NDIA
SYSTEM
ENGINEERING
CONFERENCE

JOG SYSTEM ENGINEERING
GRAND SYSTEMS DEVELOPMENT TRAINING PROGRAM
TUTORIAL

UNIVERSAL ARCHITECTURE DESCRIPTION FRAMEWORK

INTRODUCTION

Presented By
Jeffrey O. Grady
President JOG System Engineering
6015 Charae Street
San Diego, CA 92122
(858) 458-0121
jeff@jogse.com jgrady@ucsd.edu

Who Is Jeff Grady?

CURRENT POSITION

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

PRIOR EXPERIENCE

1954 -1964	U.S. Marines
1964 - 1965	General Precision, Librascope Div 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 and Group Engineer, Cruise Missile, Advanced Cruise Missile
1984 - 1993	General Dynamics Space Systems Division Functional Engineering Chief & Manager of Systems Development

FORMAL EDUCATION

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

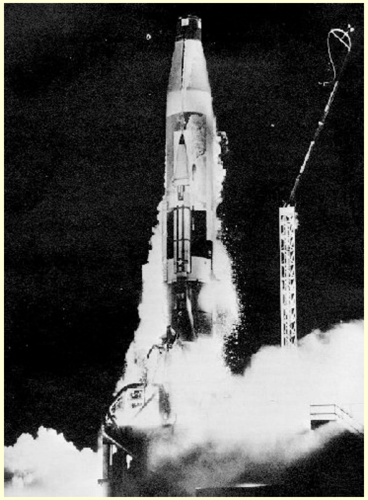
INCOSE First Elected Secretary, Fellow, Founder, Expert System Engineering Professional

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

Systems Jeff Grady Worked On

USN/Librascope ASROC/SUBROC Computer Systems

USAF/GD Convair AQM 129 Advanced Cruise Missile



USAF/GD Atlas Missile

USAF/Ryan AQM-81 Firebolt 1983

Ryan Aeronautical War Birds

USAF/Ryan Models 147G, NX, H, and J at Bien Hoa, SVN in 1968

USAF/Ryan AQM-34L Tom Cat
58 Combat Missions

U.S. Navy/Ryan
Model 147SK

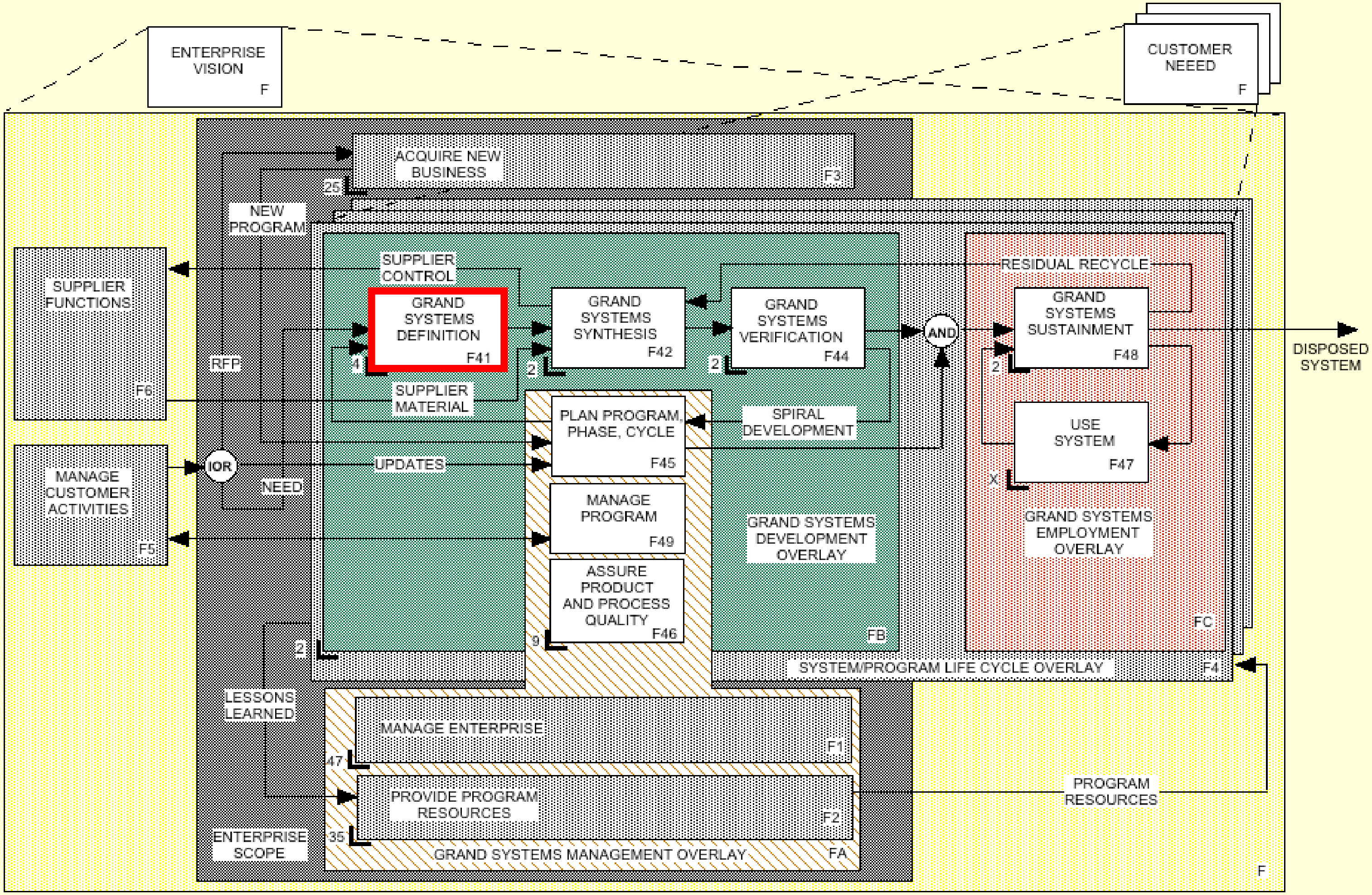
USAF/Ryan
BGM-34C

Who is Attending?

- Small class
  - Name
  - Place of employment
  - Modeling and requirements experience
- Large class
  - At first break engage in conversation with someone you did not know when arriving for class
  - Discuss modeling and requirements work

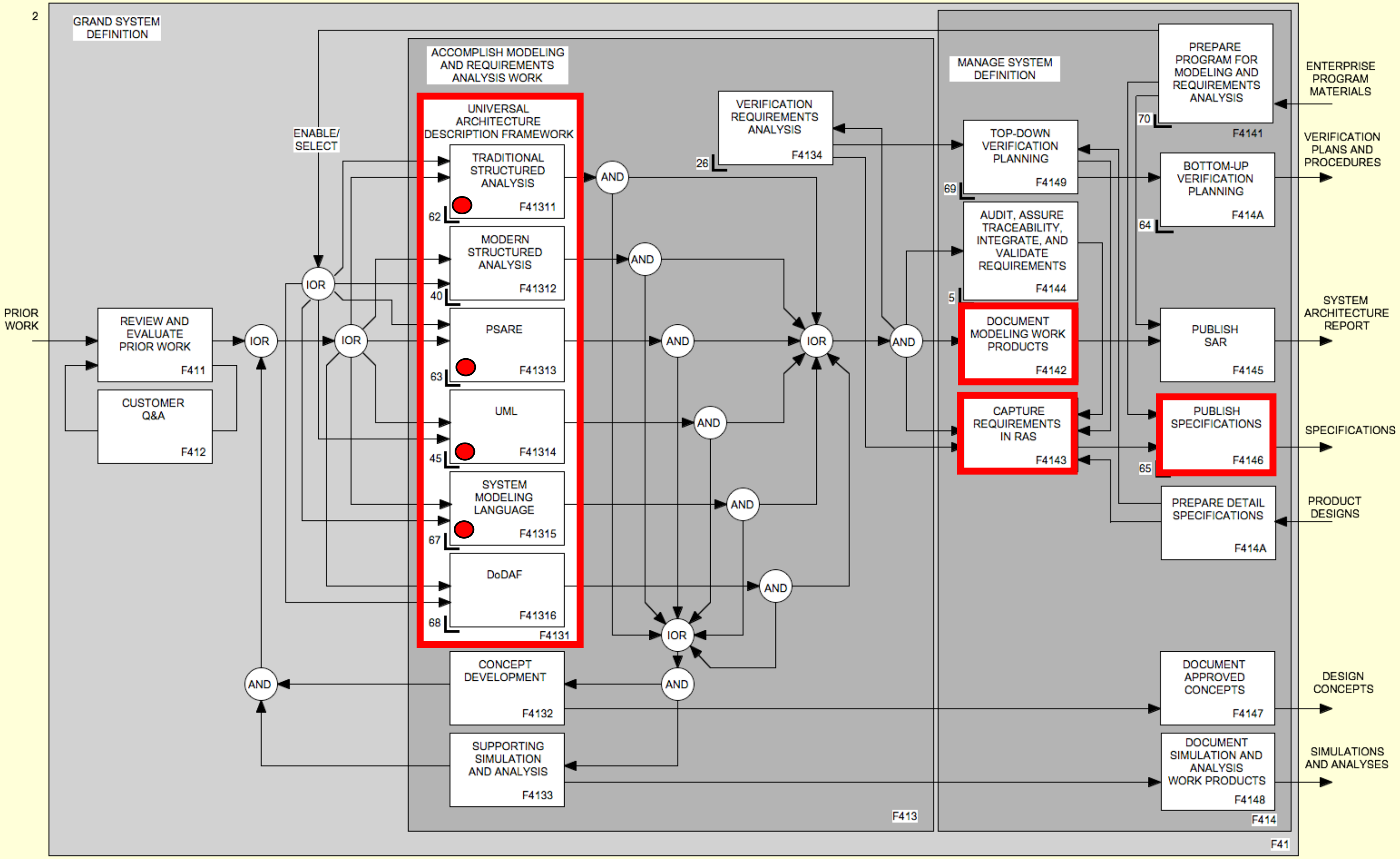
Tutorial Outline

Enterprise Common Procss



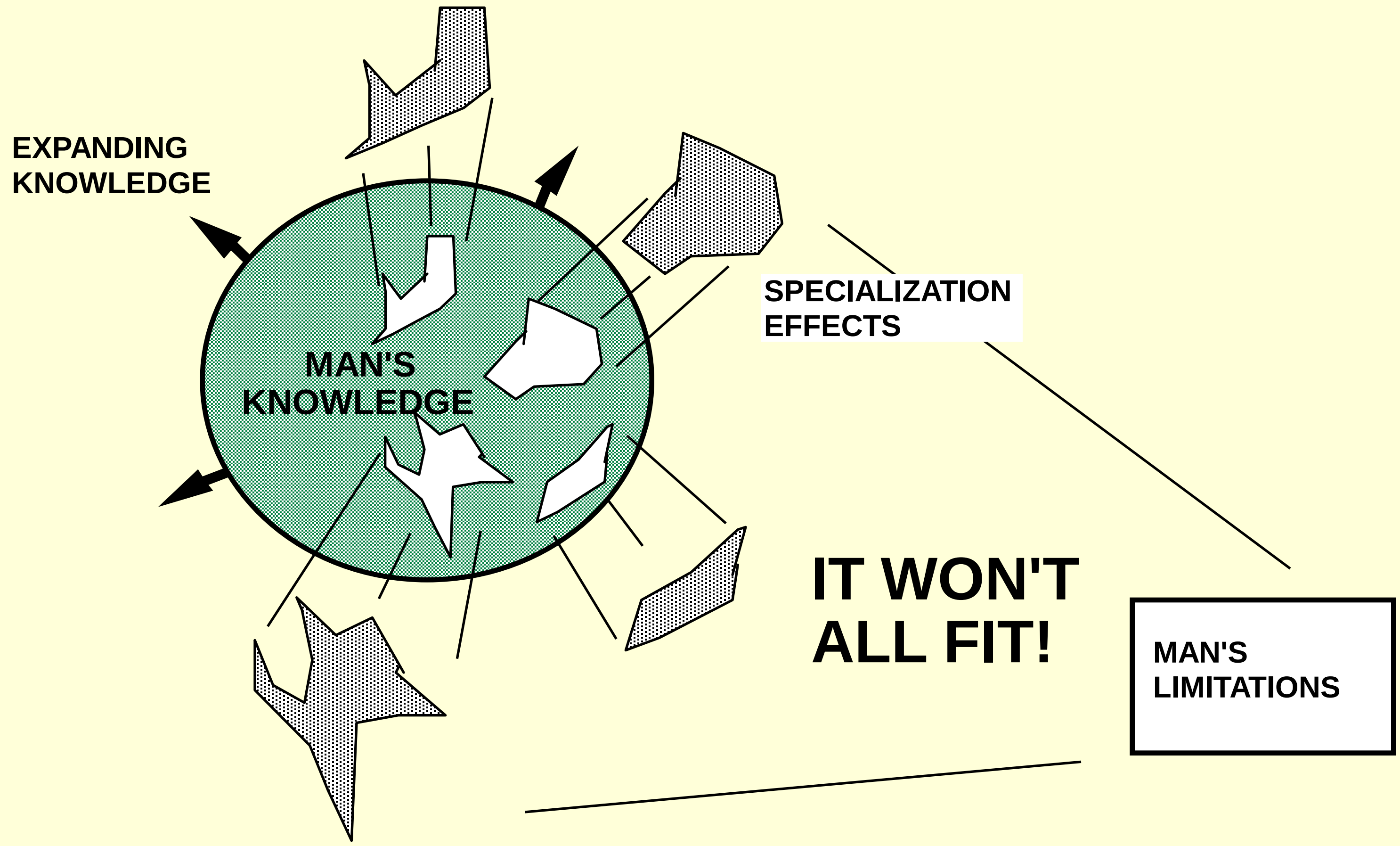
X: REFER TO PROGRAM SYSTEM DEFINITION DOCUMENT FOR EXPANSION

Common Process Areas of Interest



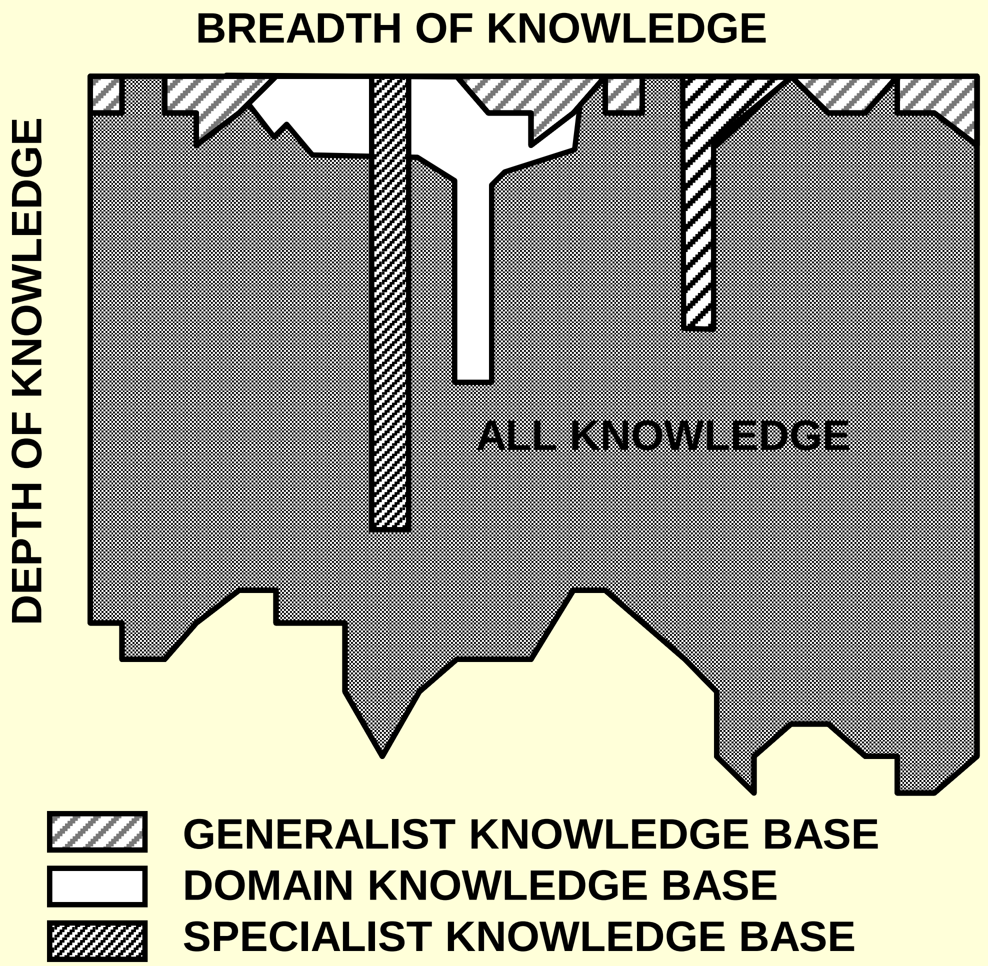
The Foundation of System Engineering

Knowledge Grows & We Have Our Limitations



We Are All Specialists

Systems Are Developed by People Sharing Knowledge



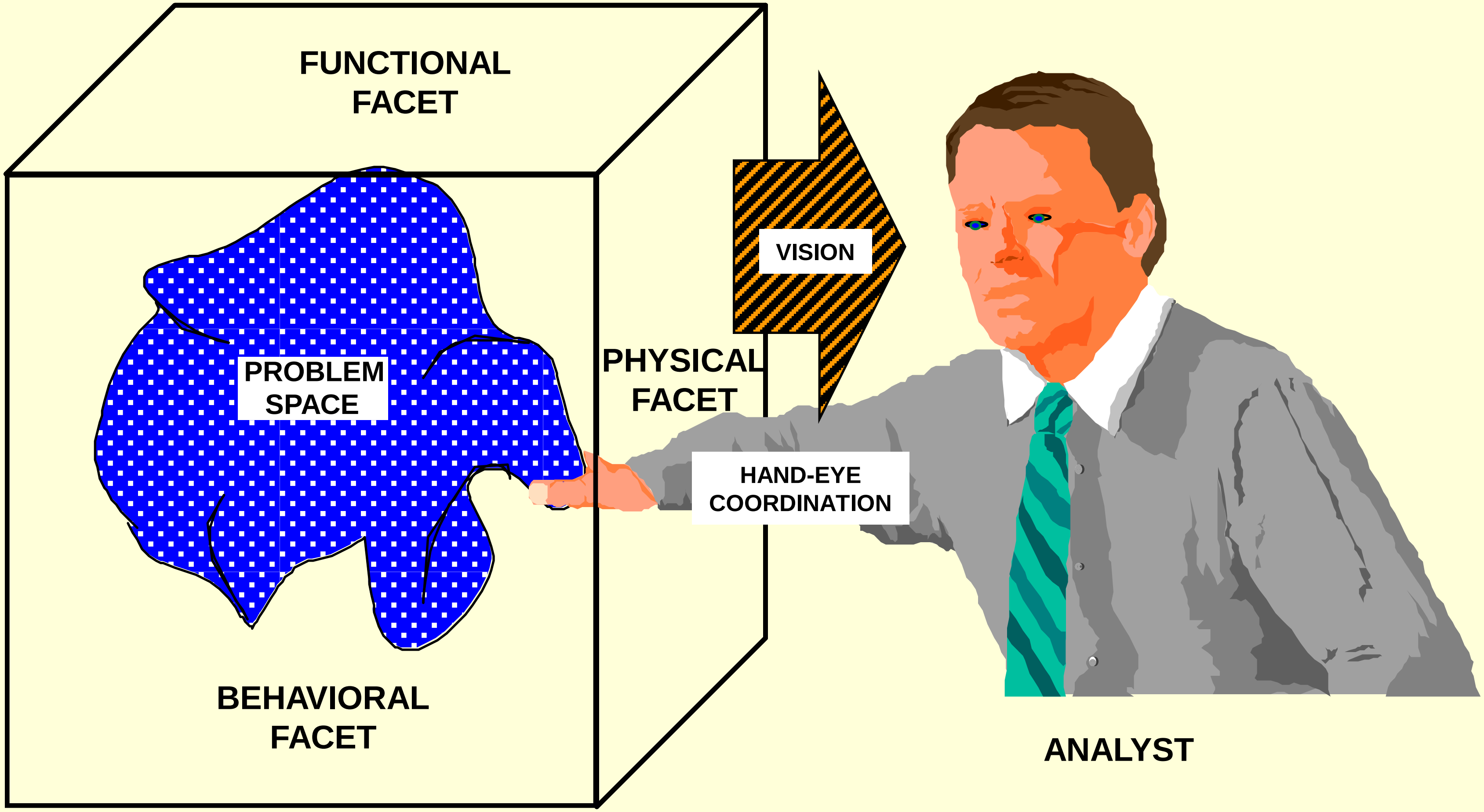
Models Support Information Sharing During Early Development

- When developing an unprecedented system it is helpful to model it as a way of learning more about it
- We have no reality to observe in the beginning so we must model

