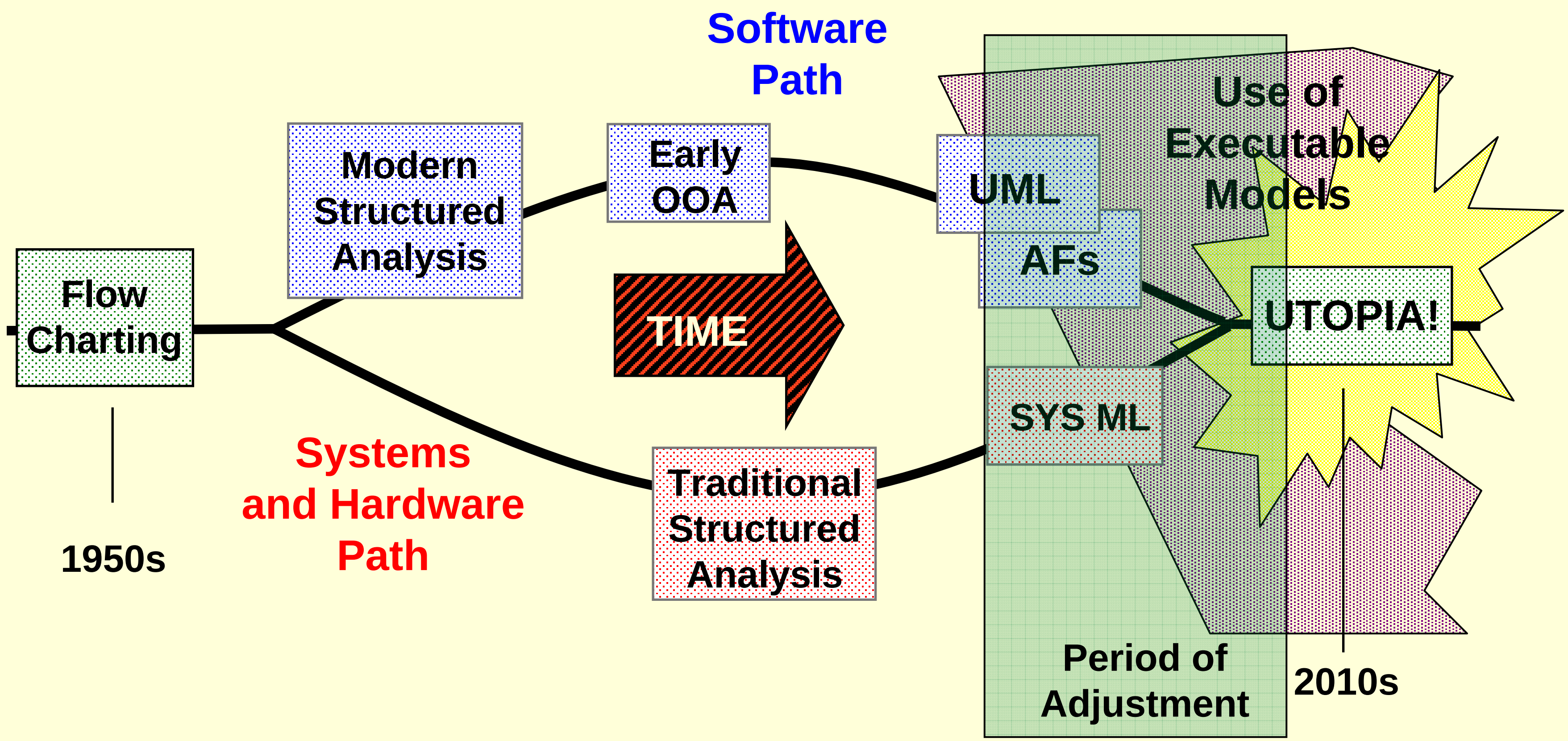
Computer Software Analysis Models

- **Process-oriented analysis**
  - **Flow charting**
  - **Modern Structured Analysis (Yourdon-Demarco)**
  - **PSARE (Hatley-Pirbhai)**
    - **Actually PSARE is a system model effective for Hardware or software**
- **Data-oriented analysis**
  - **Table normalizing**
  - **IDEF-1X**
- **Object-oriented analysis**
  - **Early models**
  - **UML**
- **DoD architecture framework**