Bran Selic's Model Characteristics

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

We Apply Models For Good Reasons

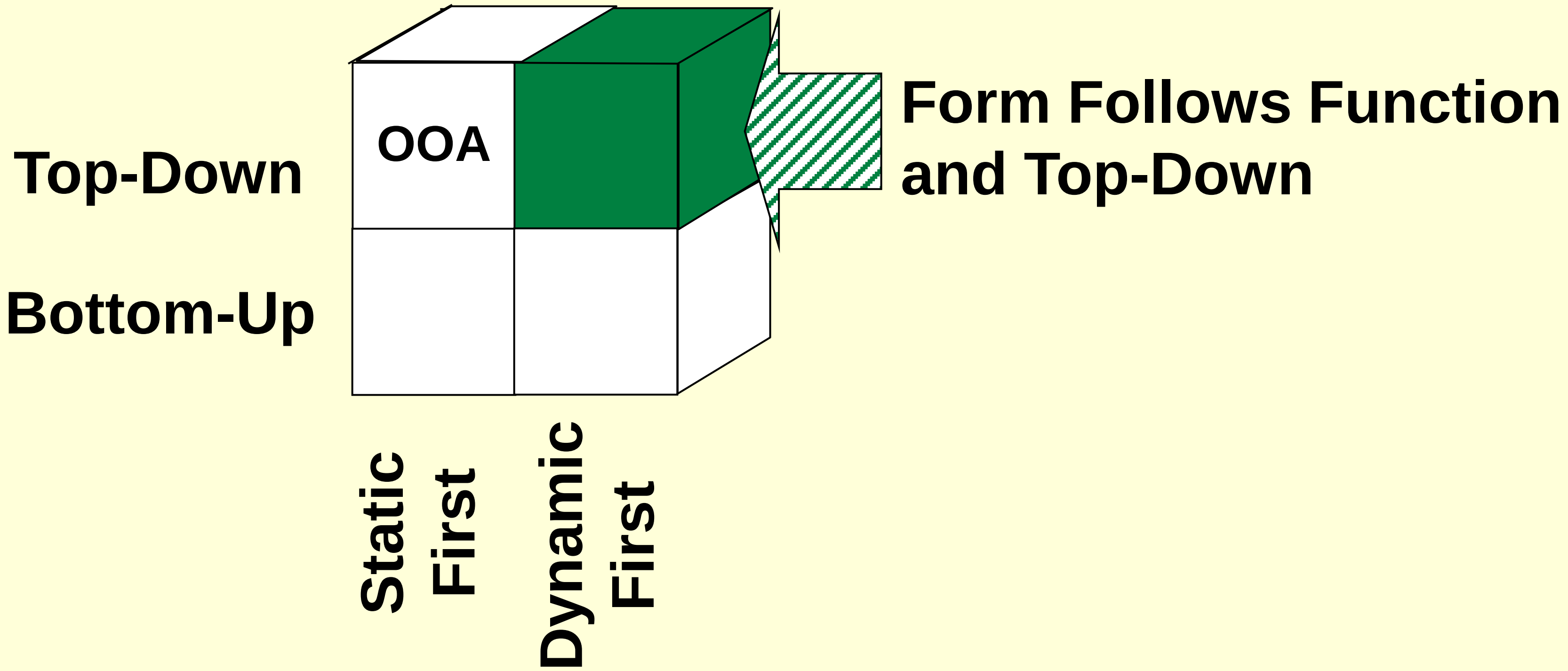


Visual Complexity Continuum

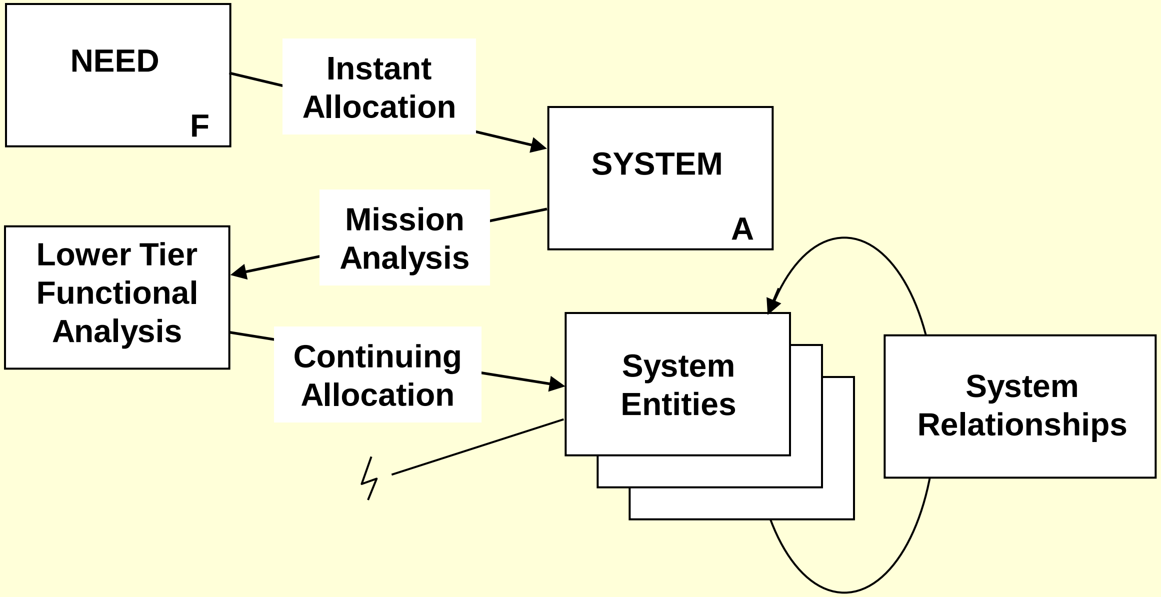
Some Models Are Richer Than Others

- Some models are visually simple like functional flow diagramming
  - Ideas flow readily from model into the human mind through vision because of the simple graphics
  - Not a very rich story comes through
- Other models are more visually complex like IDEF-0, EFFBD, DoDAF
  - The picture does not transfer so easily into the mind visually
  - DoDAF includes 26 different artfacts
  - But it conveys a very rich story when it gets there

Modeling Sequencing



Which Comes First?



Clearly the functionality

Architecture First Counter-Argument

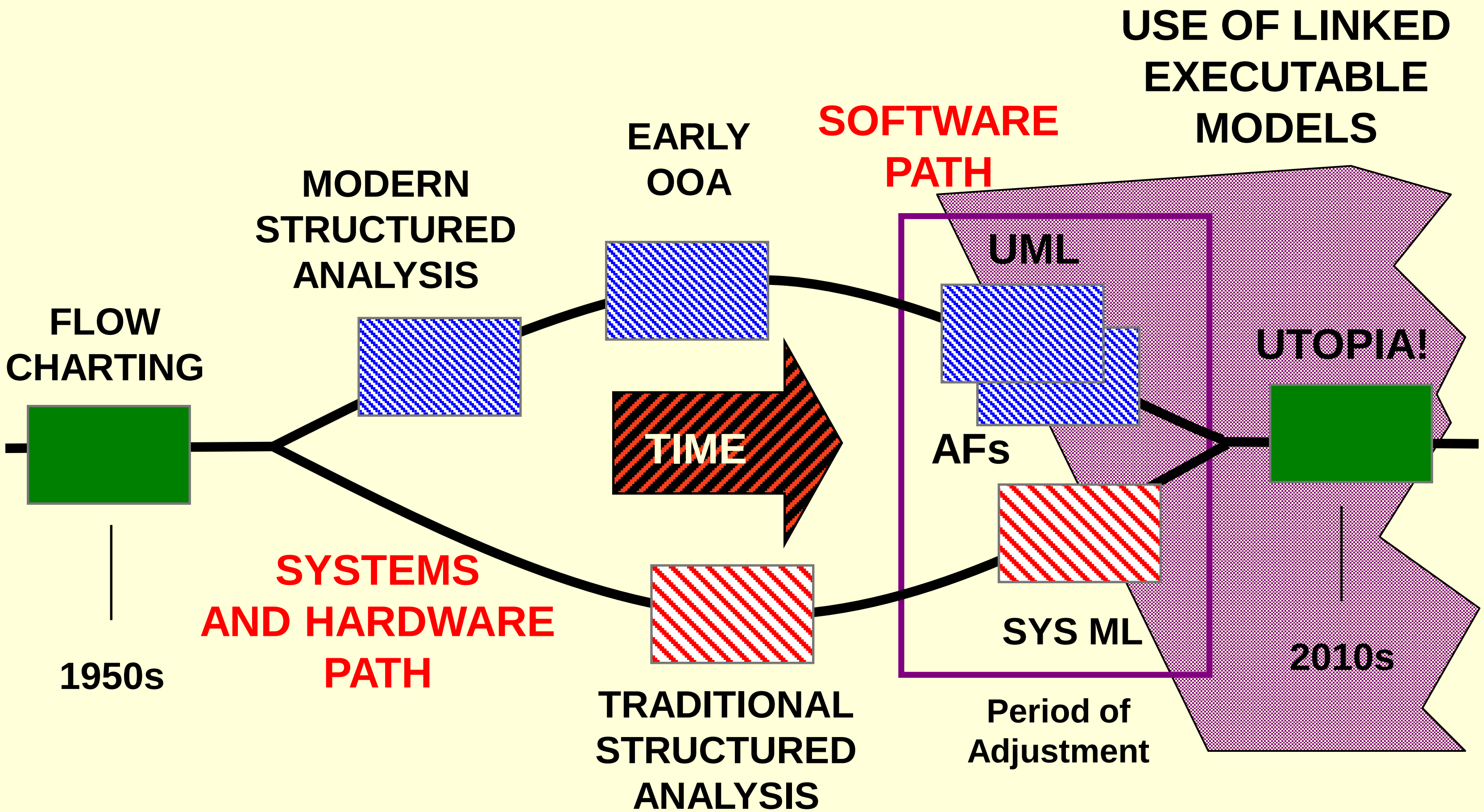
- Early OOA authors all supported object entry into the problem space with DFD and state machine examination of objects
- Exactly opposite to Sullivan’s idea of form follows function
- Murray Cantor’s “Thoughts on Functional Decomposition” in The Rational Edge offers the best printed argument for this approach
- Principal argument seems to be that multiple lower tier alternative solutions will appear in the product but this is a failure of lower tier system integration and optimization and the standard PMP concept can be employed with software as well as hardware

Model Orientation Relative to Dynamic and Static Components

	STATIC COMPONENT	DYNAMIC COMPONENTS	
MULTI-FACETED MODELS	PRODUCT ENTITY FACET	FUNCTIONAL FACET	BEHAVIOR FACET
TRADITIONAL STRUCTURED ANALYSIS	ARCHITECTURE BLOCK DIAGRAM	FUNCTIONAL FLOW DIAGRAM ●	SCHEMATIC BLOCK DIAGRAM
MODERN STRUCTURED ANALYSIS AND HP	HIERARCHICAL DIAGRAM	DATA FLOW DIAGRAM ●	P SPEC, STATE DIAGRAM
EARLY OBJECT-ORIENTED ANALYSIS	CLASS AND OBJECT DIAGRAM ●	DATA FLOW DIAGRAM	STATE DIAGRAM
UML VARIATION OF OOA	CLASS AND OBJECT DIAGRAM, COMPONENT DIAGRAM, AND DEPLOY-MENT DIAGRAM	USE CASES AND ACTIVITY DIAGRAM ●	SEQUENCE DIAGRAM, STATECHART, AND COMMUNICATION DIAGRAM
SysML	BLOCK DIAGRAMS	USE CASES AMD ACTVITY DIAGRAM ●	SEQUENCE DIAGRAM AND STATECHART

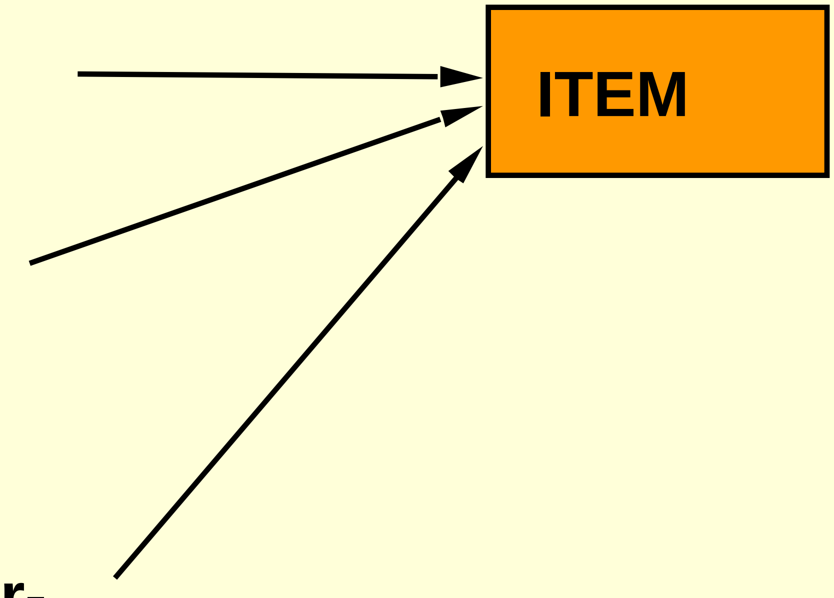
● ANALYTICAL ENTRY FACET

The History of Requirements Modeling



The First Objective of Modeling
- Requirements Identification

- **Something wanted or necessary.**
- **Something essential to the existence or occurrence of something else.**
- **A necessary character-istic or attribute of some thing (or item).**



Requirement Types

- **System composition**
  - What does the system consist on in terms of entities?
  - What relationships (interfaces) must exist between them?
- **Entity and relationship essential characteristics**
  - Performance (Functional)
    - What does it have to do and how well does it have to do it?
  - Design constraints (Non-functional)
    - Boundary conditions that the design team must remain within while satisfying performance requirements of three kinds
      - Interface
      - Specialty Engineering/Quality
      - Environmental

Writing Requirements is not Difficult

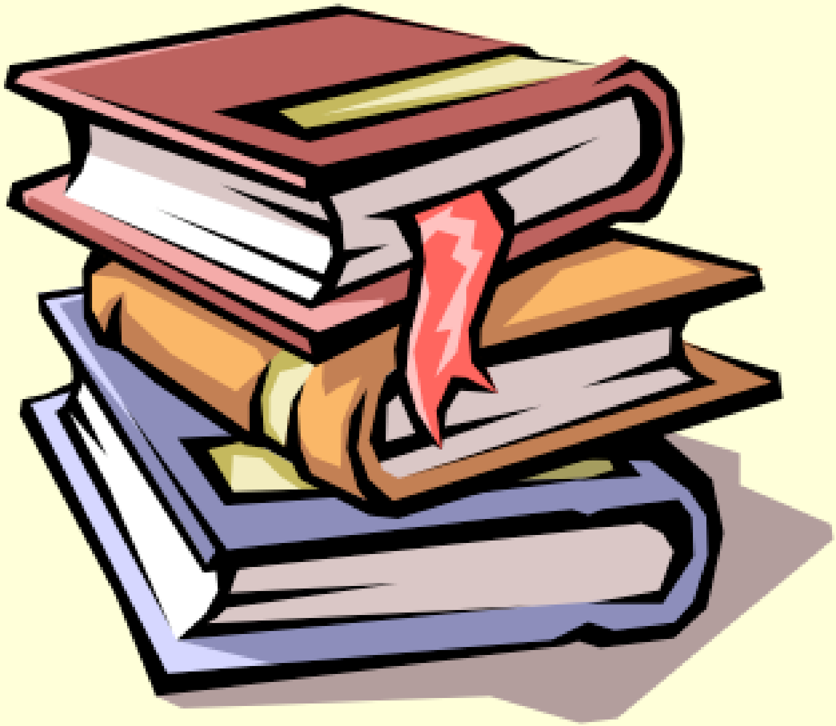
- The hard job is
  - Knowing what to write them about and
  - Determining numerical values that should be in them
- Thus we use models to gain insight into the essential characteristics
  - The models are composed of simple graphics
  - Model symbols (lines, block, bubbles, ...) relate to requirements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

Requirement Primitive Form

- The subject identifies the attribute that must be controlled
- A form of the verb shall shows the degree of determination that the design must possess a capability.
- The remainder of the sentence provides a value and the relationship between the value and the attribute.
- In primitive form:

$$\text{Speed} \geq 695 \text{ knots}$$

What is a Specification?

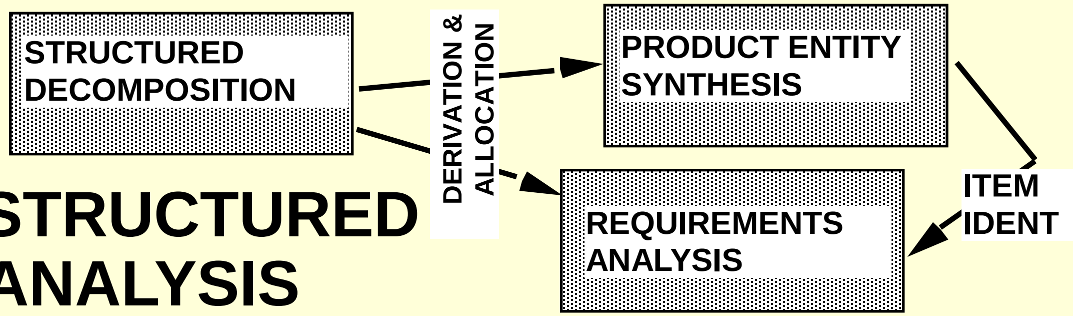


A specification is a document that contains all of the essential characteristics for a given item.

Must it be a paper document?

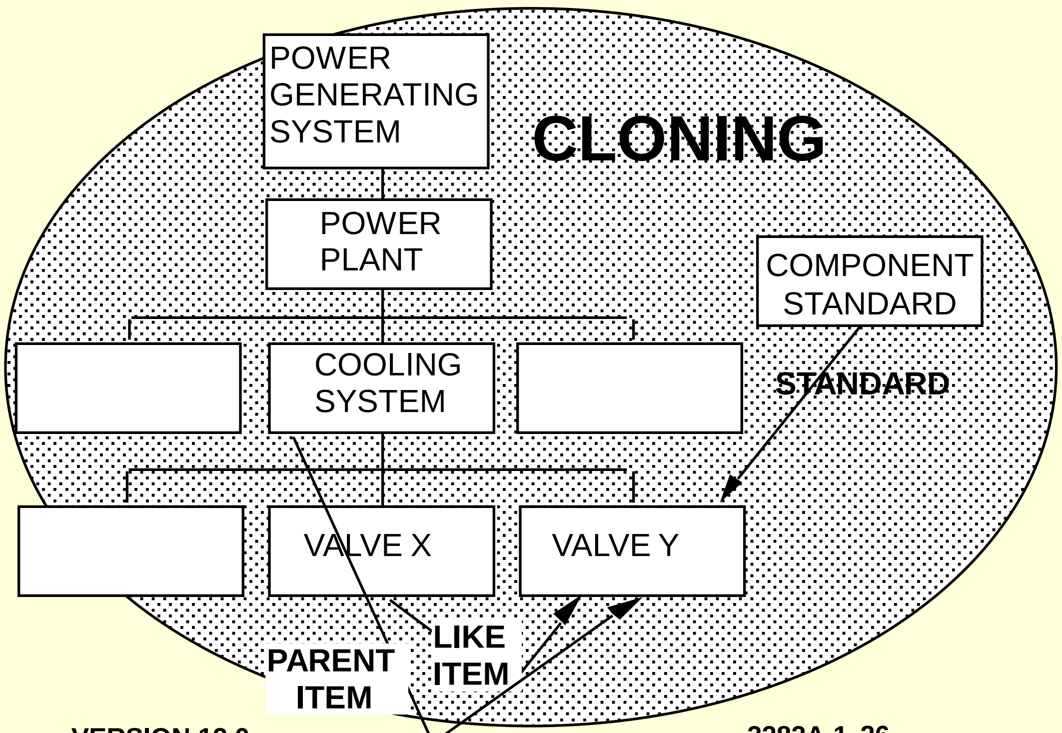
In Writing a Specification, What Is the Target?

Requirements Derivation Strategies



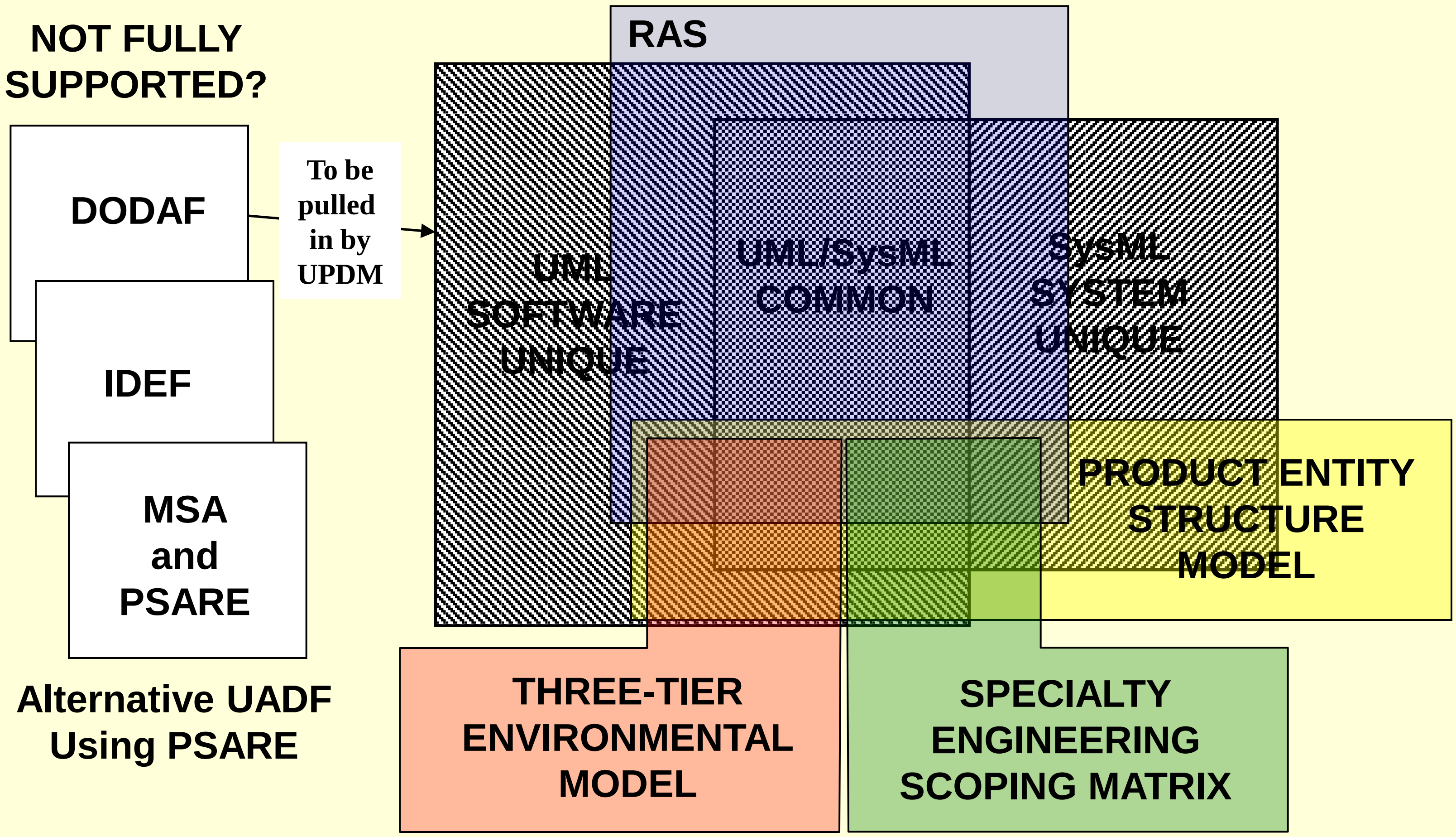
STRUCTURED ANALYSIS

FREESTYLE OR AD HOC



THE CUSTOMER INTERVIEW APPROACH

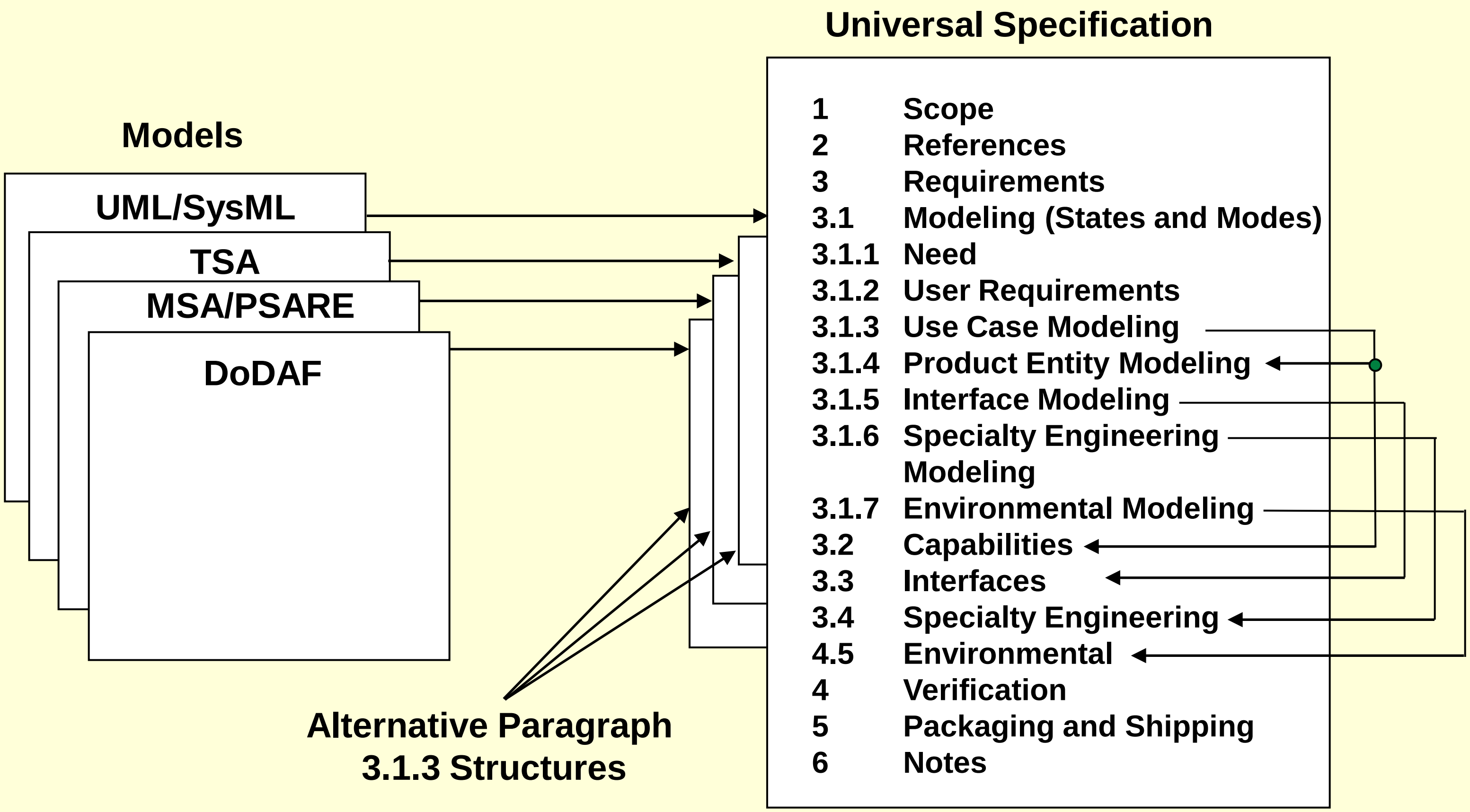
A Universal Model for the Future?



UADF Modeling Artifacts Summary

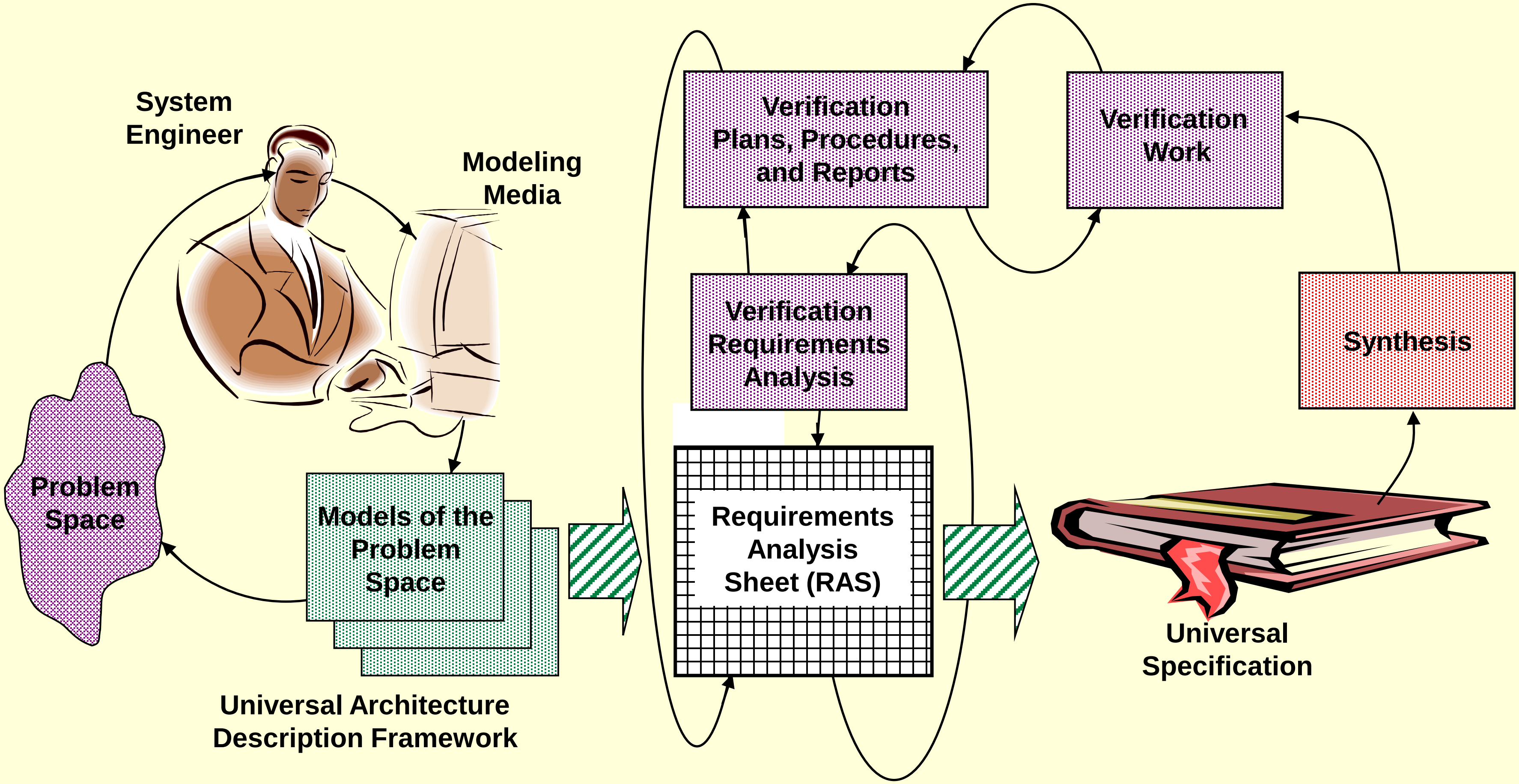
		UNIVERSAL ARCHITECTURE DESCRIPRION FRAMEWORKS			
		FUNCTIONAL	MSA-PSARE	UML-SYSML	
PRODUCT ENTITY DEFINITION		ALLOCATION	SUPPER BUBBLES	SEQ DIAGRAM LIFE LINES OR ACTIVITY DIAG SWIM LANES	
PRODUCT ENTITY REQUIREMENTS	PERFORMANCE REQUIREMENTS	FLOW DIAGRAMS, IDEF-0, BEHAVIOR DIAGRAMS, OR ENHANCED FFBD	DFD/CFD, P-SPEC, C-SPEC, AND DATA DICTONARY	USE CASE, SEQUENCE, ACTIVITY, AND STATE DIAGRAMS	
	INTERFACE REQUIREMENTS	N-SQUARE AND SCHEMATIC BLOCK DIAGRAM	DIRECTED LINE SEGMENTS THAT CROSS SUPER BUBBLE BOUNDARIES	DIRECTED LINE SEGMENTS THAT JOIN LIFE LINES	
	SPECIALTY ENGINEERING REQUIREMENTS	SPECIALTY ENGINEERING SCOPING MATRIX	SPECIALTY ENGINEERING SCOPING MATRIX	SPECIALTY ENGINEERING SCOPING MATRIX	CONSTRAINTS MODELING EXTENSION
	ENVIRONMENTAL REQUIREMENTS	THREE-LAYER MODEL	THREE-LAYER MODEL	THREE-LAYER MODEL	
SOLUTION SPACE MODELS		PRODUCT ENTITY BLOCK AND SCHEMATIC BLOCK DIAGRAM	ARCHITECTURE MODEL	CLASS/OBJECT DIAGRAM, COMPONENT DIAGRAM, DEPLOYMENT DIAGRAM, BLOCK DEFINITION DIAGRAM, AND INTERNAL BLOCK DIAGRAM	

Model Results Flow Into Specification Content



Development Life Cycle Overview

From a Universal Architecture Perspective



JOG SYSTEM ENGINEERING
GRAND SYSTEMS DEVELOPMENT TRAINING PROGRAM
TUTORIAL

UNIVERSAL ARCHITECTURE DESCRIPTION FRAMEWORK

TRADITIONAL STRUCTURED ANALYSIS OVERVIEW

Hardware and Systems Analysis Models

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

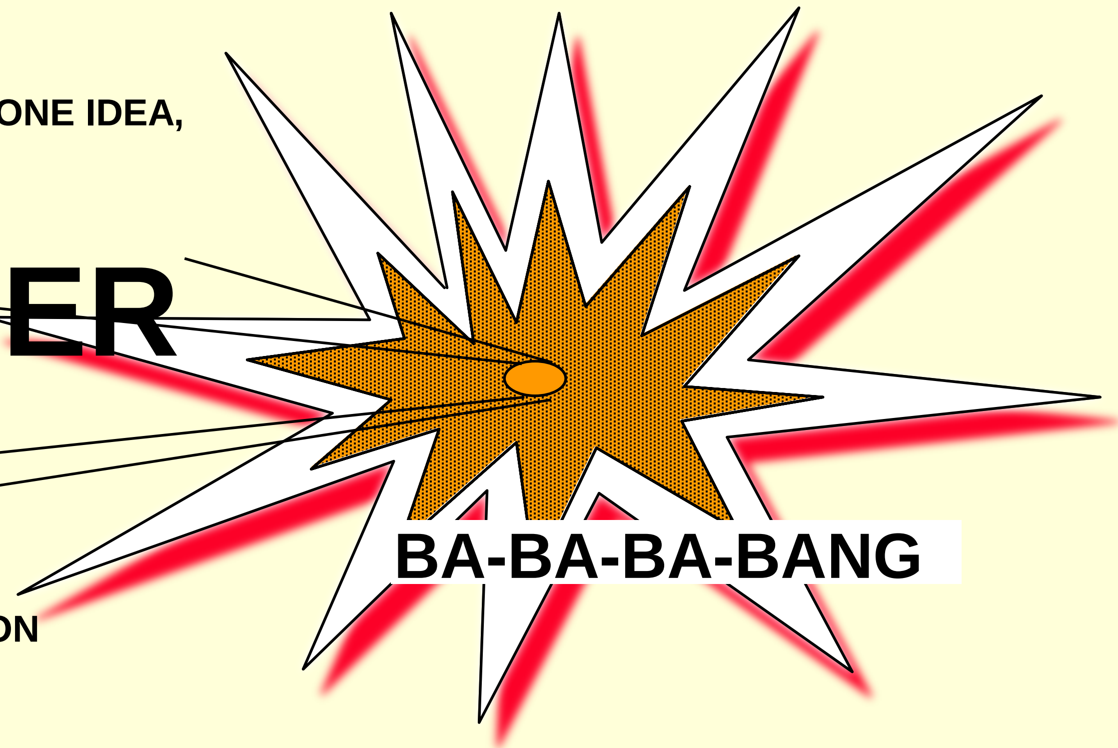
The Big Bang Theory Of System Development

THE TRADITIONAL SRA APPROACH

EVERYTHING FLOWS FROM ONE IDEA,

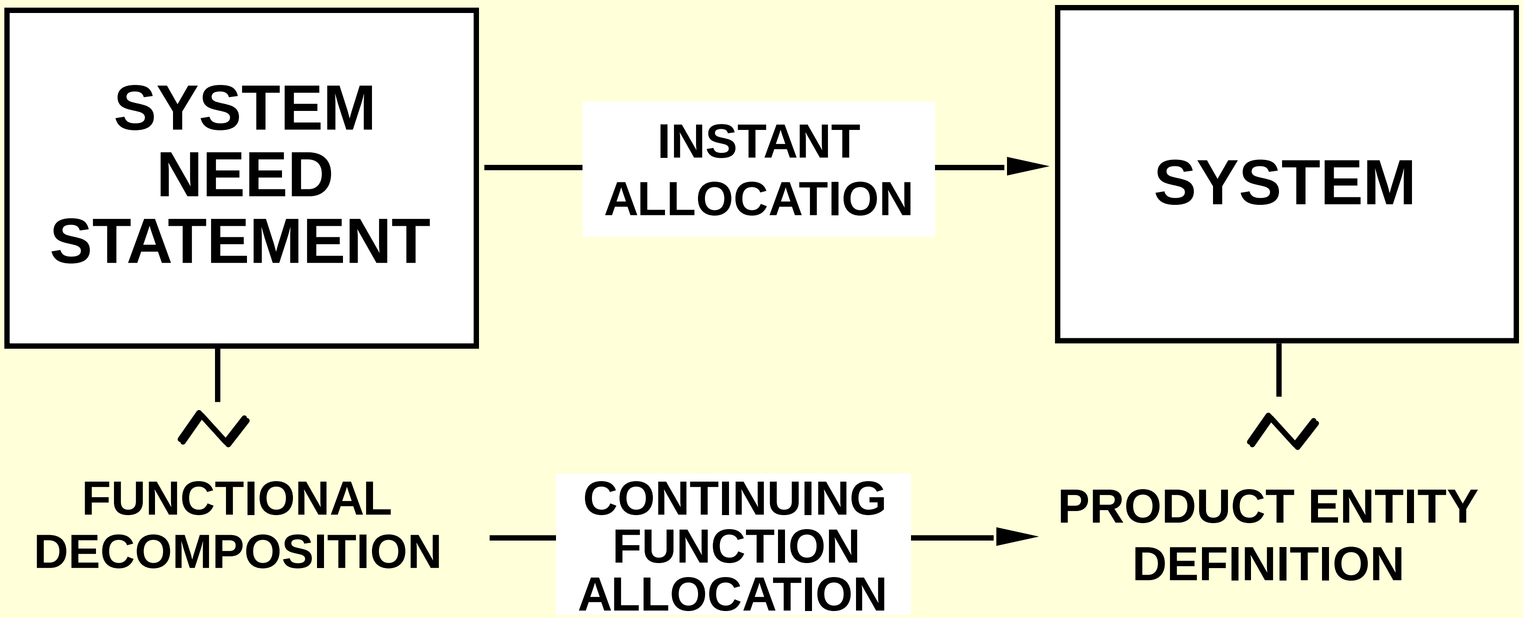
THE

CUSTOMER NEED

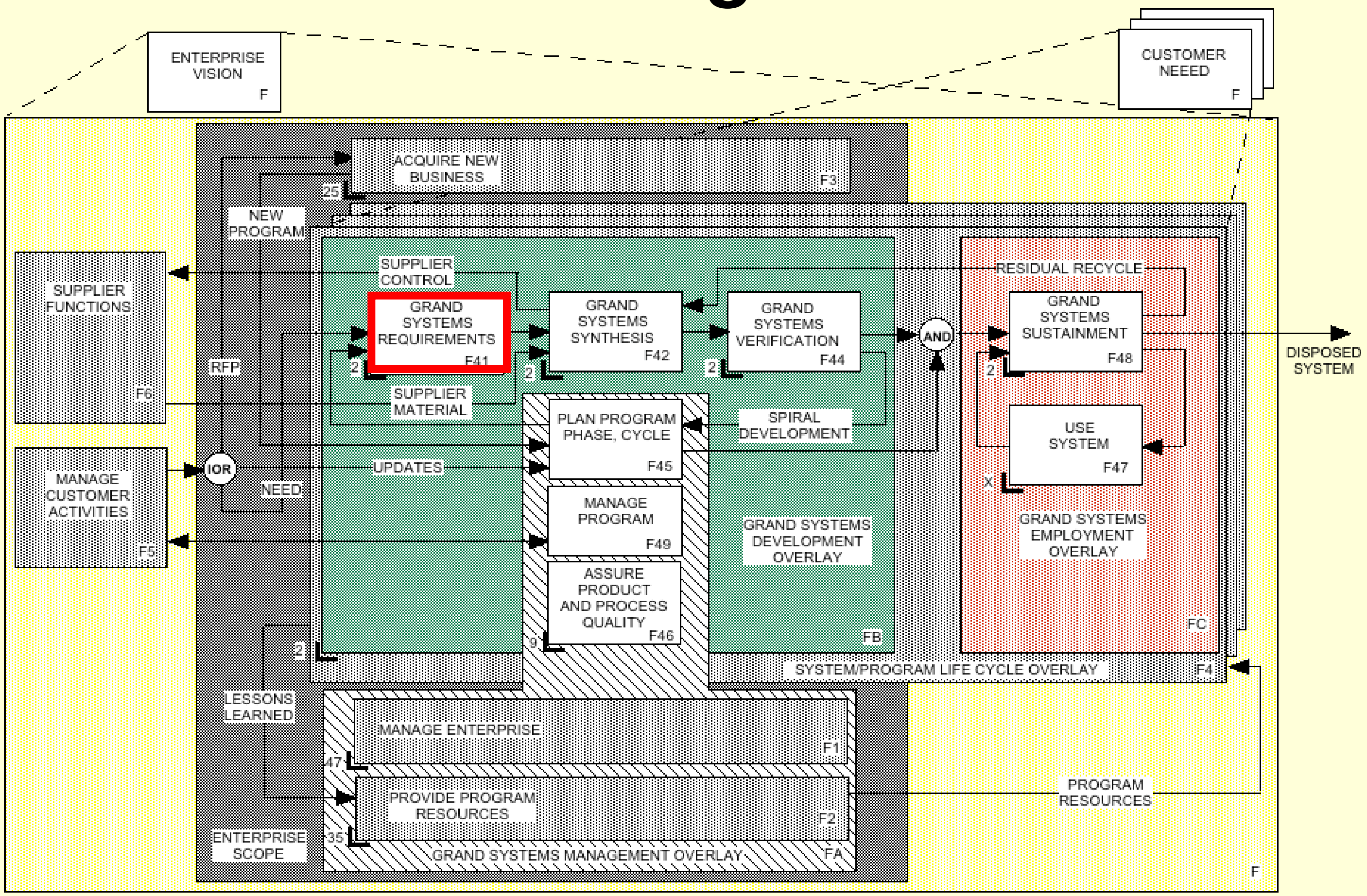


THE ULTIMATE FUNCTION

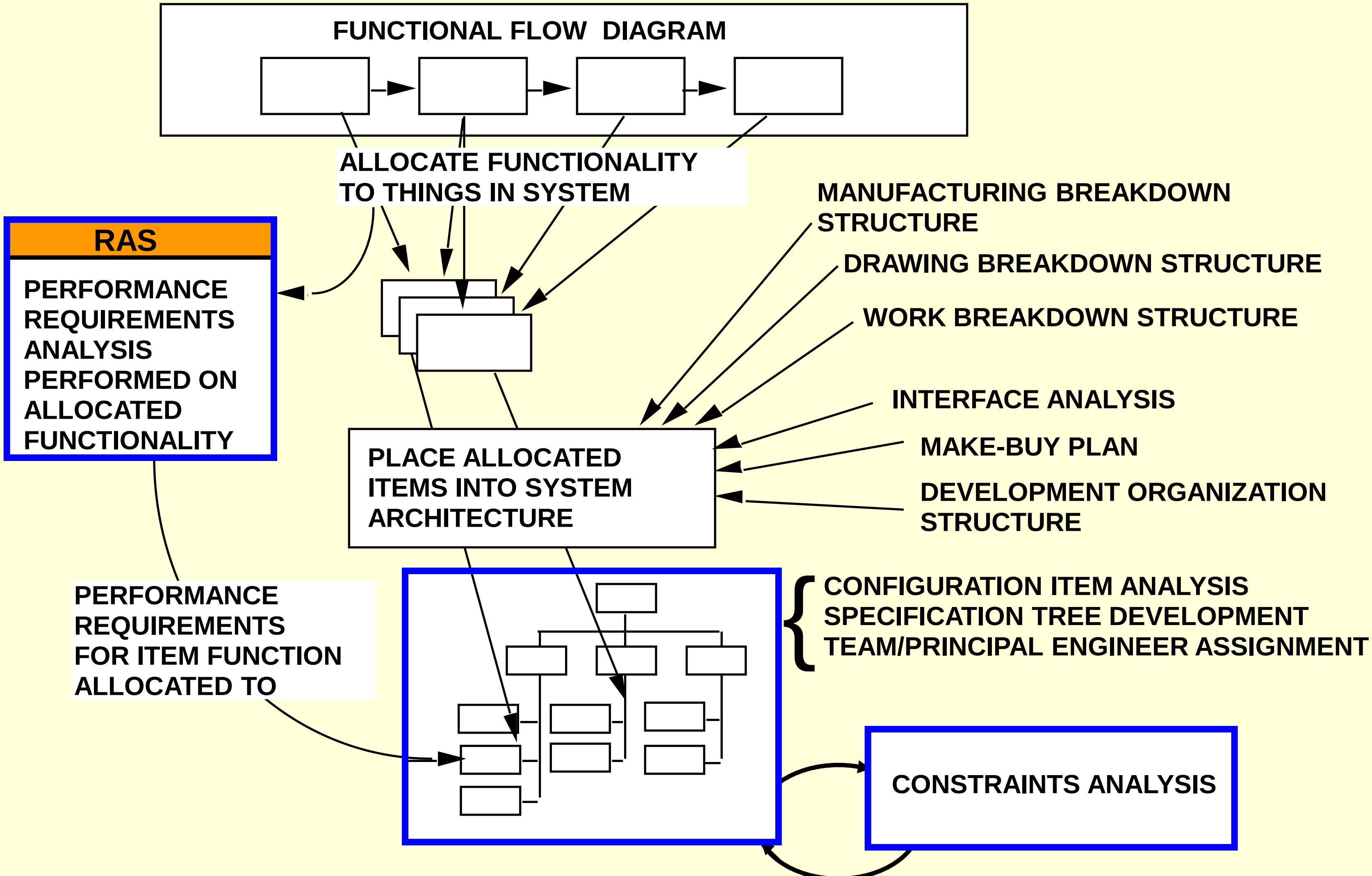
The Beginning of Functional Decomposition