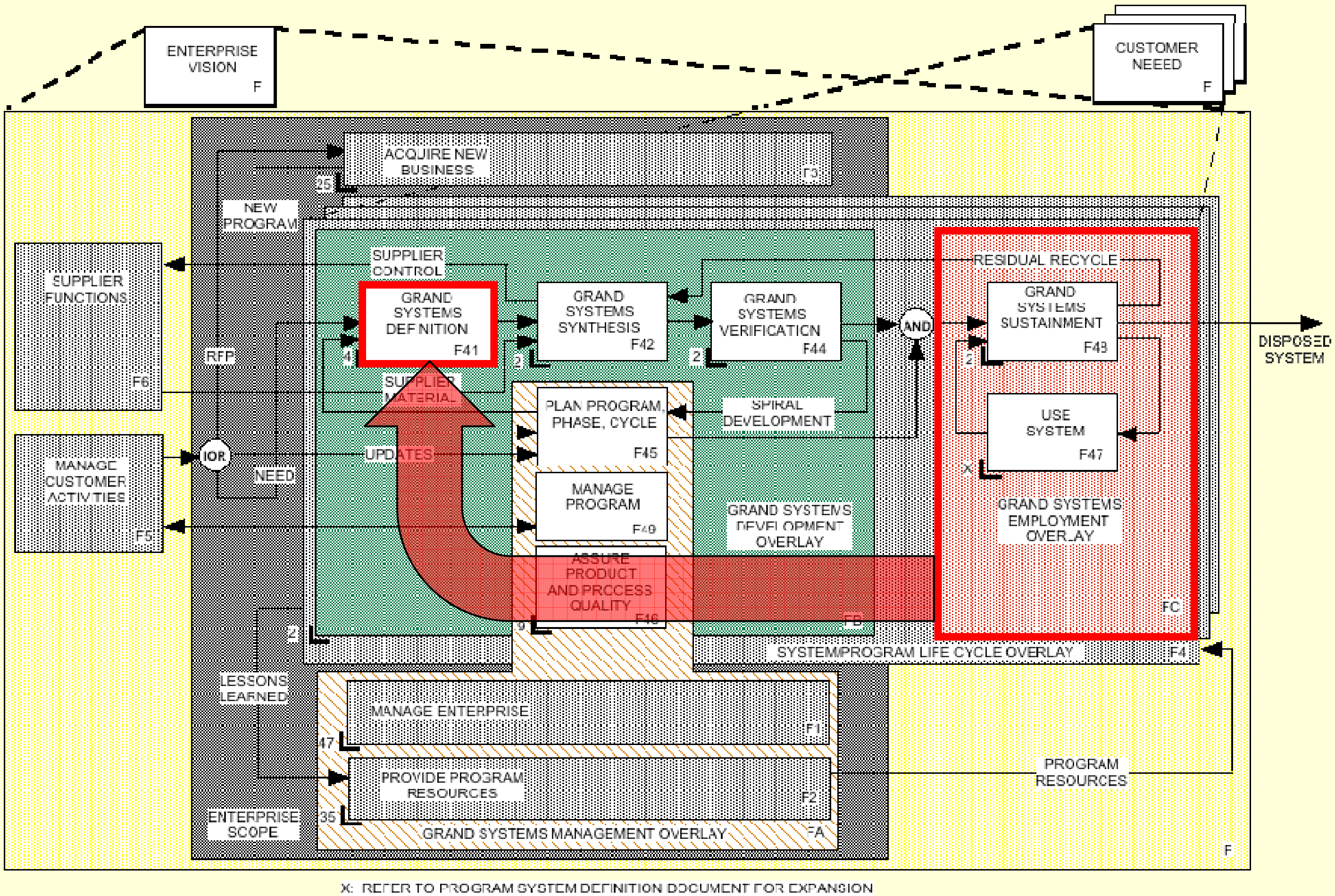
The Current Problem

- We have been tremendously creative in developing new models
- But very ineffective in integrating and optimizing across these available models
- So, that there is no single comprehensive model from which all essential characteristics can be derived
- This has led to use of unique hardware and software models resulting in some difficulty in hardware software integration