The Current System Development Paradigm

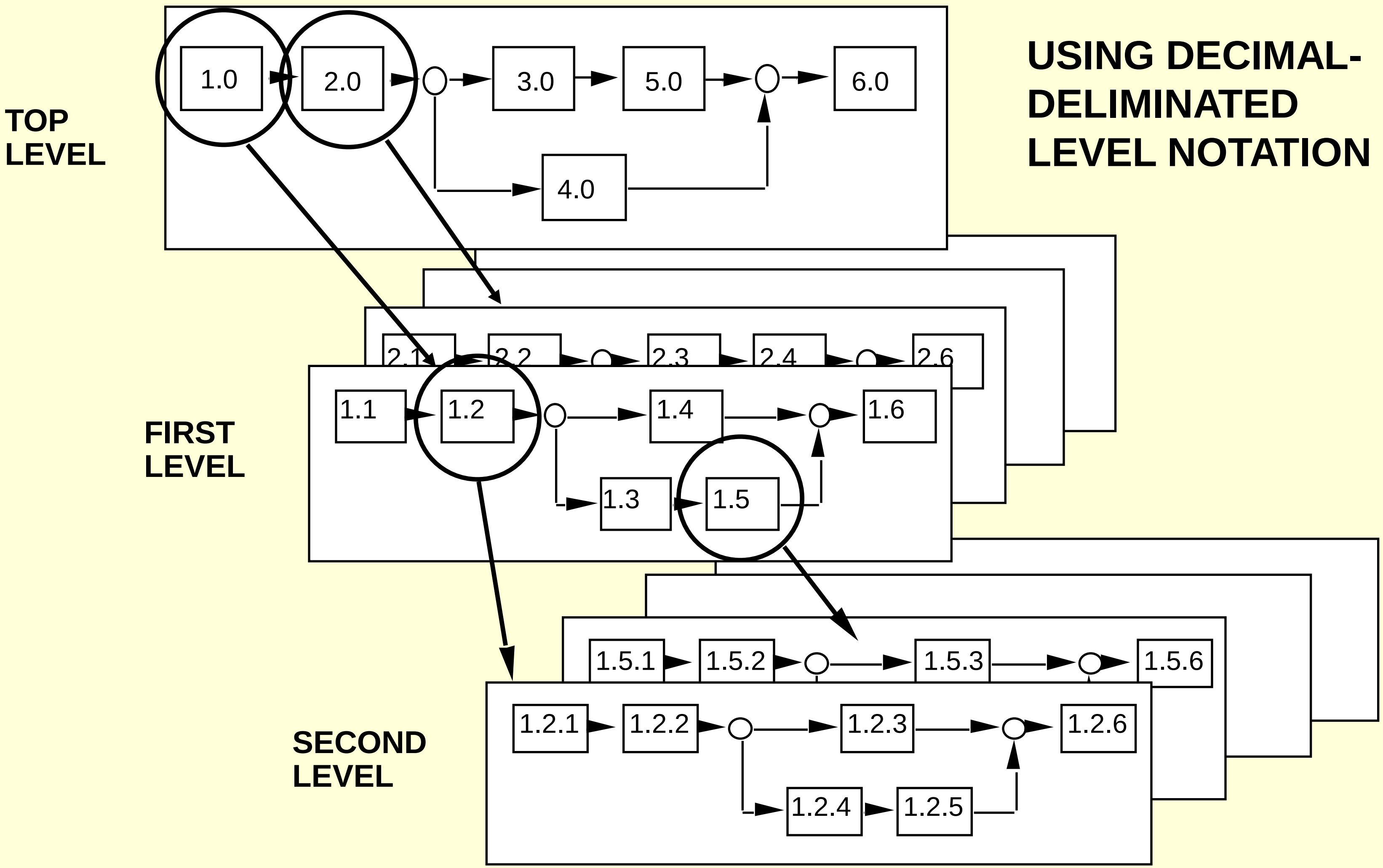


Functional Analysis and Allocation



Functional Flow Diagramming Levels

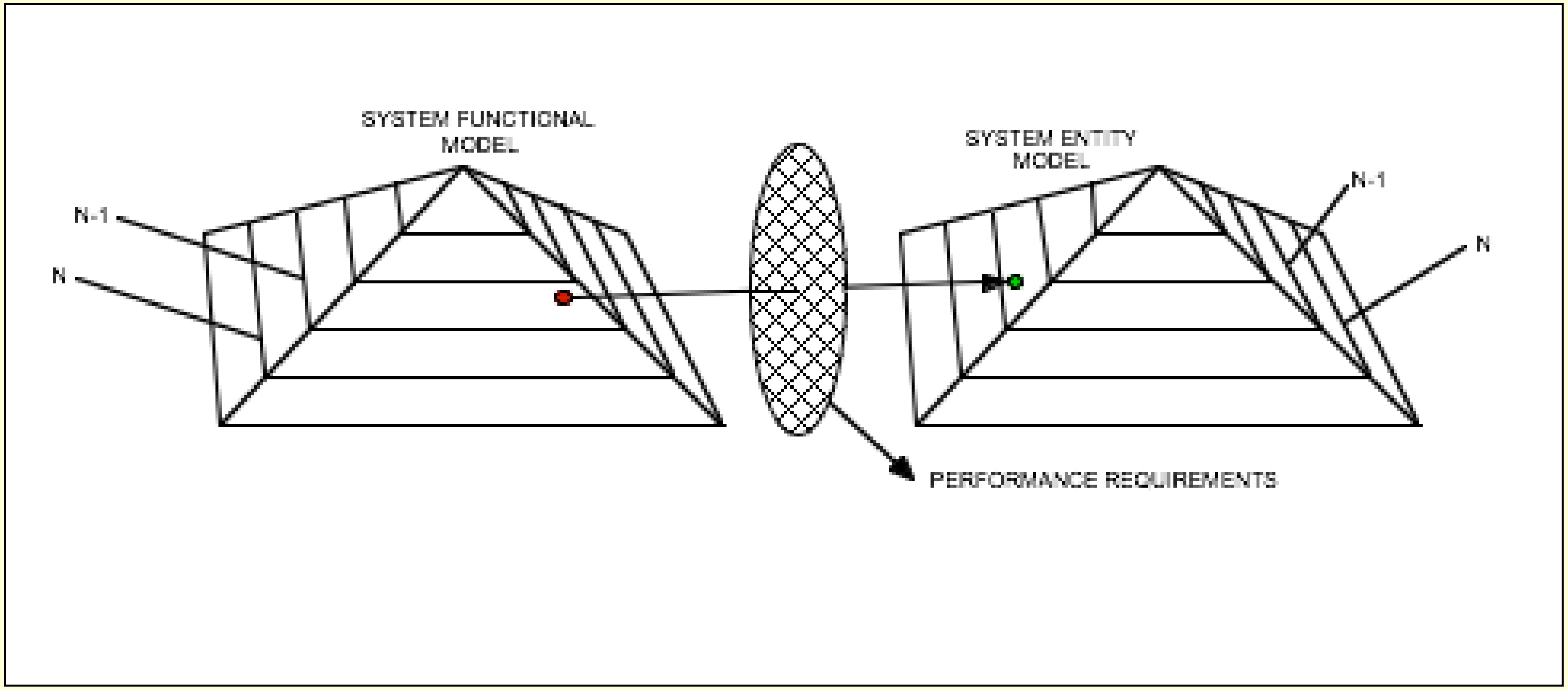
USING DECIMAL-DELIMINATED LEVEL NOTATION



Functional Allocation Pacing Alternatives

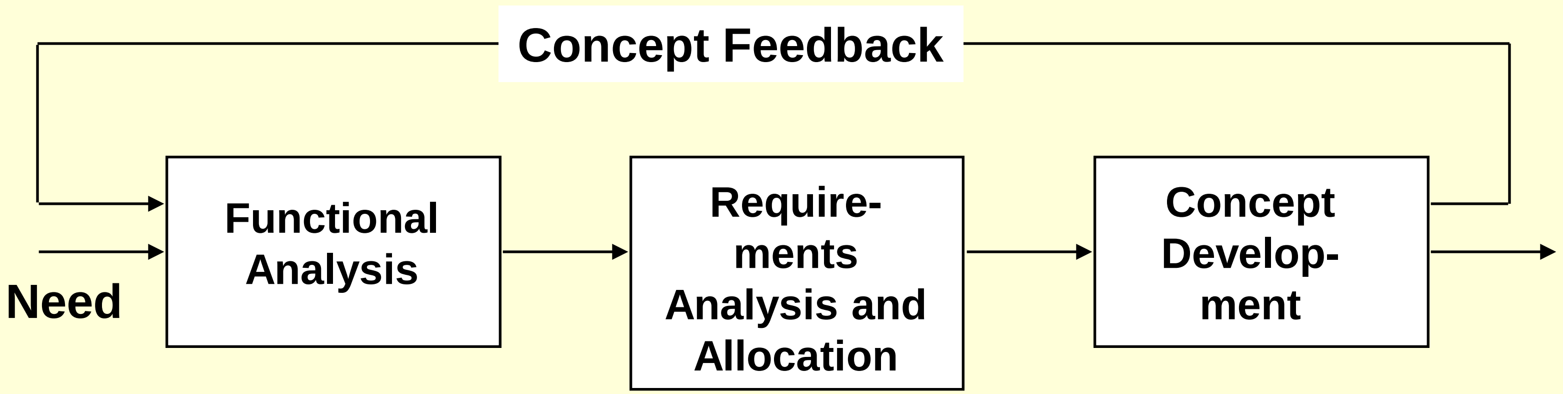
- **Serial performance**
  - Functional analysis until complete then allocate and conceptualize
- **Instant allocation**
  - See function, allocate function
- **Layered allocation**
  - FRAT concept
- **Progressive allocation**

FRAT Layered Perspective



From the work of Bernard Morias and Brian Mar.

Progressive Tuning of the Functional Analysis

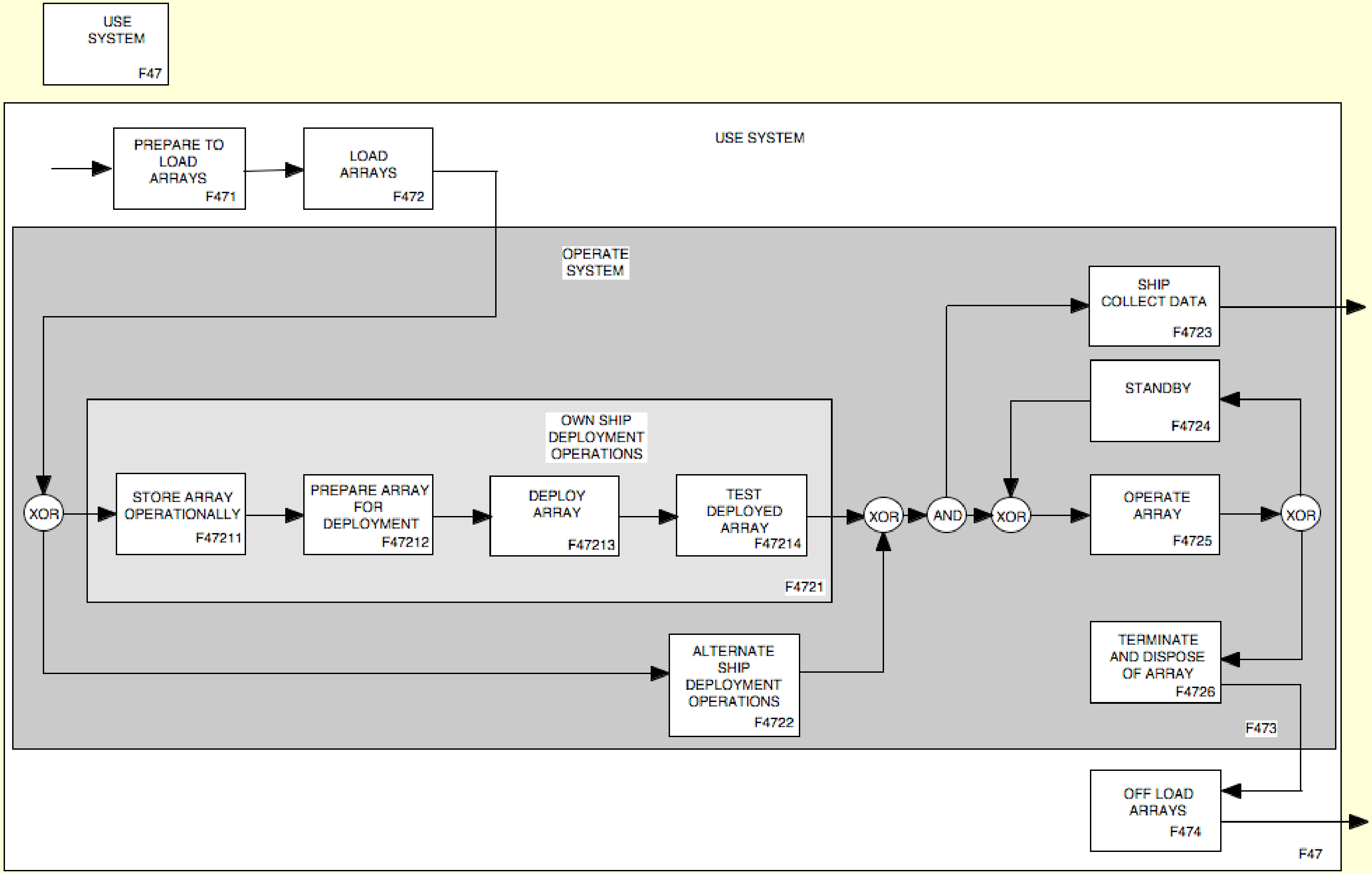


Lower level functional analysis guided by higher level concept definition

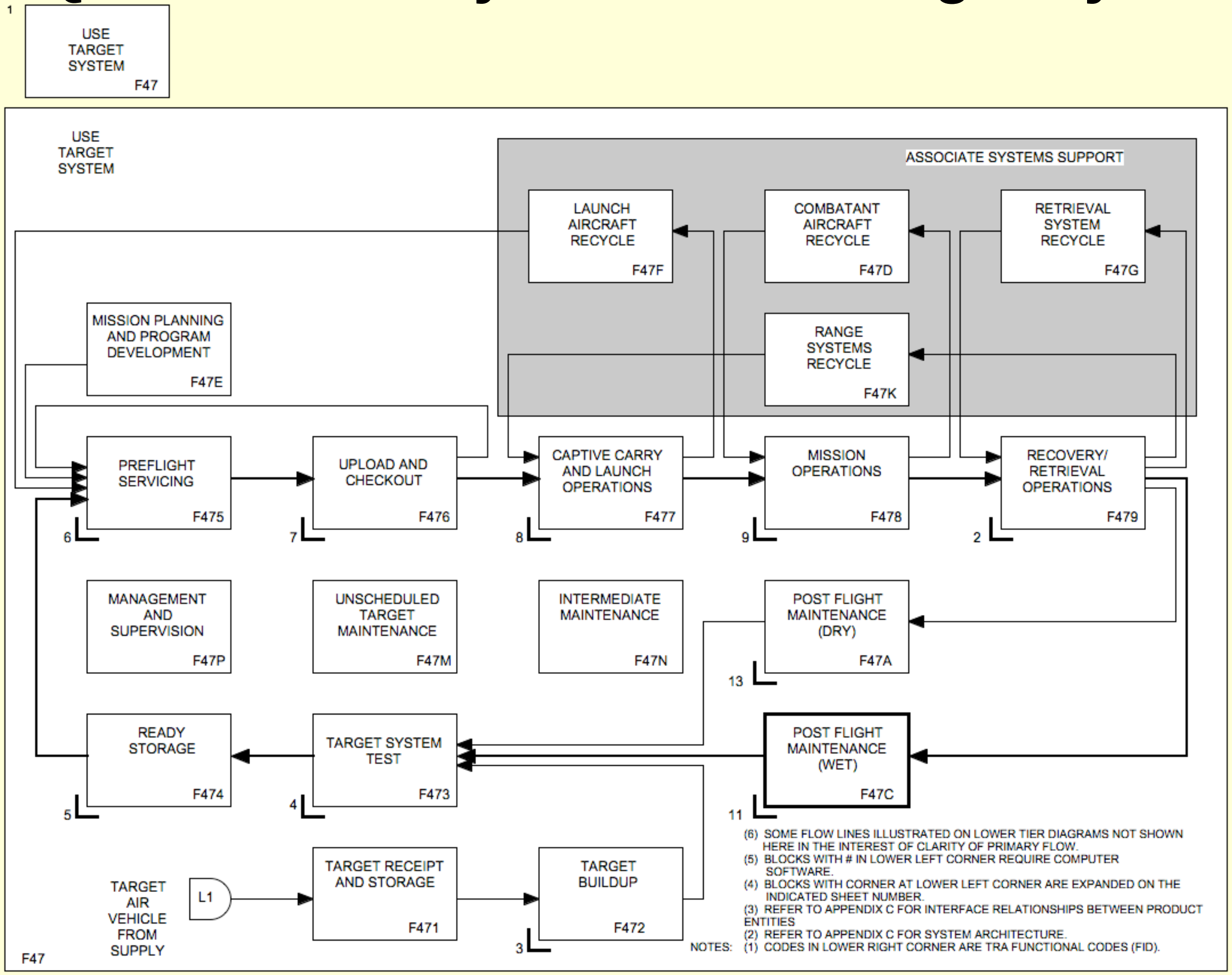
Results in tuning the action oriented functional flow diagram to a physical process diagram adequate for detailed logistics support analysis at lower tiers and environmental use profiling.

Use System Decomposition

Deployable Sonar Array

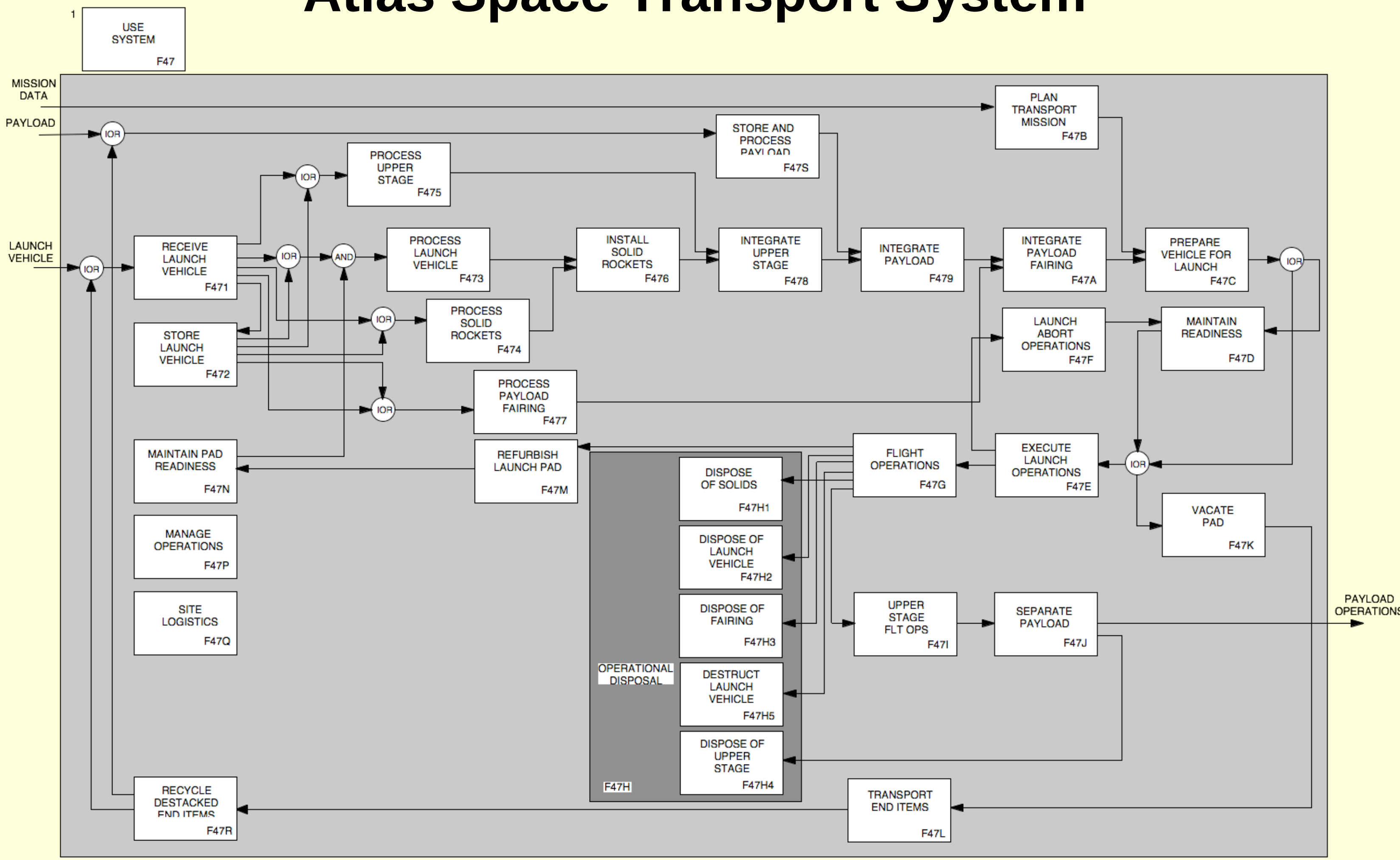


Use System Decomposition
AQM-81 USAF/Ryan Firebolt Target System



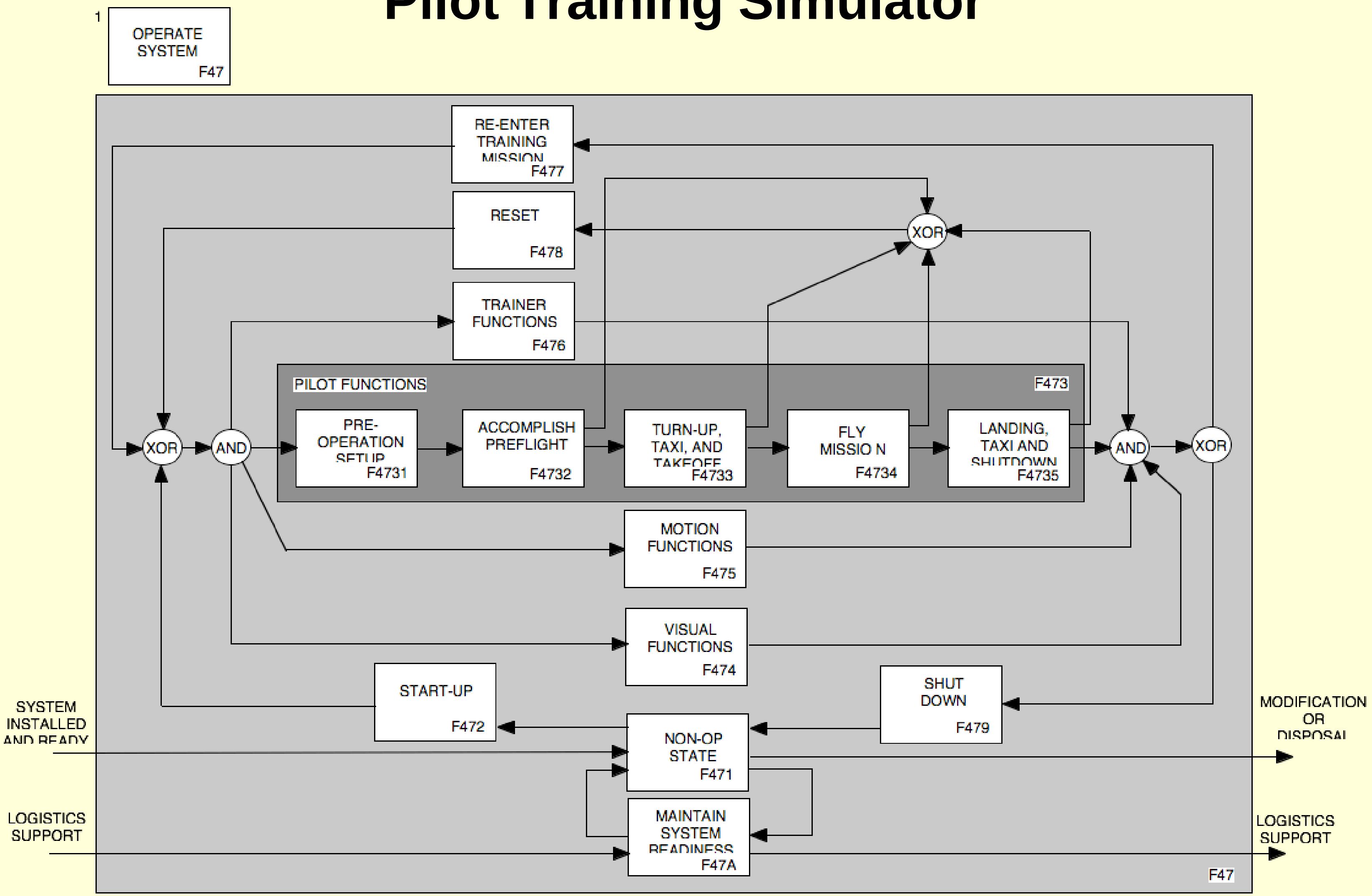
Use System Decomposition

Atlas Space Transport System

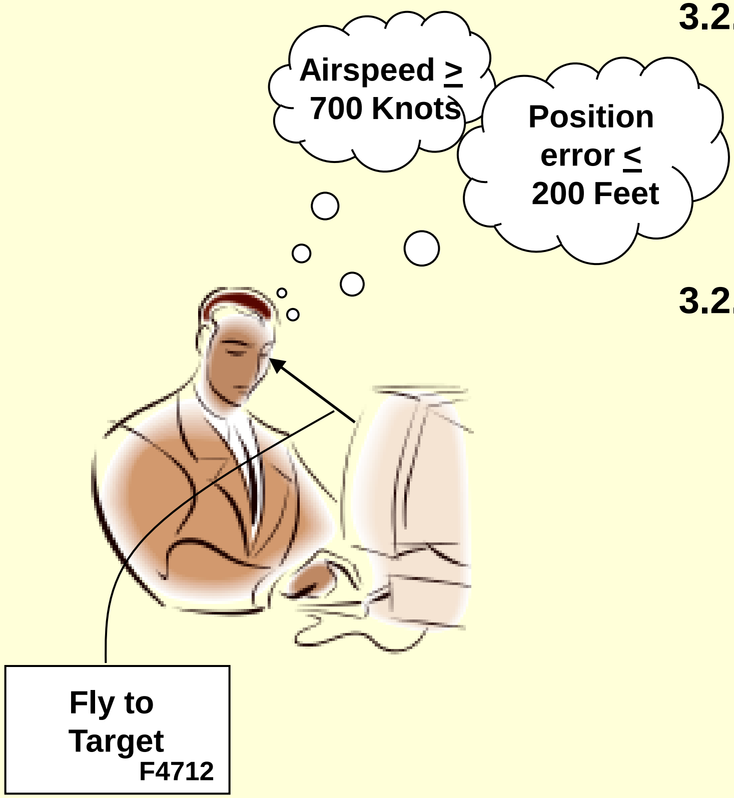


Use System Decomposition

Pilot Training Simulator



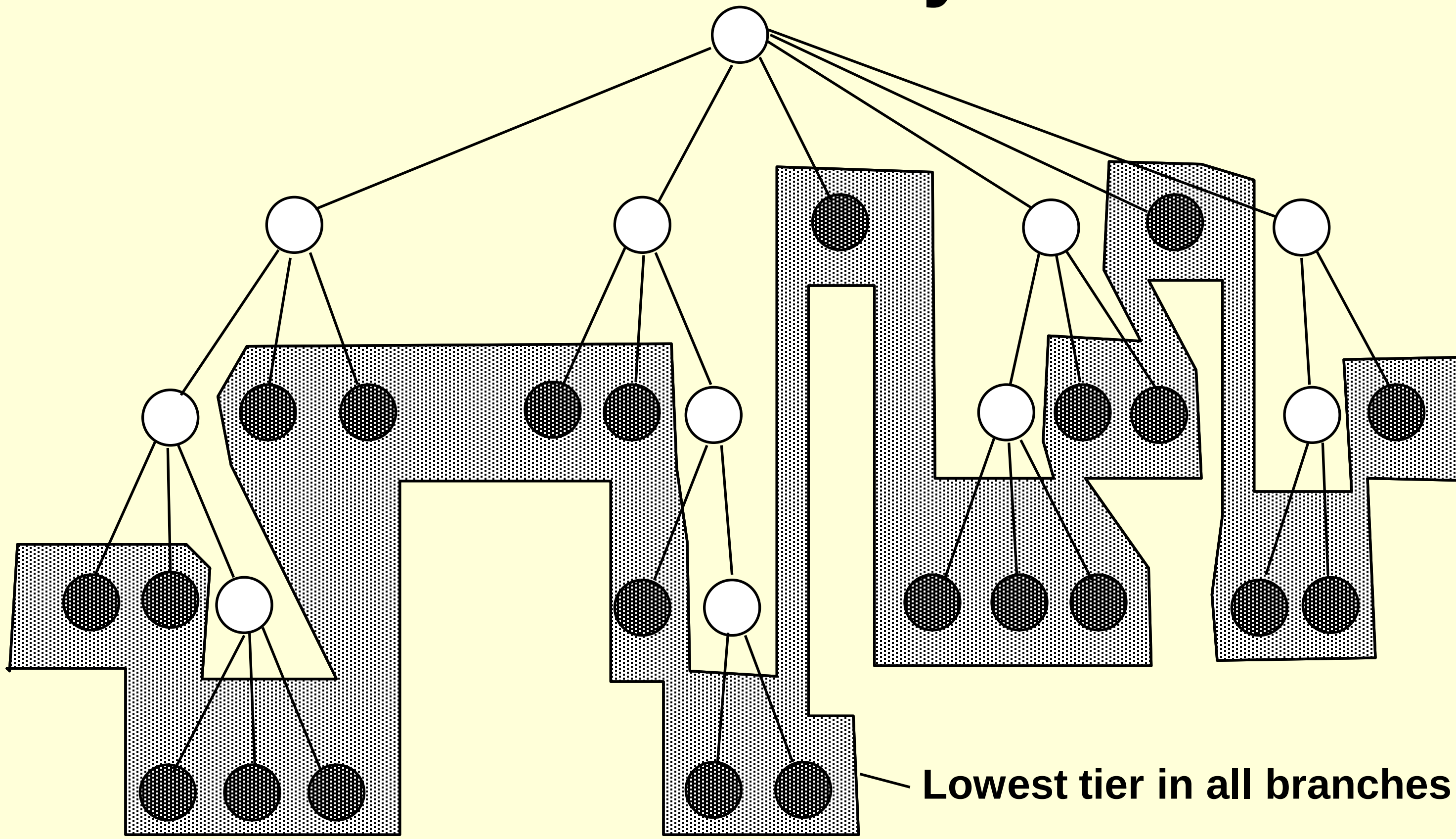
Deriving Performance Requirements



3.2.1.1 Aircraft shall be capable of flight at an airspeed $\geq$ 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.

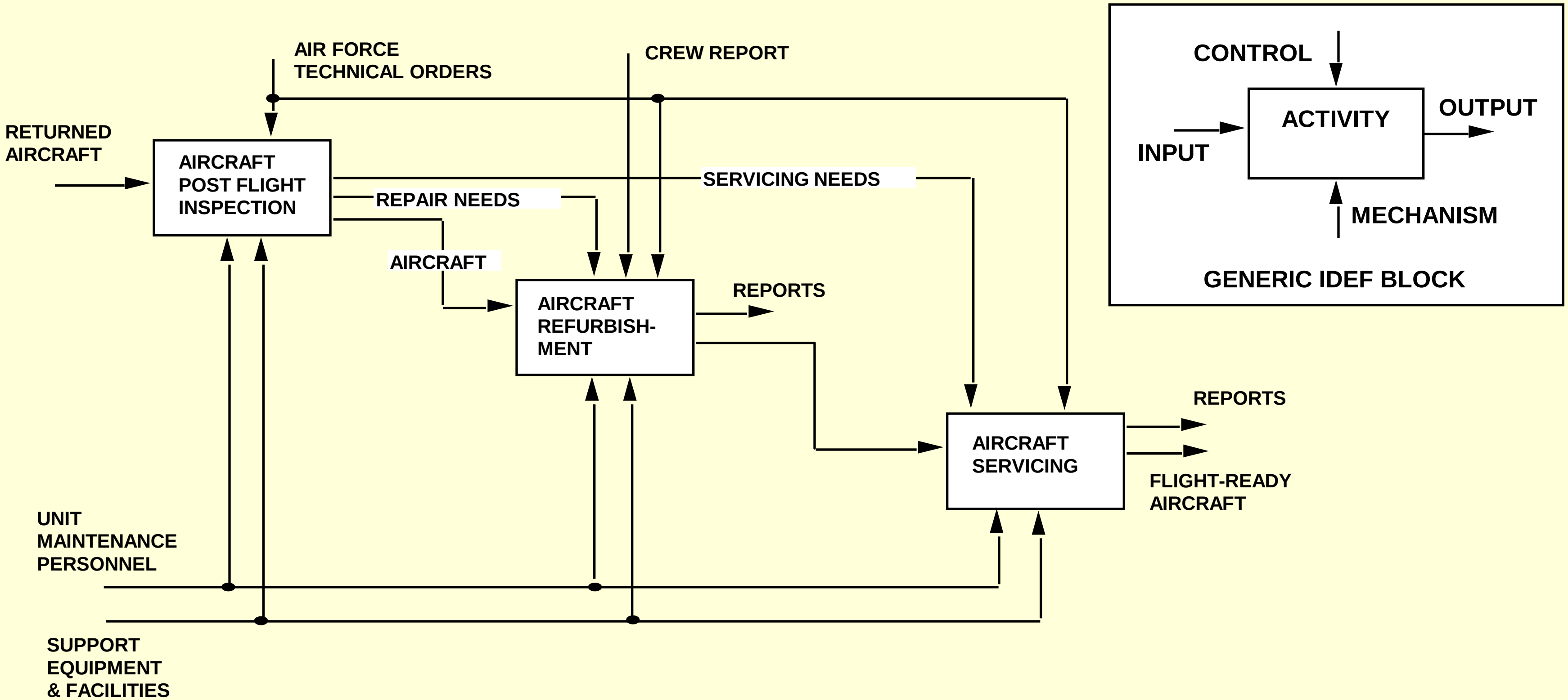
The End of Functional Decomposition In the Product Entity Structure



- Buy it at that level
- Will surrender to design by one of your design teams

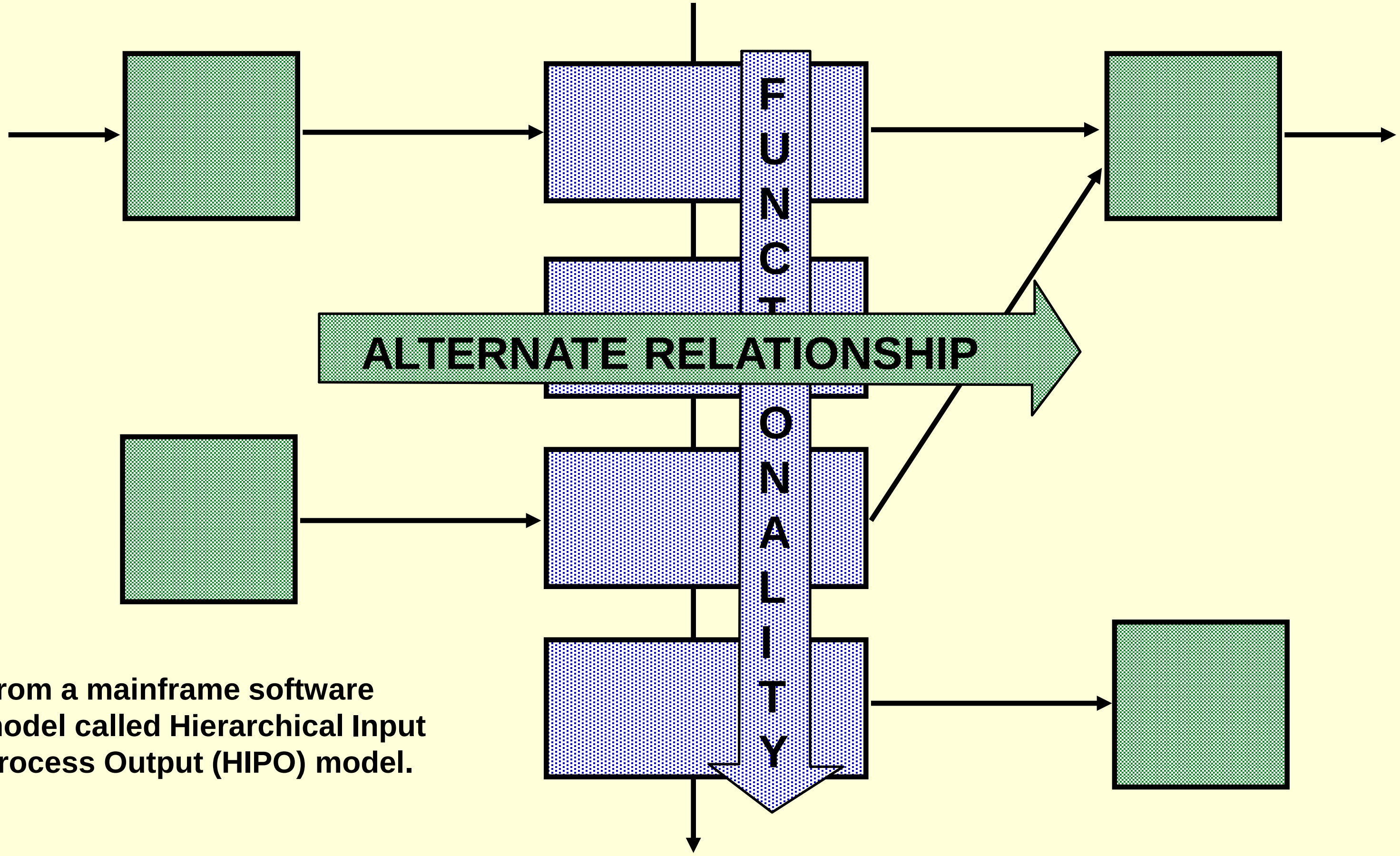
Alternative Functional Models

IDEF-Ø Diagram



Alternative Functional Models

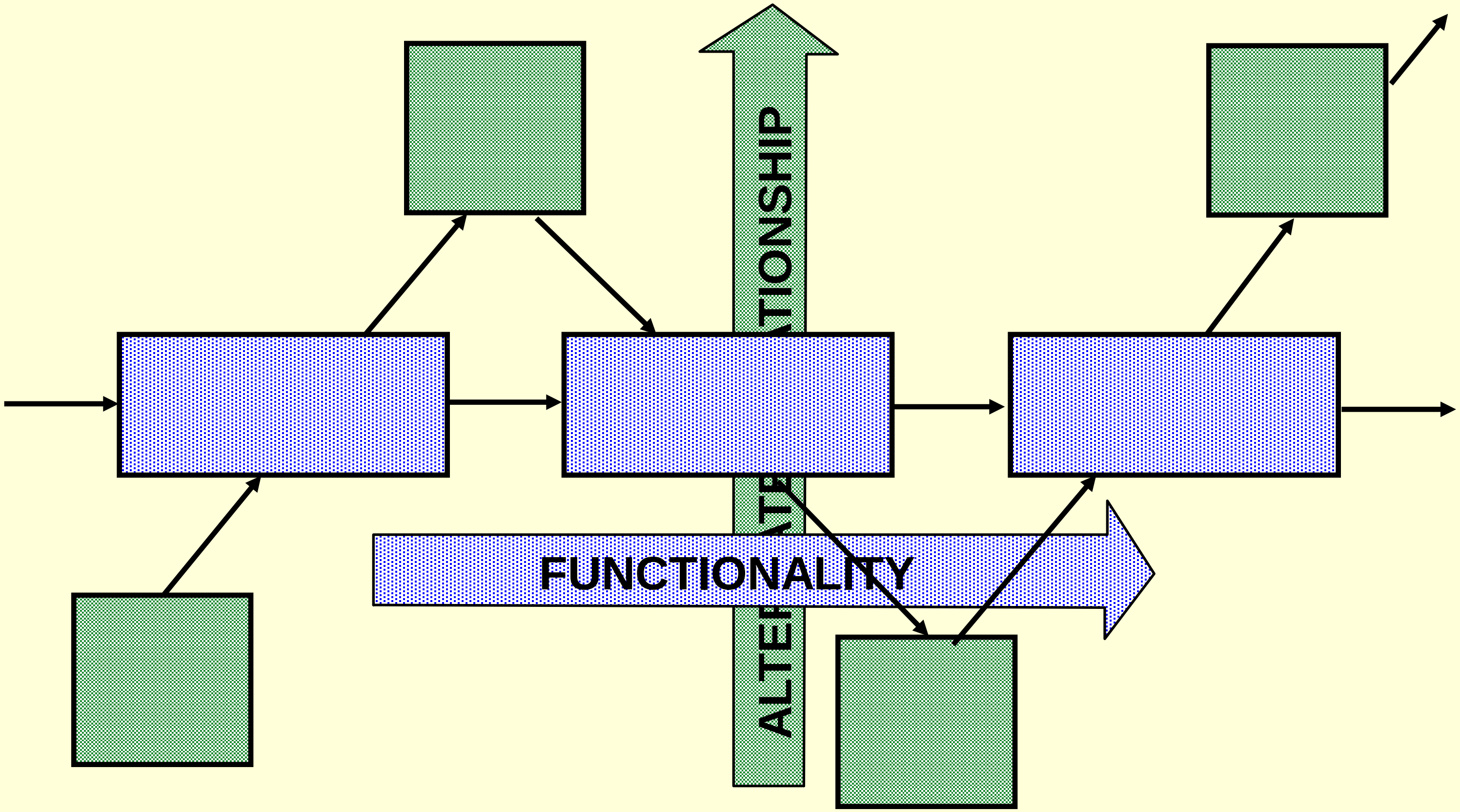
Behavioral Diagramming From RDD-100



From a mainframe software model called Hierarchical Input Process Output (HIPO) model.