A Brief History of Requirements Modeling

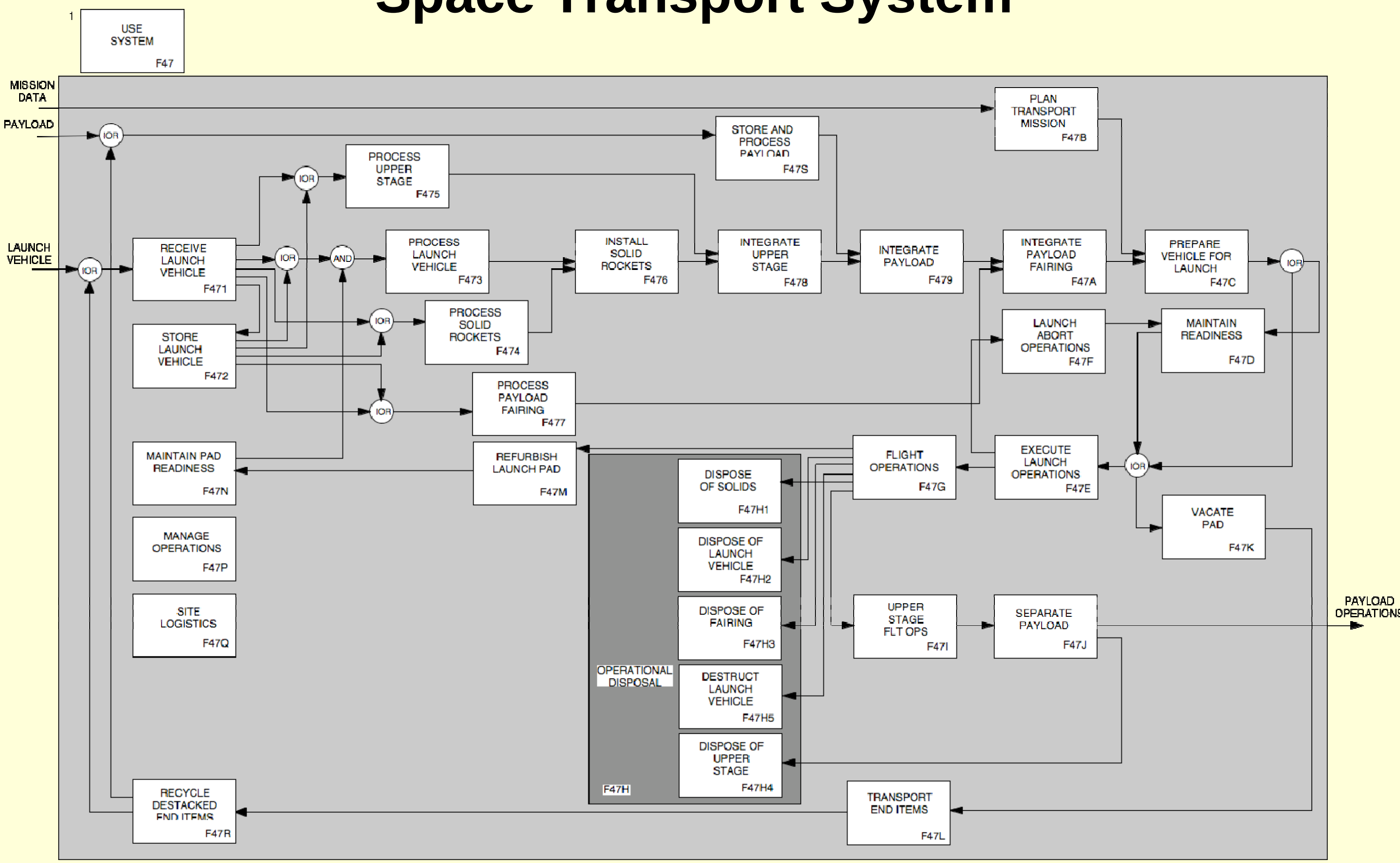


We Use the Models to Describe System Employment During System Definition

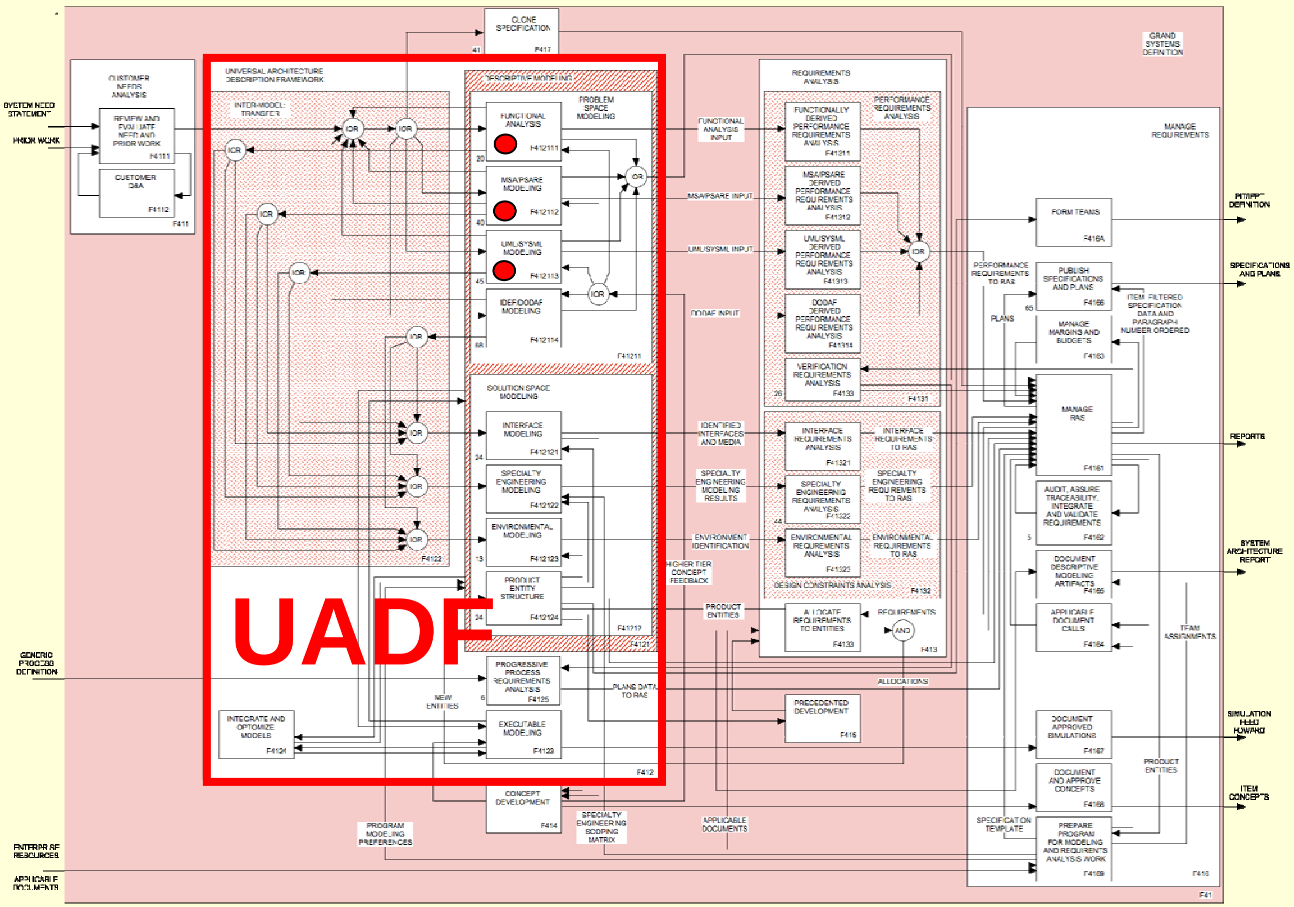


Use System Decomposition Example

Space Transport System



System Definition Should Include Problem and Solution Space Modeling

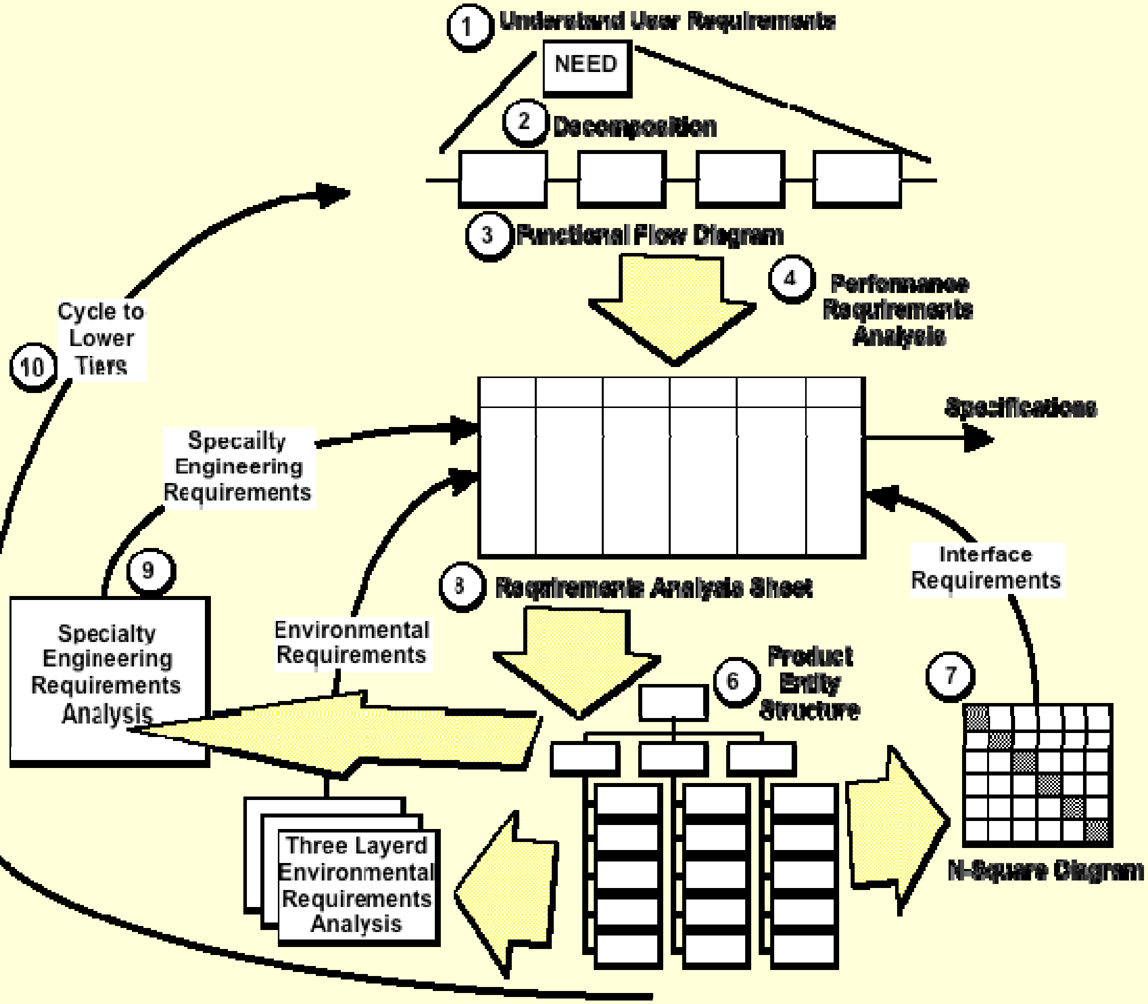


But We Have to Make Choices to Form Our Own UADF

- **Traditional Structured Analysis (TSA) Model**
  - **Flow diagramming linked to a RAS and Product Entity Diagram**
  - **Supplemented with n-square analysis for interface, specialty engineering scoping matrix for specialty engineering direction coordinated with the discipline models, and a three layered environmental model.**
  - **Could be applied to software (flow charts) as well as systems and hardware but probably not a popular choice**
- **PSARE augmented with TSA solution space models**
- **UML-SysML augmented with TSA solution space models**

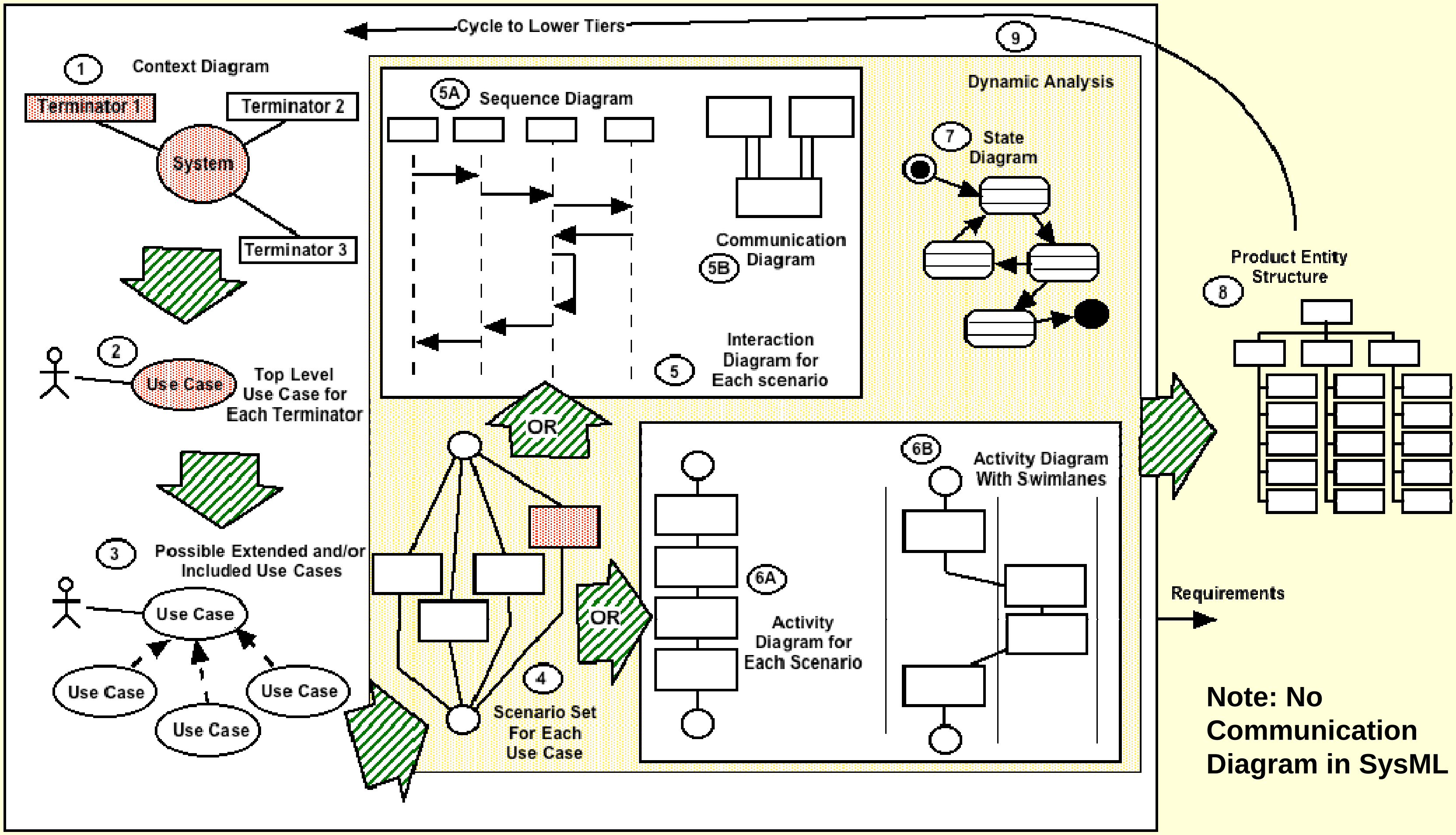
Traditional Structured Analysis

Overview



UML-SysML UADF

Overview

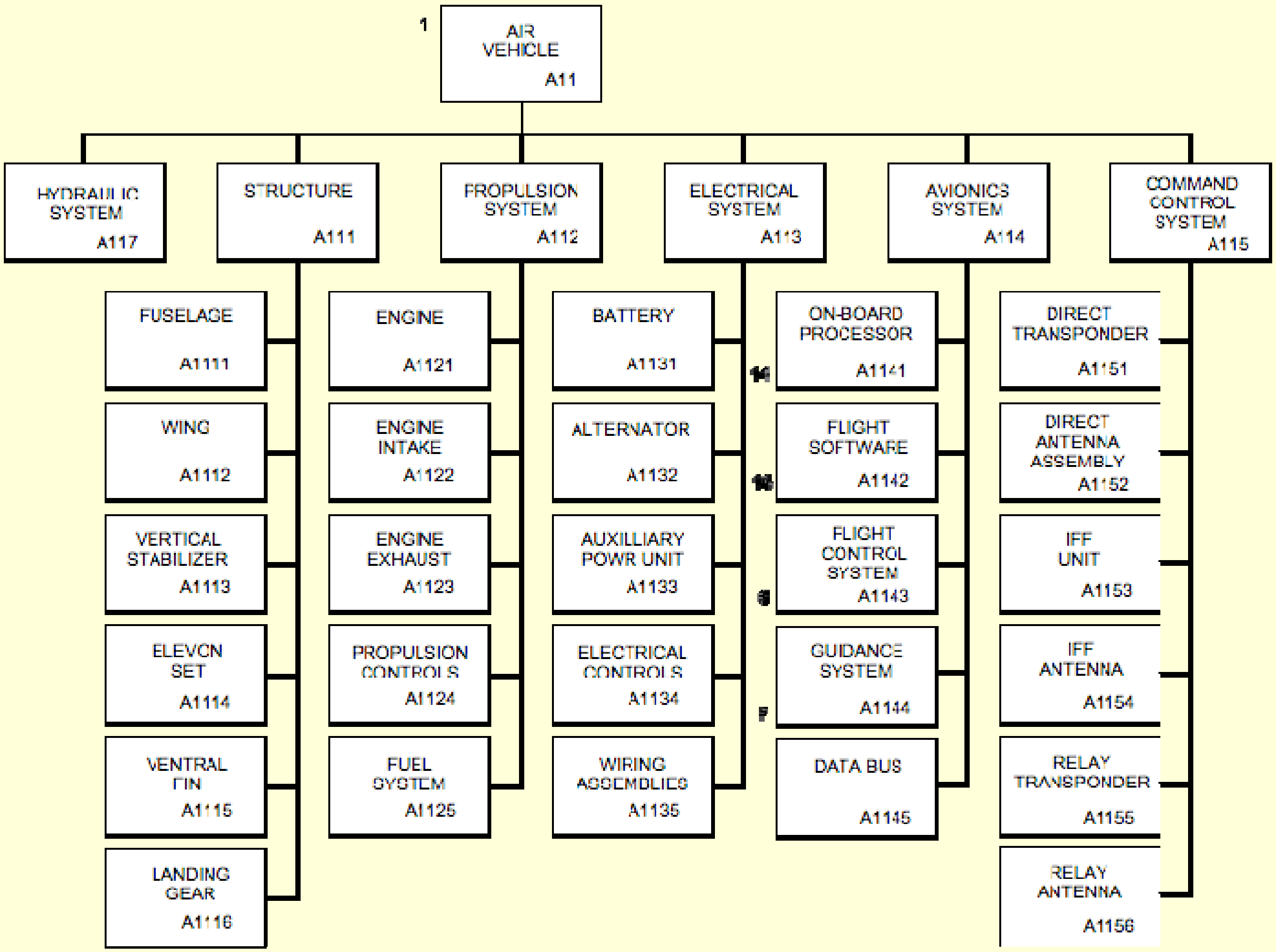


Note: No Communication Diagram in SysML

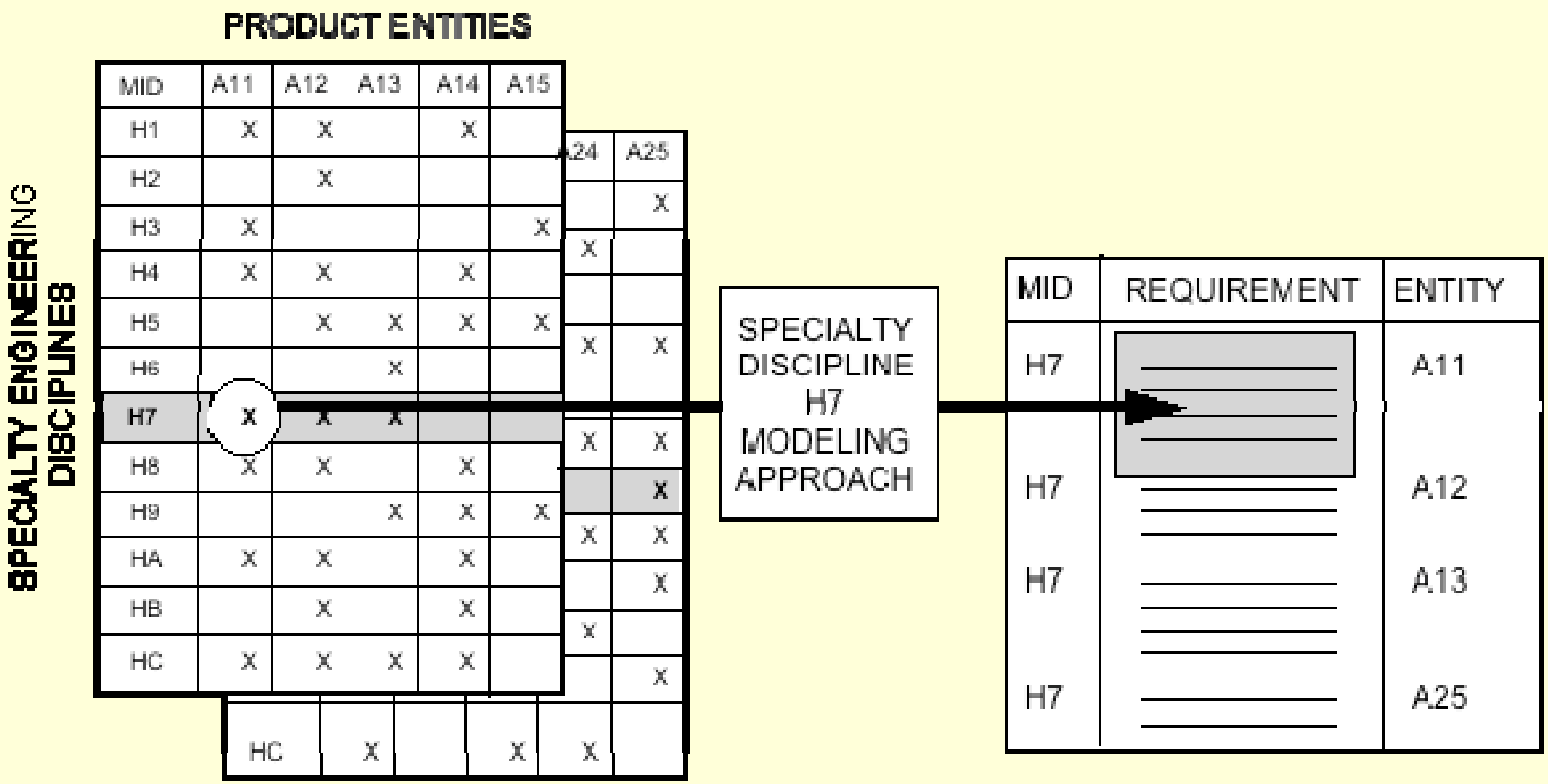
TSA Augmentation for PSARE or UML-SysML UADF RAS-Complete

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	XR87	Storage Volume < 10 ISO Vans	A1	Sensor Subsystem		
H	Specialty Engineering Disciplines			A	Product System		
H11	Reliability	EY34	Failure Rate < 10 x 10-6	A1	Sensor Subsystem	3.1.5	Reliability
H11	Reliability	RG31	Failure Rate < 3 x 10-6	A11	Cable	3.1.5	Reliability
H11	Reliability	F7H4	Failure Rate < 5 x 10-6	A12	Sensor Element	3.1.5	Reliability
H11	Reliability	G8R4	Failure Rate < 2 x 10-6	A13	Pressure Vessel	3.1.5	Reliability
H12	Maintainability	8GHU	Mean Time to Repair < 0.2 Hours	A1	Sensor Subsystem	3.1.6	Maintainability