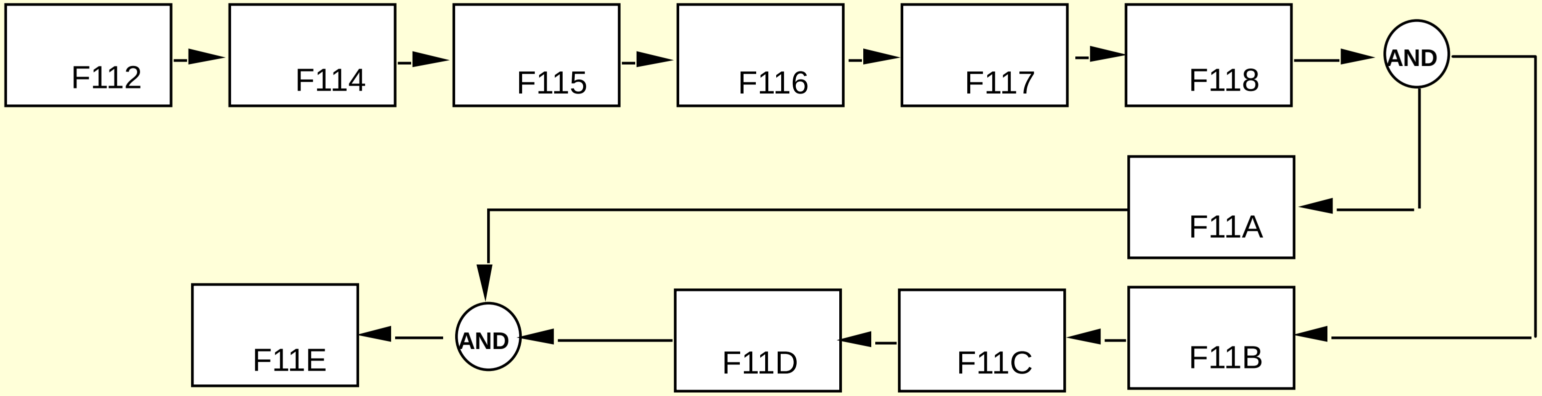
Alternative Functional Models

Enhanced Functional Flow Block Diagramming

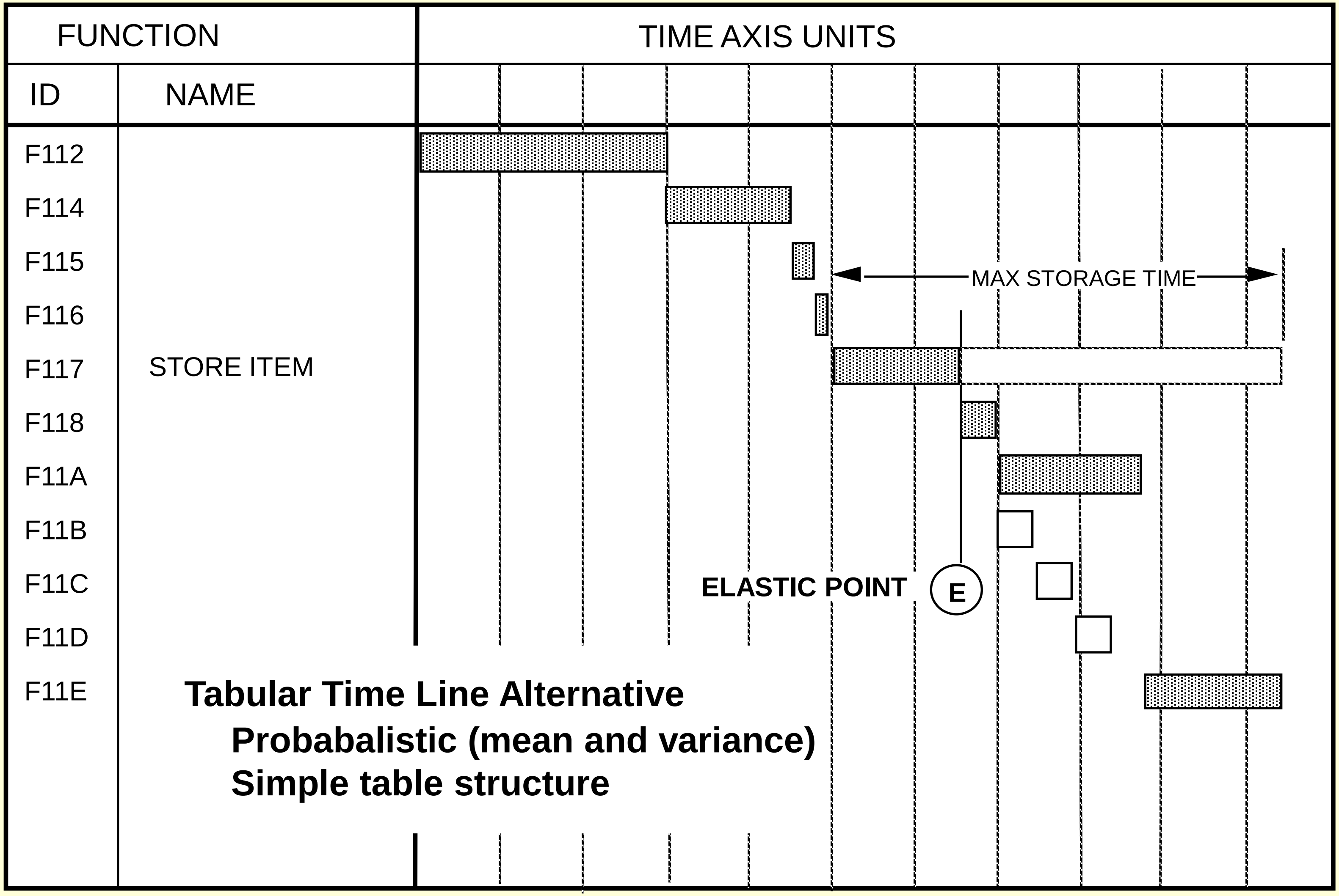


Timeline Diagramming

Assume for the moment that we have this functional flow diagram. How might we depict the timing requirements for these functions?



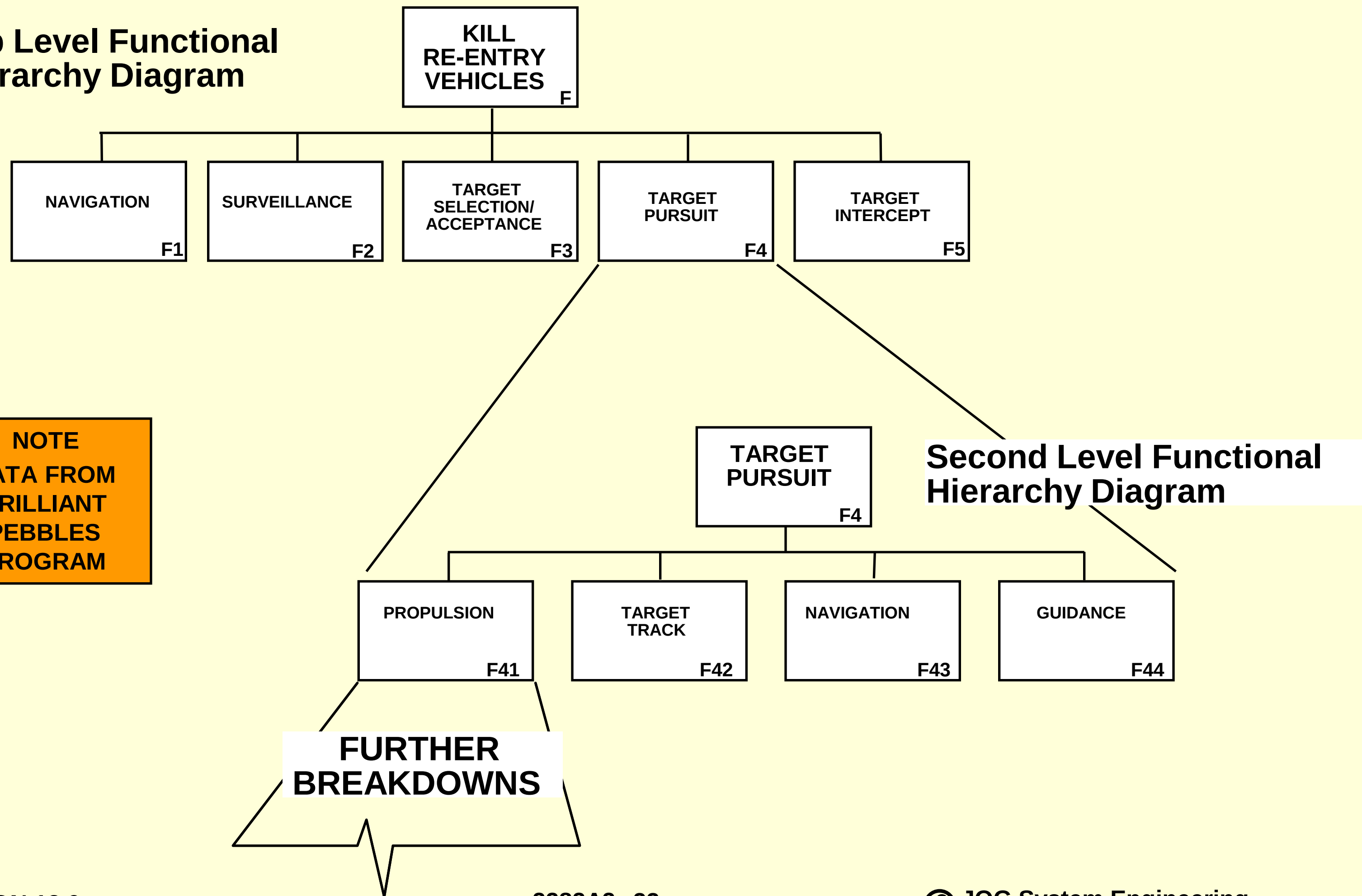
Timeline Diagramming



Alternative Functional Models

Hierarchical Functional Analysis

Top Level Functional Hierarchy Diagram

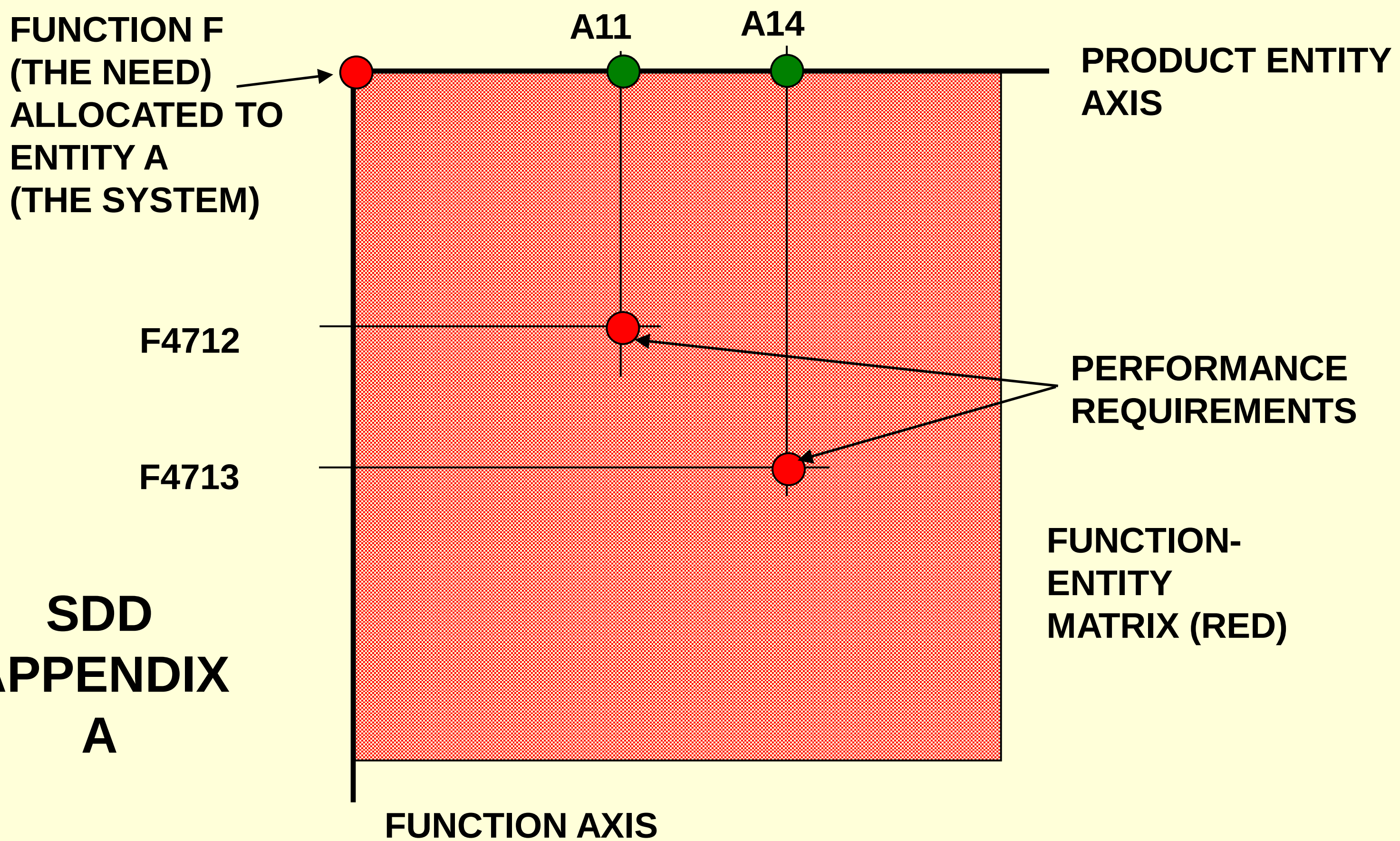


NOTE
DATA FROM BRILLIANT PEBBLES PROGRAM

Traditional RAS

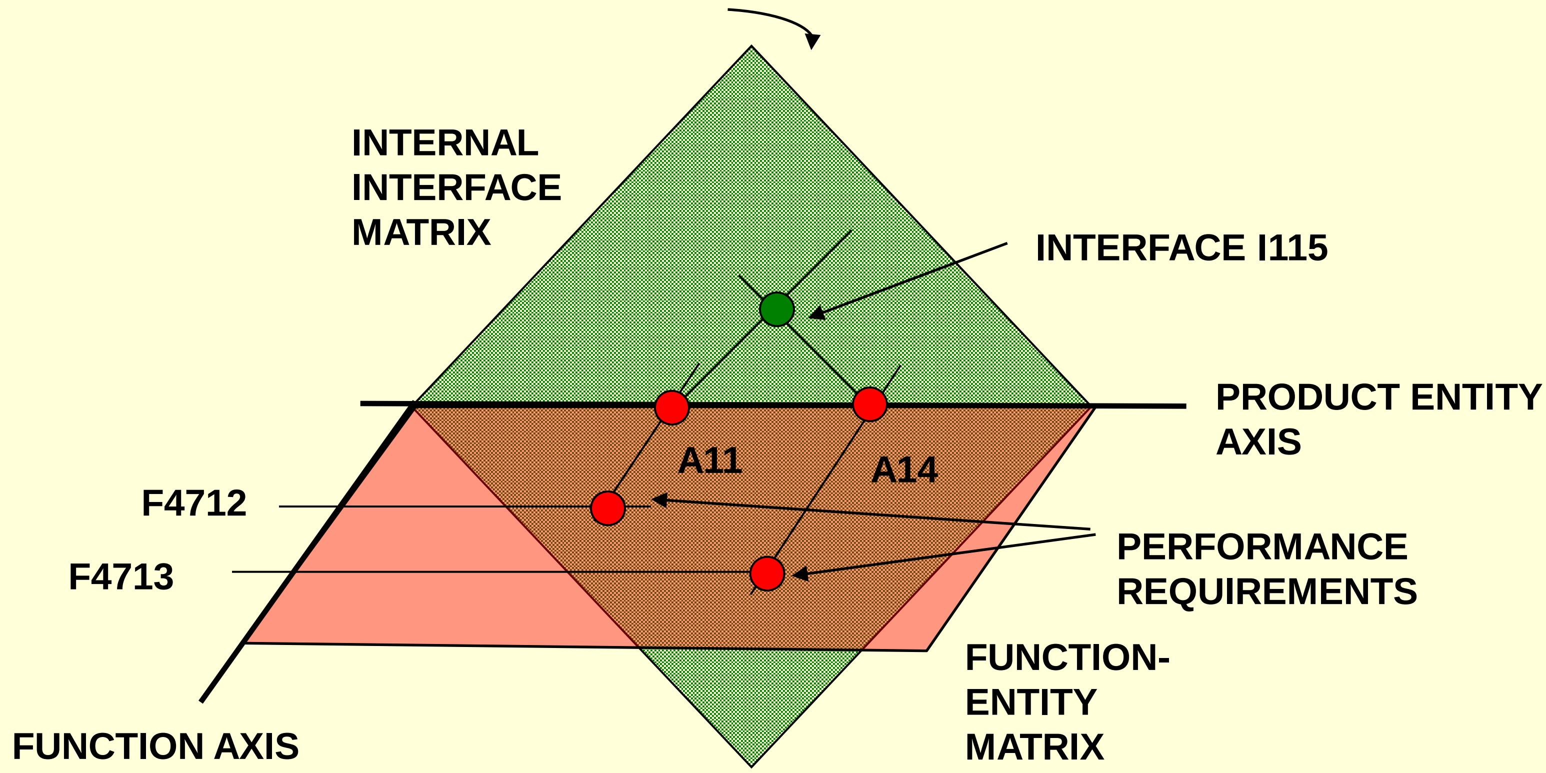
FUNCTION		PERFORMANCE REQUIREMENTS	PRODUCT ENTITY	
ID	NAME		ID	NAME
F4712 F4713 F4714 F4715	Flt to target	Airspeed $\geq$ 700 knots	A11 A14	Flight vehicle

Function-Entity Matrix (Traditional RAS)