H12	Maintainability	U8R4	Mean Time to Repair < 0.4 Hours	A11	Cable	3.1.6	Maintainability
H12	Maintainability	J837	Mean Time to Repair < 0.2 Hours	A12	Sensor Element	3.1.6	Maintainability
H12	Maintainability	9D7H	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 Interface			A1			
I161	Aggregate Signal Feed Source Impedance	E37H	Aggregate Signal Feed Source Impedance= 32 ohms + 2 ohms	A1	Sensor Subsystem		
I161	Aggregate Signal Feed Load Impedance	E371	Aggregate Signal Feed Load Impedance= 32 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	8D74	-40 degrees F< Temperature < +140 degrees F	A	Product System		
QX	Non-Cooperative Environmental Stresses			A	Product System		

TSA Augmentation for PSARE or UML-SysML UADF Product Entity Structure



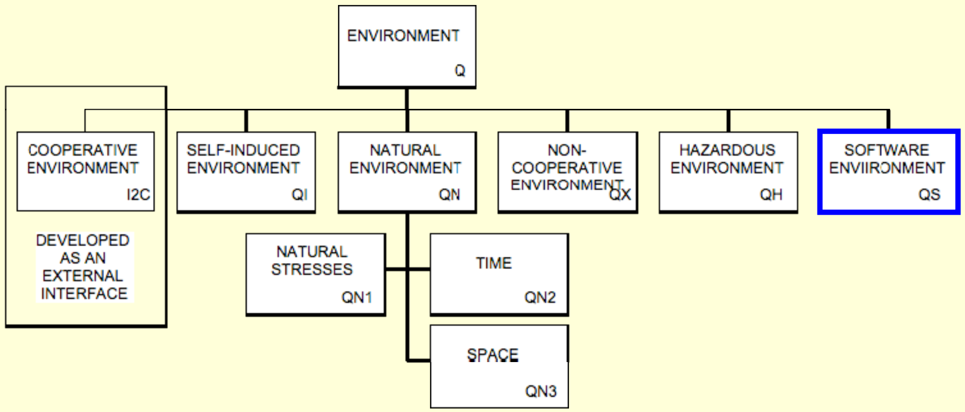
TSA Augmentation for PSARE or UML-SysML UADF
Specialty Engineering Scoping Matrix

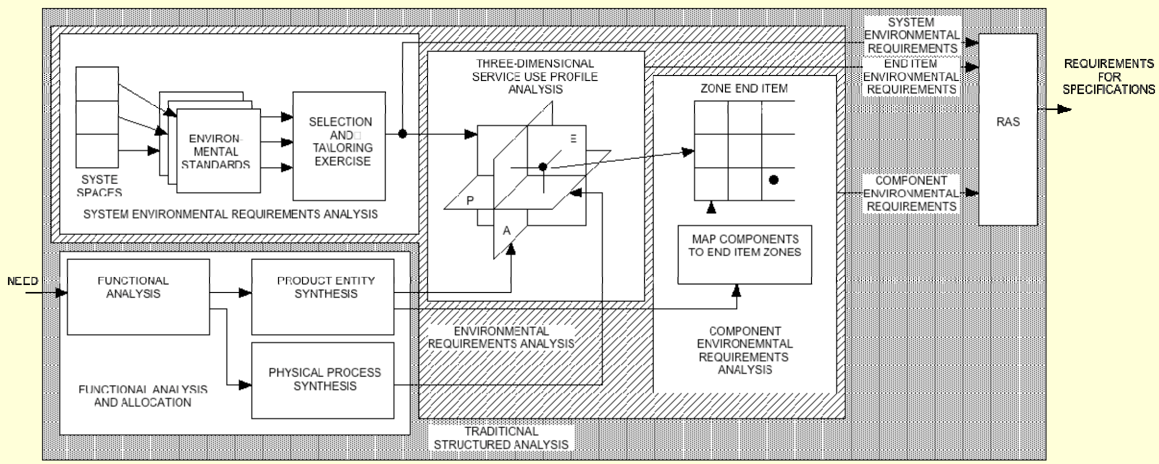


a. Specialty Engineering Scoping Matrix

b. Requirements Analysis Sheet (RAS)

Environment Classes and Three-Tiered Environmental Requirements Construct

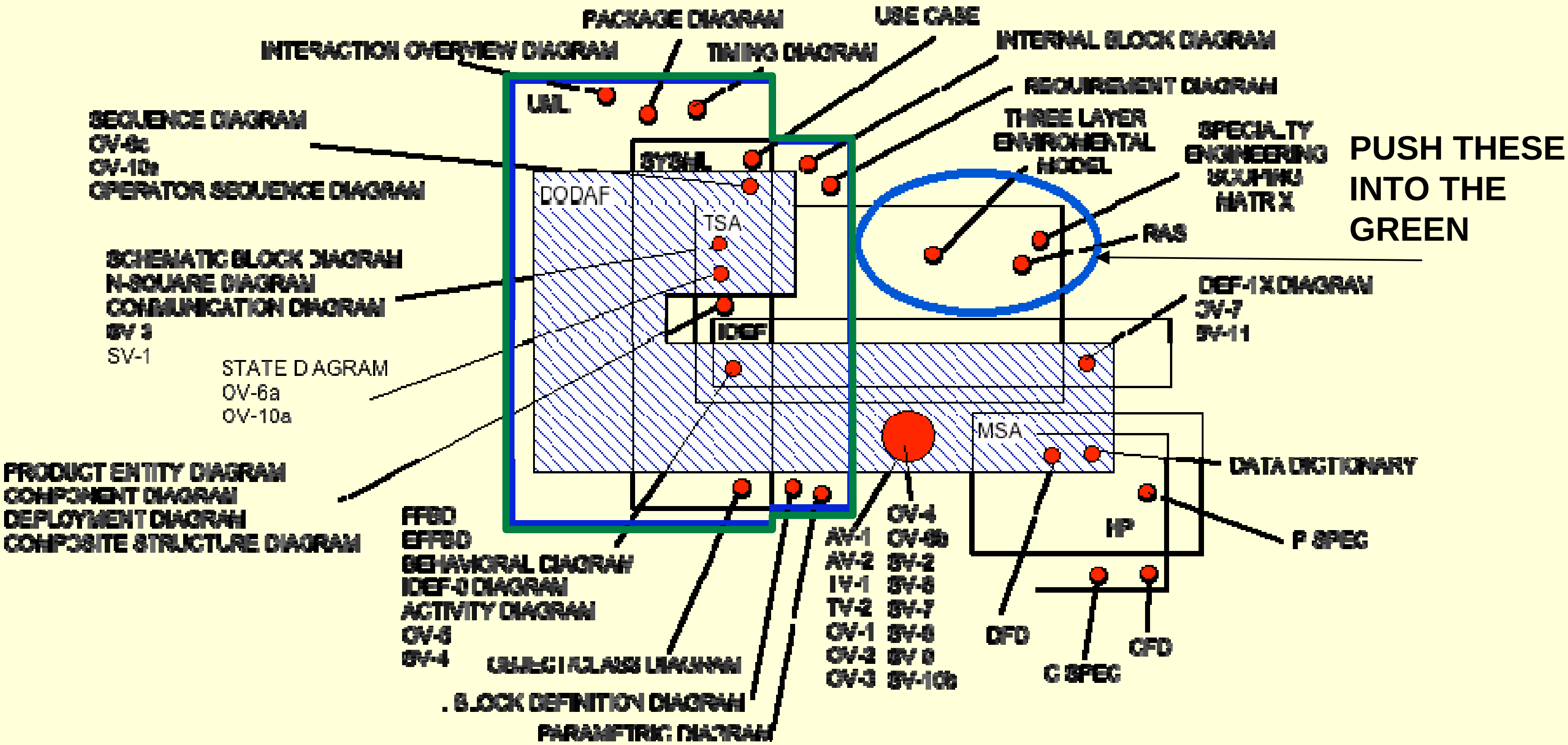




Three-Tiered Environmental Model

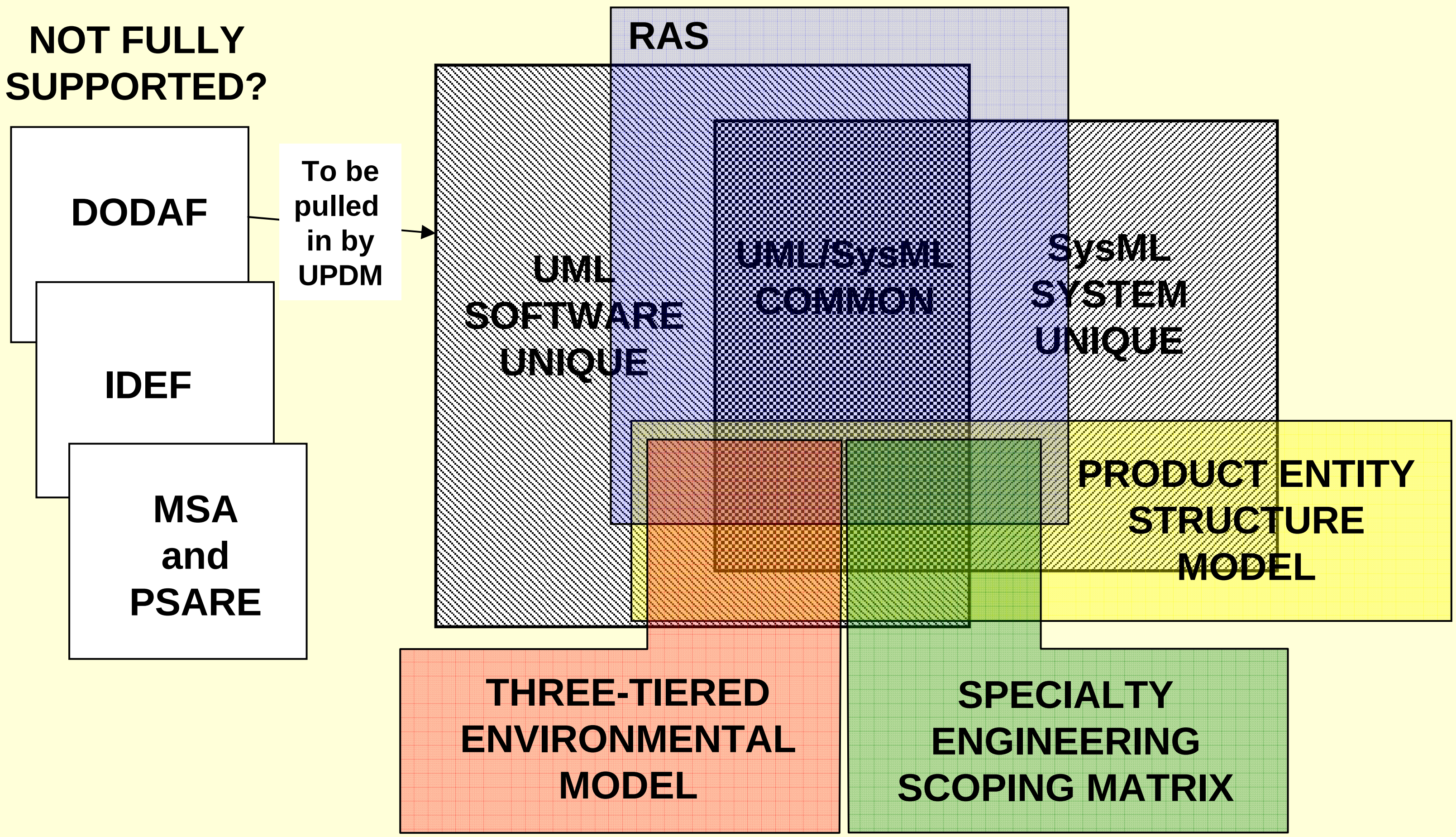
- **System level**
  - **List all spaces within which the system must function, map them to environmental standards, select parameters that apply, tailor the range of selected parameters**
- **End item level**
  - **Define three dimensional service use profile**
  - **Map system environmental requirements to process steps**
  - **Map product entities to process steps**
  - **Extract environmental requirements linked to entities**
- **Component level**
  - **Zone end item and map components to zones**