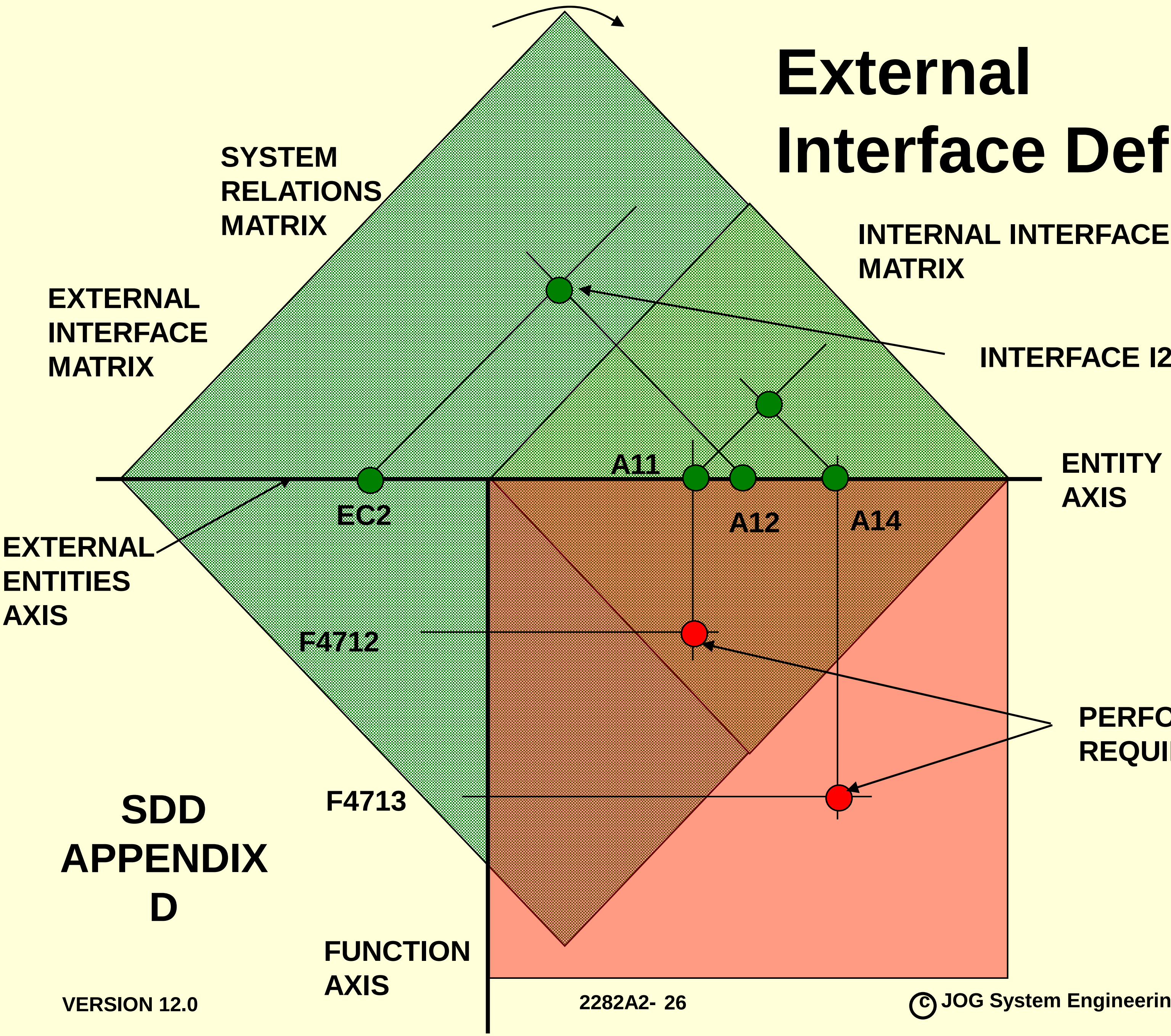


SDD APPENDIX A

RAS-Coordinated N-Square Diagram

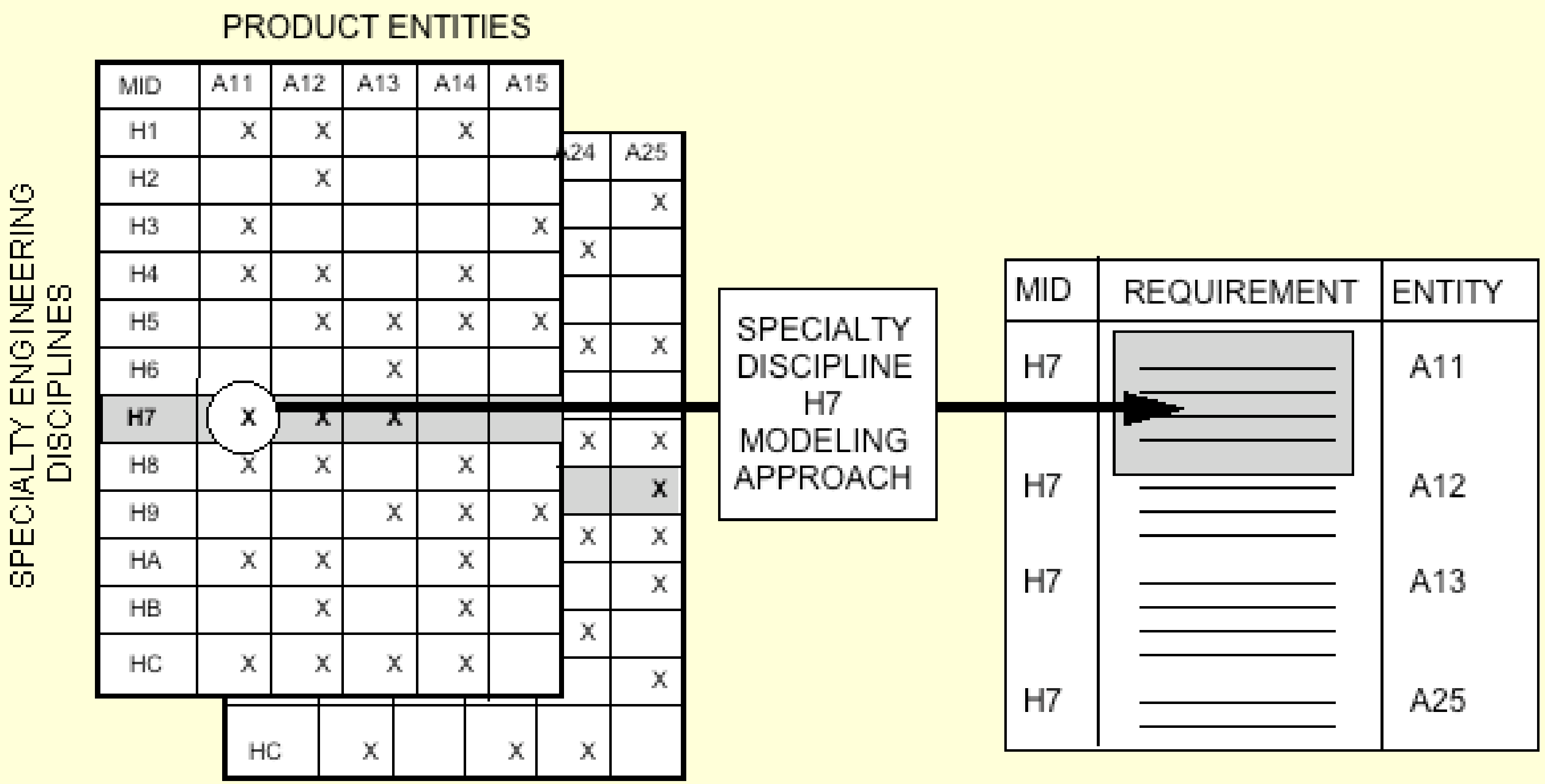


External Interface Definition



SDD APPENDIX D

Specialty Engineering Scoping Matrix

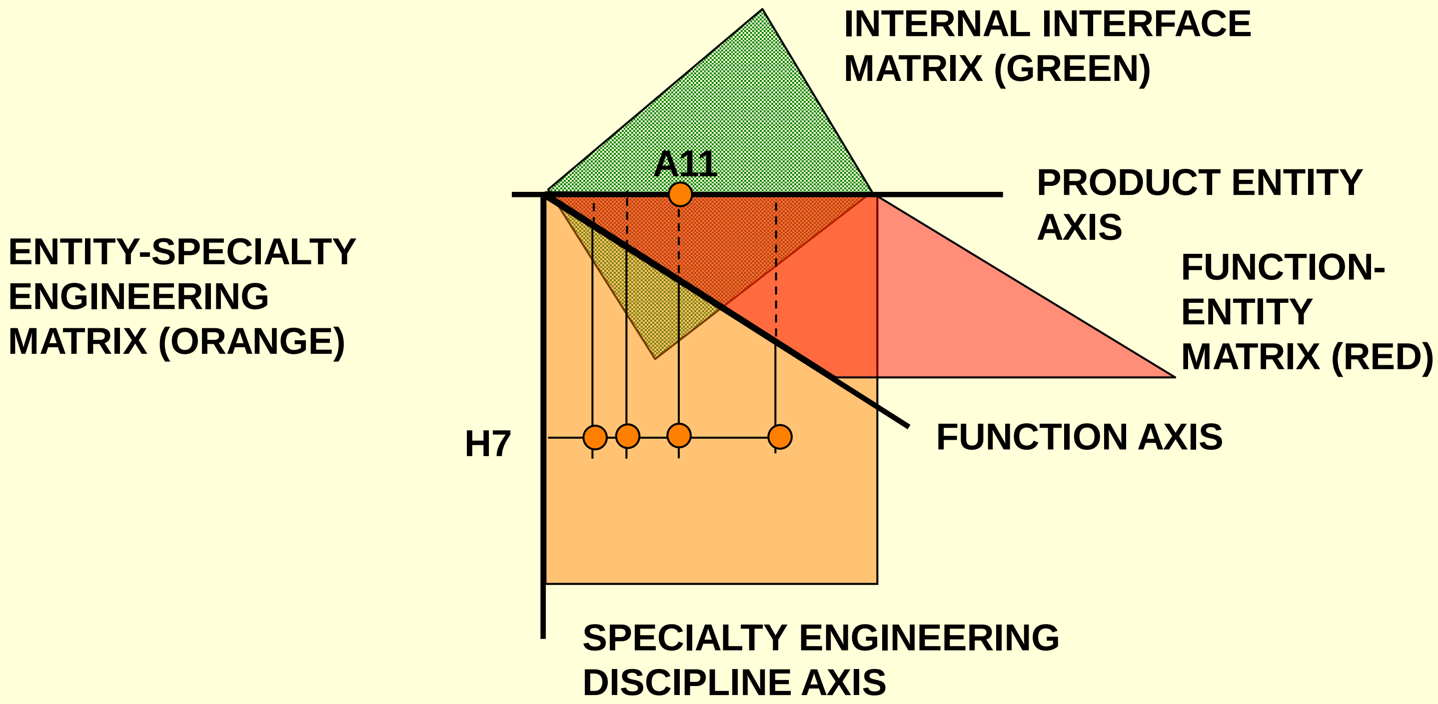


a. Specialty Engineering Scoping Matrix

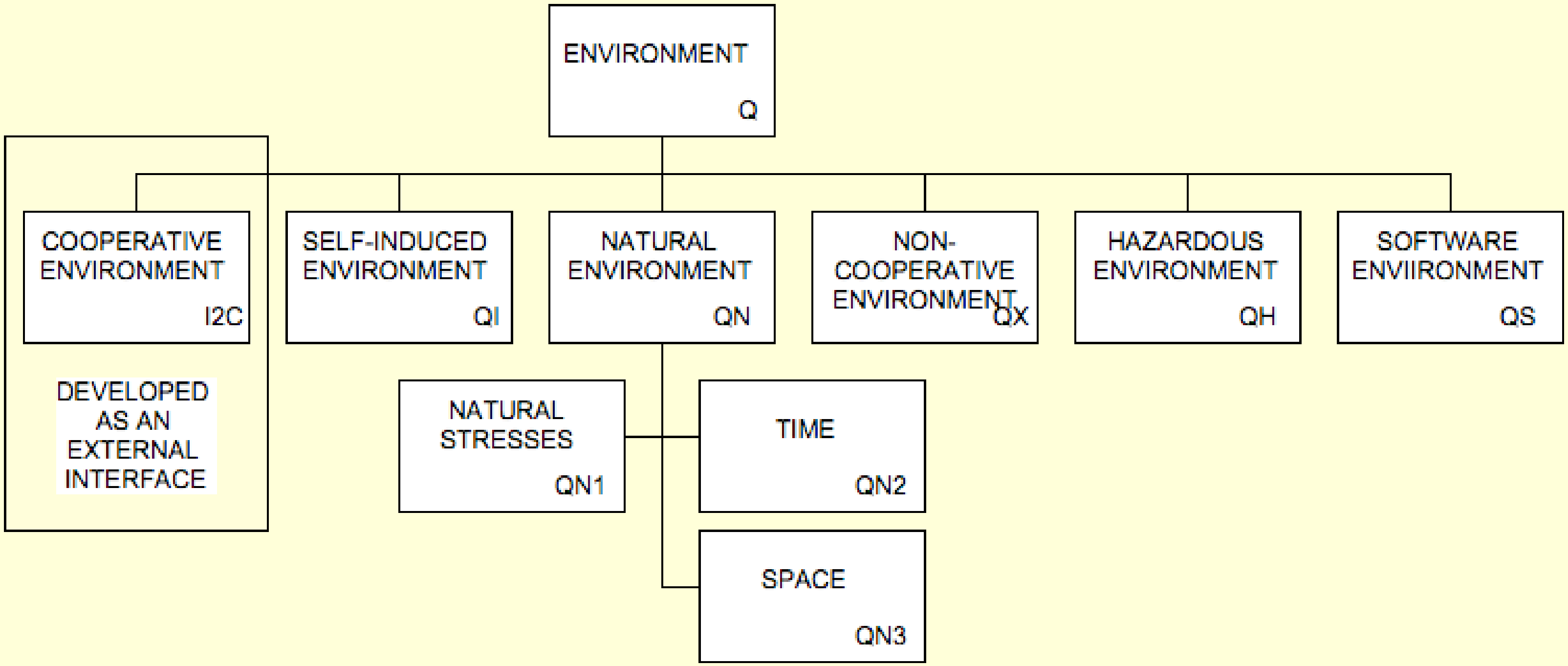
b. Requirements Analysis Sheet (RAS)

Specialty Engineering Allocation

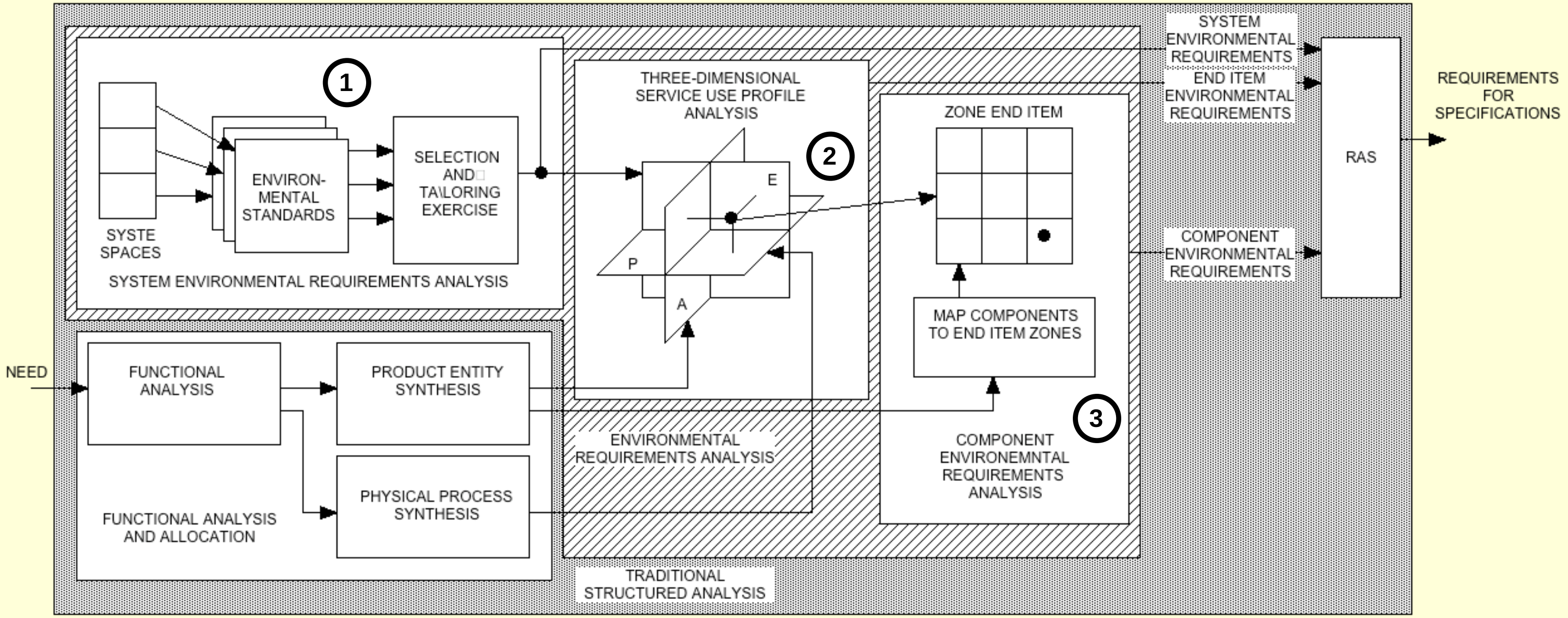
SPECIALTY DISCIPLINE H7 ALLOCATED TO ENTITY A11



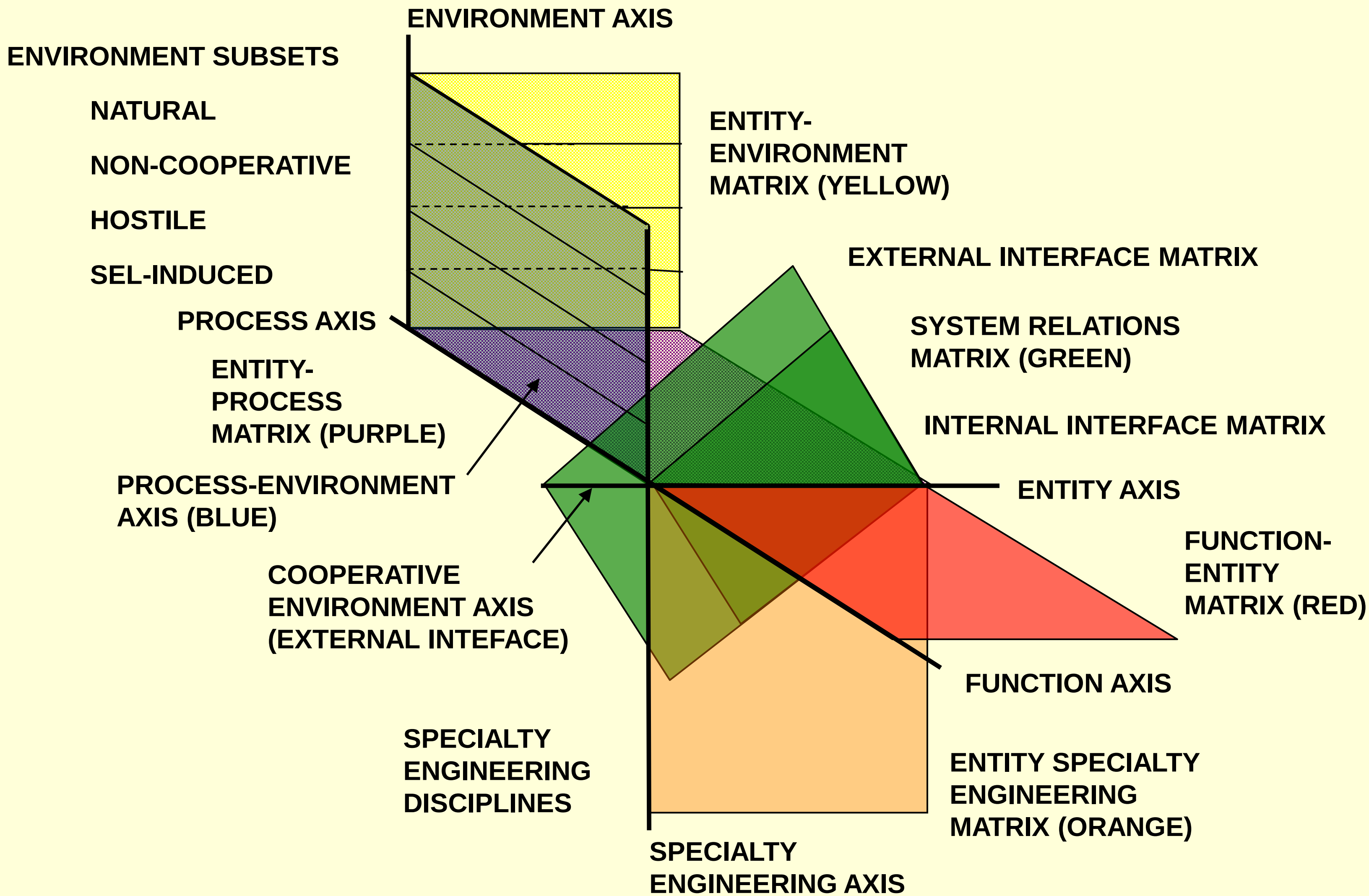
System Environmental Classes



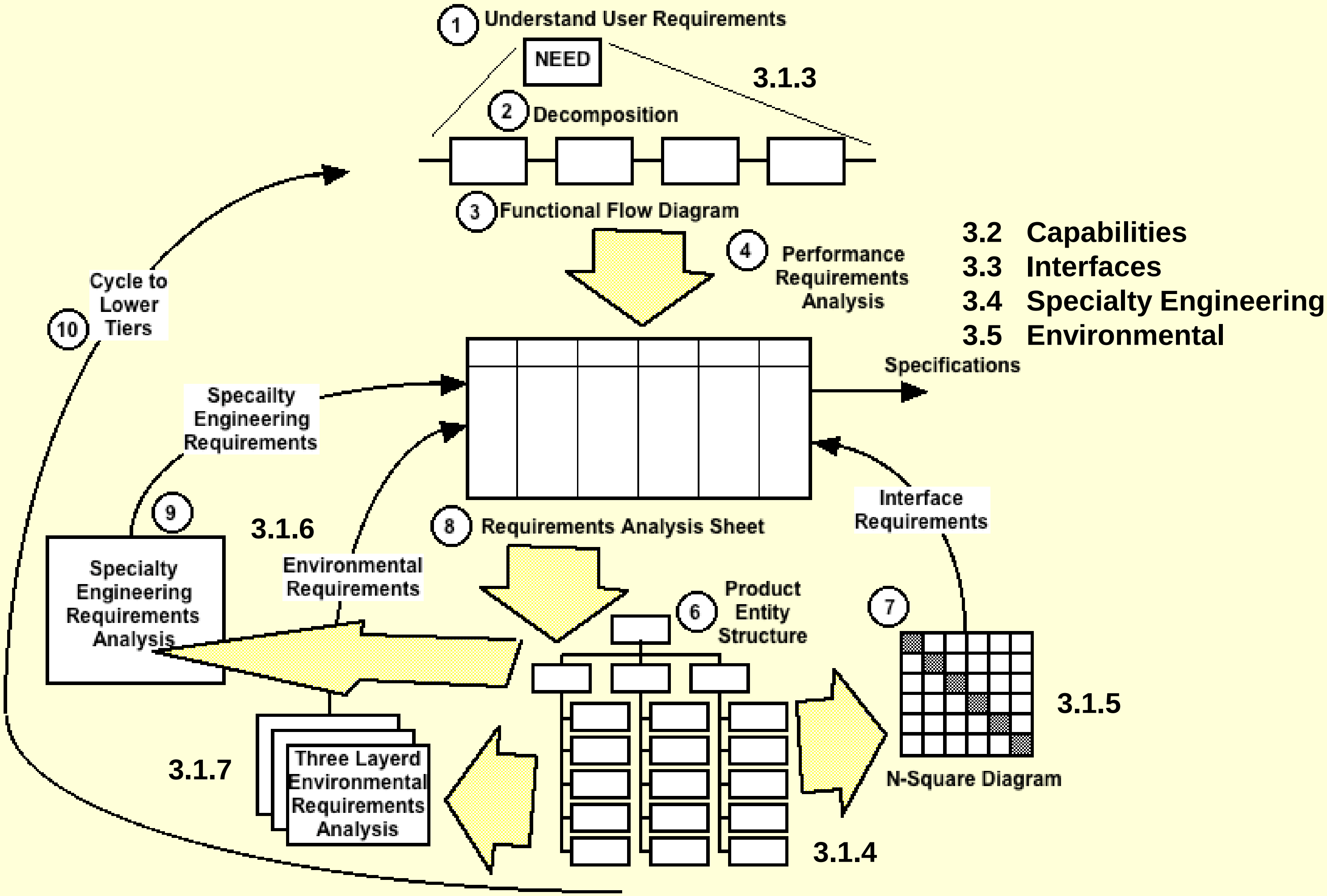
Three-Tier Environmental Requirements Construct