Venn Diagram View of the Universal Model Set In 2008

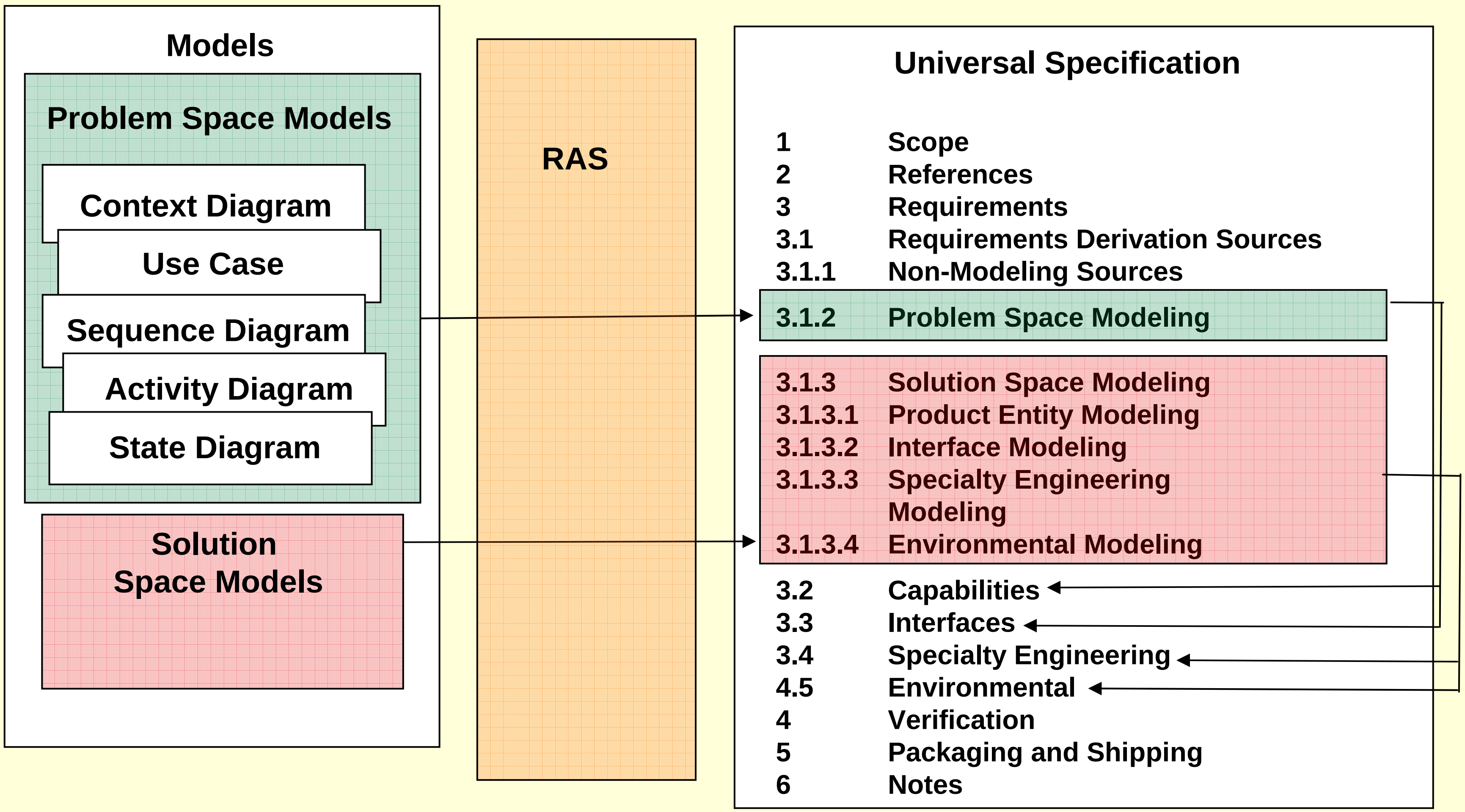


UML-SysML Model Combination

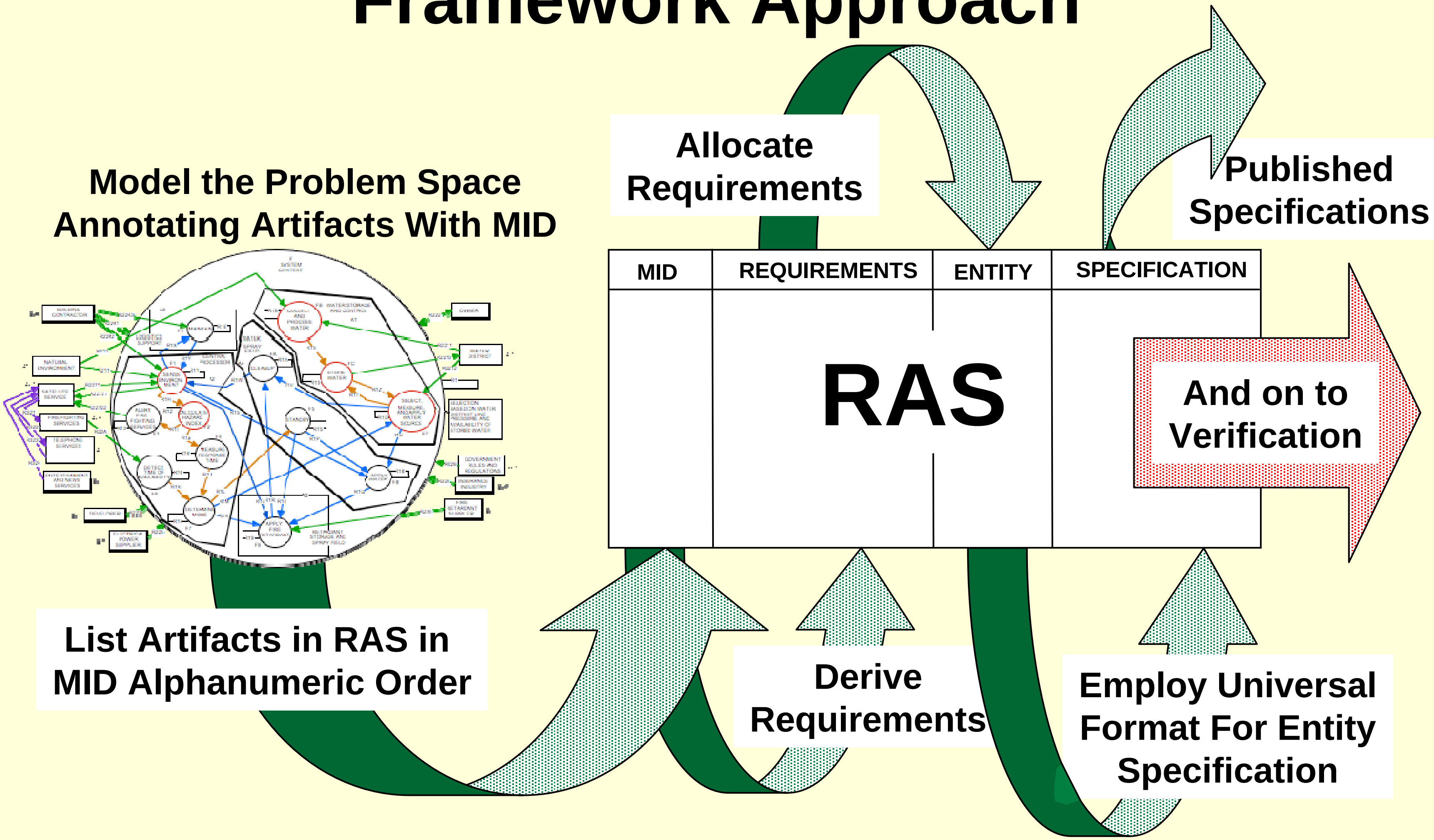
A Universal Model Using SysML-UML



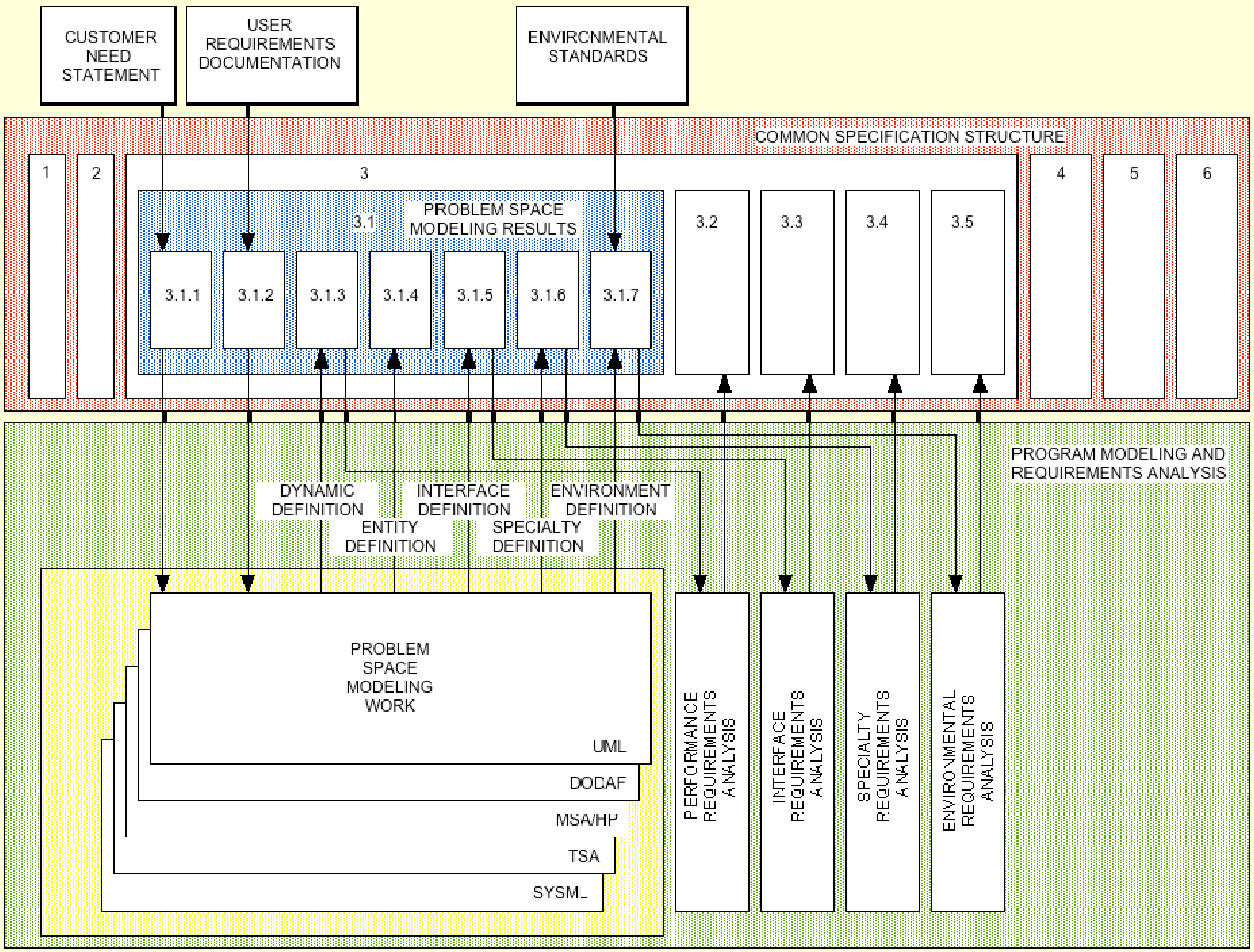
Model Results Flow Into Specifications Content Through the RAS



Universal Architecture Description Framework Approach



Building Universal Specifications



Benefits of Universal Modeling

- **Alignment between system, hardware, and software modeling orientations making it easier for management and system engineering people to understand and control the overall process.**
- **Improved hardware - software integration capability.**
- **Improved requirements traceability across the hardware - software gap.**
- **Everyone will be able to understand the system development process no matter their specialty supporting the notion of maximizing the communication capabilities of team members while minimizing the need to communicate improving the signal to noise ratio of program communications.**

Action Items For You

- Continue your studies of requirements work
- Come to an understanding about UML and SysML
- Within your companies and programs develop modeling skills and work toward transforming your combined set of models into a universal set
- Work toward correlating the SW and HW development work patterns so as to encourage more effective integration