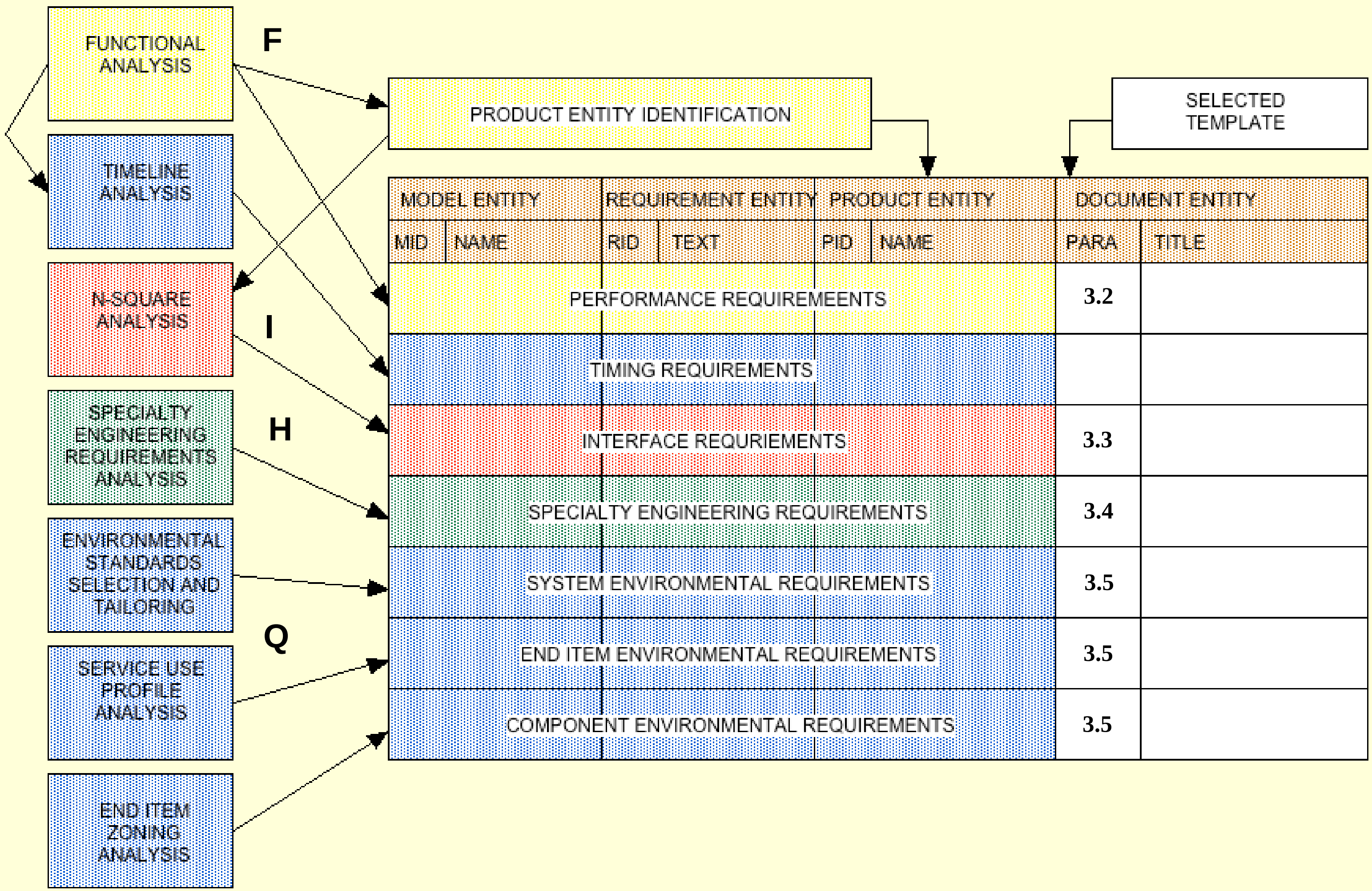
RAS Complete



Traditional Structured Analysis Overview



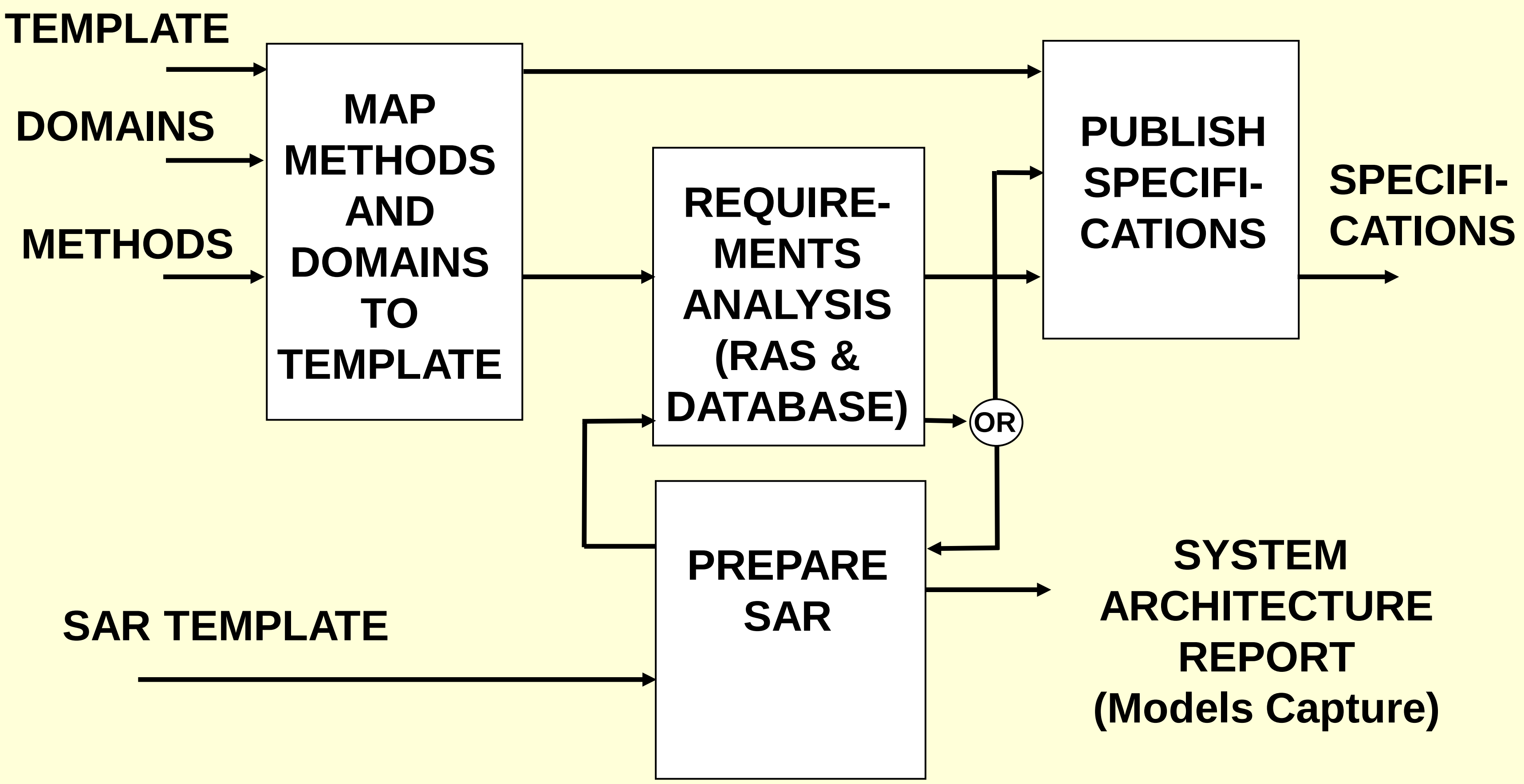
RAS-Complete, Fed From All Sources



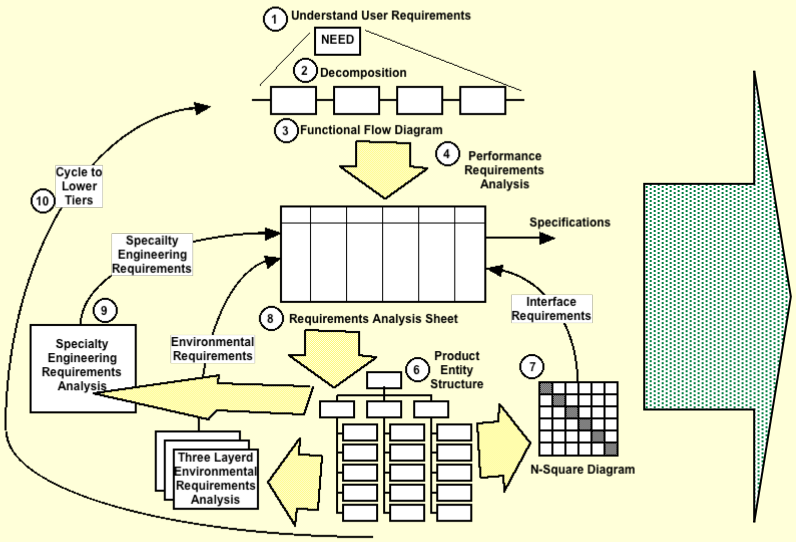
RAS-Complete In Table Form

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	XR67	Storage Volume < 10 ISO Vans	A1	Sensor Subsystem		
H	Specialty Engineering Disciplines			A	Product System		
H11	Reliability	EW34	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	FYH4	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	6GHU	Mean Time to Repair < 0.2 Hours	A1	Sensor Subsystem	3.1.6	Maintainability
H12	Maintainability	U9R4	Mean Time to Repair < 0.4 Hours	A11	Cable	3.1.6	Maintainability
H12	Maintainability	J897	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 Innerface			A1			
I181	Aggregate Signal Feed Source Impedance	E37H	Aggregate Signal Feed Source Impedance= 52 ohms + 2 ohms	A1	Sensor Subsystem		
I181	Aggregate Signal Feed Load Impedance	E37I	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	6D74	-40 degrees F< Temperature < +140 degrees F	A	Product System		
QX	Non-Cooperative Environmental Stresses			A	Product System		

Specification Generator and Modeling Work Product Capture

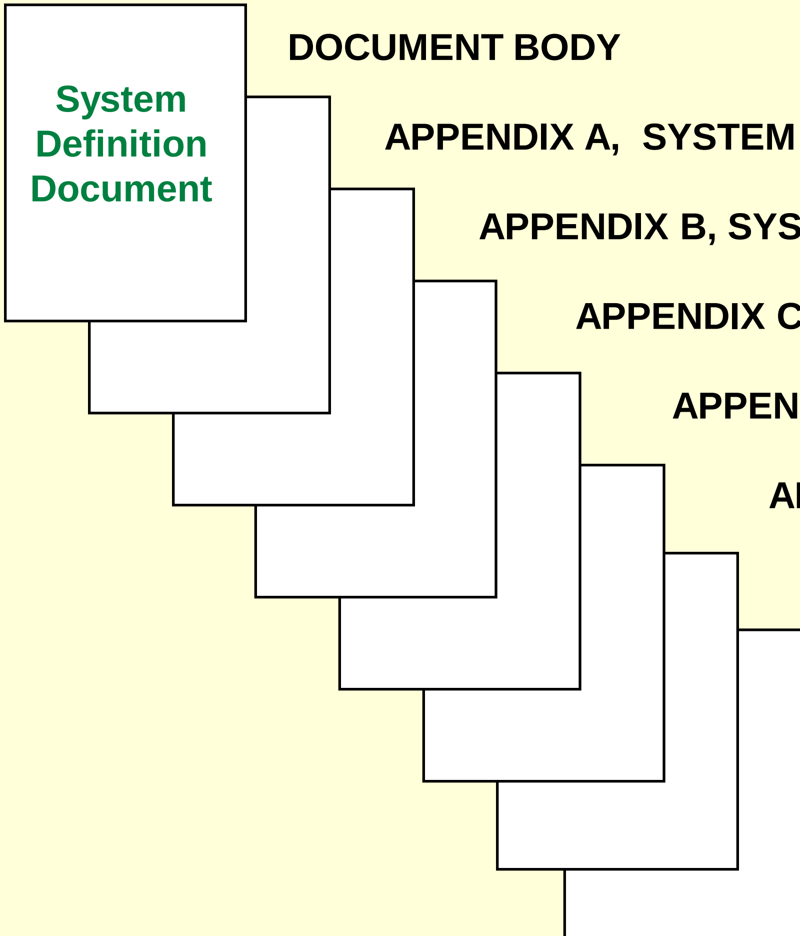


Derive Requirements From the TSA Model



3 REQUIREMENTS
3.1 Modeling
3.1.1 Need
3.1.2 User Requirements
3.1.3 Traditional Structured Modeling
3.1.4 Product Entity Modeling
3.1.5 Interface Modeling
3.1.6 Specialty Engineering Modeling
3.1.7 Environmental Classes and Modeling
3.2 Capabilities
3.3 Interface Requirements
3.4 Specialty Engineering Requirements
3.5 Environmental Requirements

SAR Organization



DOCUMENT BODY

APPENDIX A, SYSTEM FUNCTIONALITY

APPENDIX B, SYSTEM ENVINROMENT

APPENDIX C, PRODUCT ENTITY

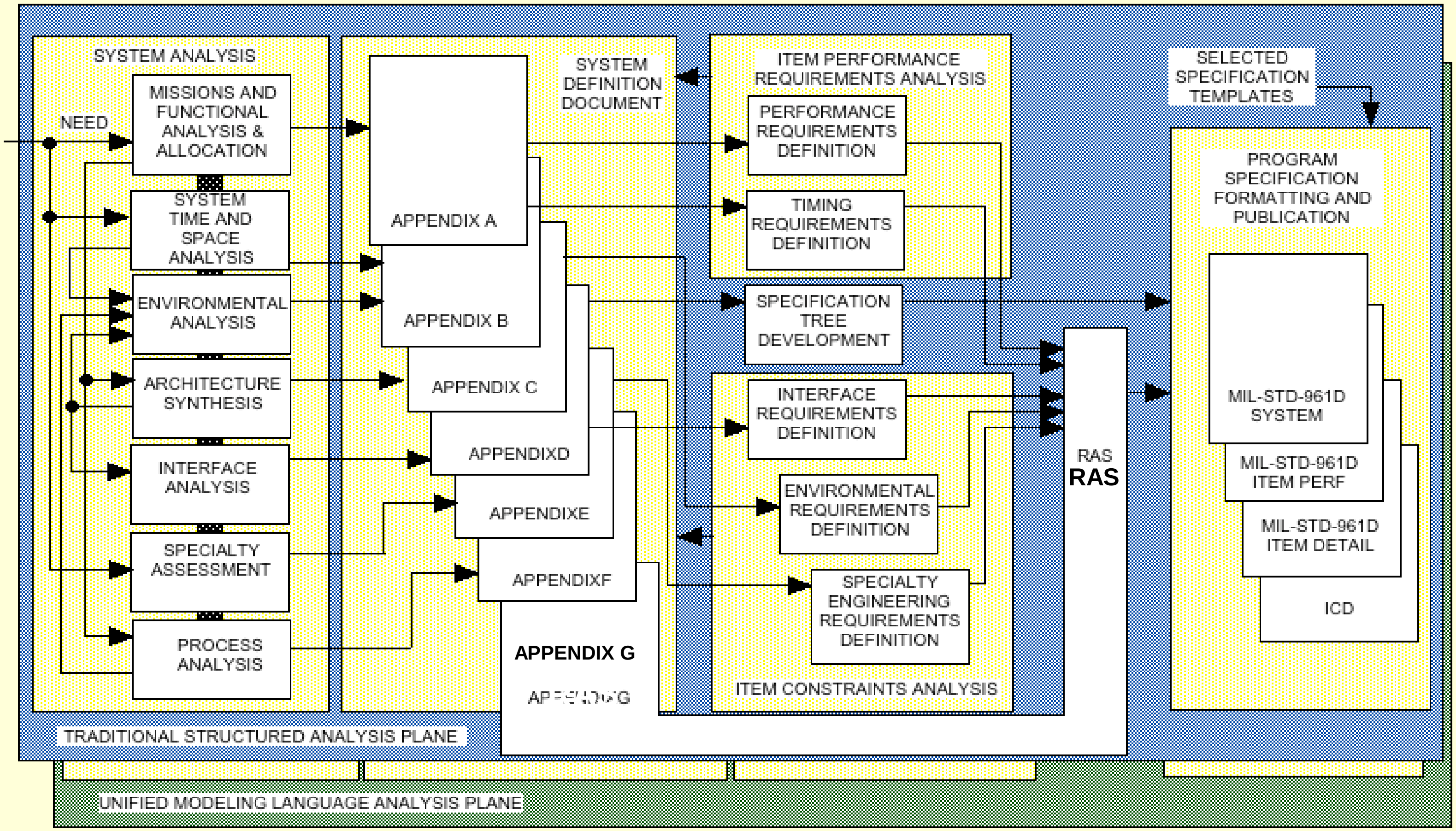
APPENDIX D, INTERFACE

APPENDIX E, SPECIALTY ENGINEERING

APPENDIX F, PROCESS

APPENDIX G, RAS

SAR Organization For Traditional Structured Analysis



Extending TSA to Software

- We will not spend a lot of time on this because most of us understand how flow charting was used in software development
- But a UADF can clearly be built reflecting how modeling was done in the 1950 and 1960s
- The blocks represented computer processing required and the directed line segments the sequence of processing.
- HIPO extended the diagram latterly to cover data flow.

JOG SYSTEM ENGINEERING
GRAND SYSTEMS DEVELOPMENT TRAINING PROGRAM
TUTORIAL

UNIVERSAL ARCHITECTURE DESCRIPTION FRAMEWORK

MSA/PSARE OVERVIEW AND UADF CONSTRUCT

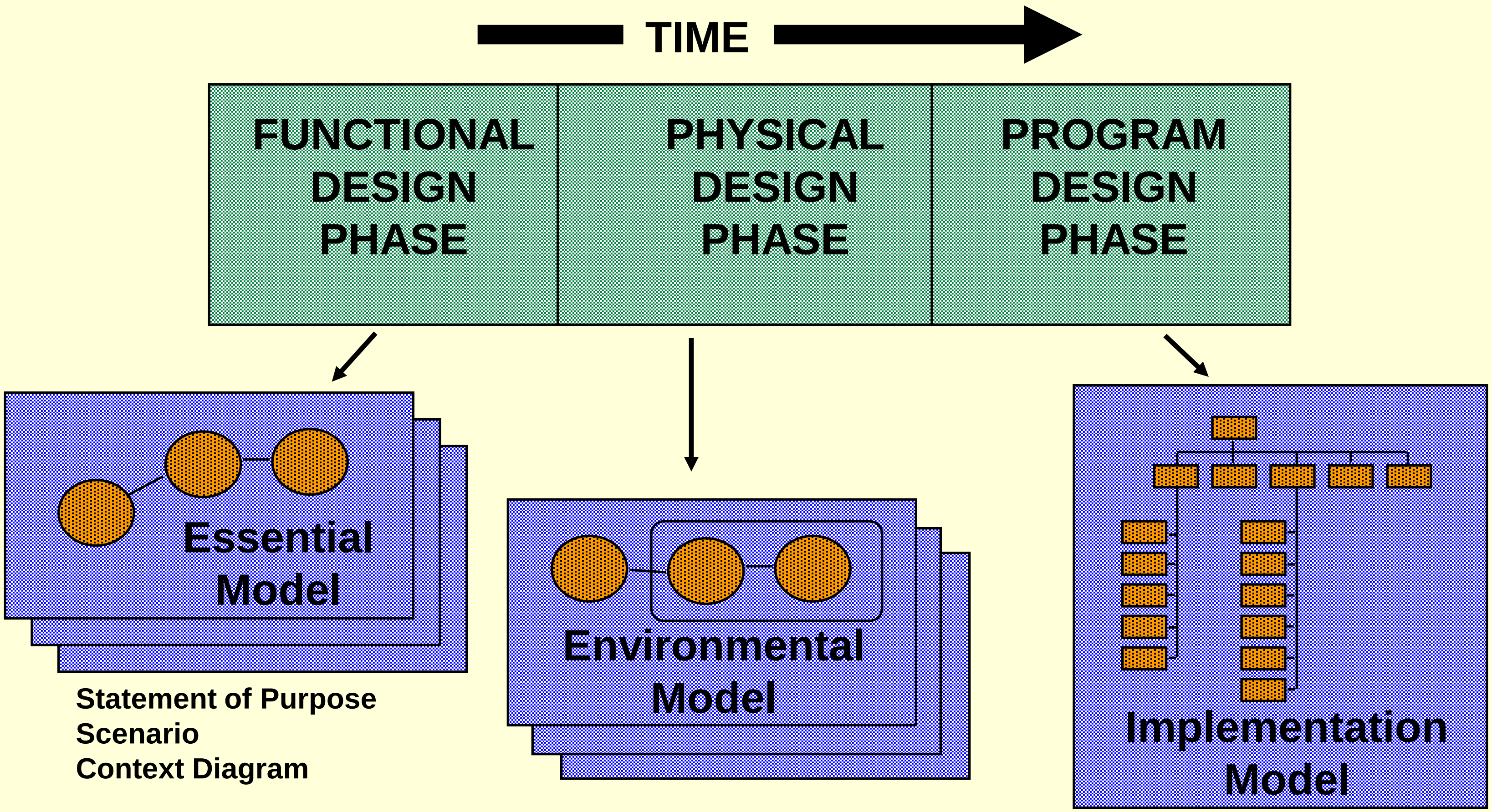
Period Goals

- Review the MSA approach derived from work by Yourdon, DeMarco, and others
- Extend the MSA model to the PSARE model that deals with control aspects of problem space and extends coverage to systems and hardware entities.
- Show how PSARE applies to hardware as well as software development

Computer Software Modeling Alternatives

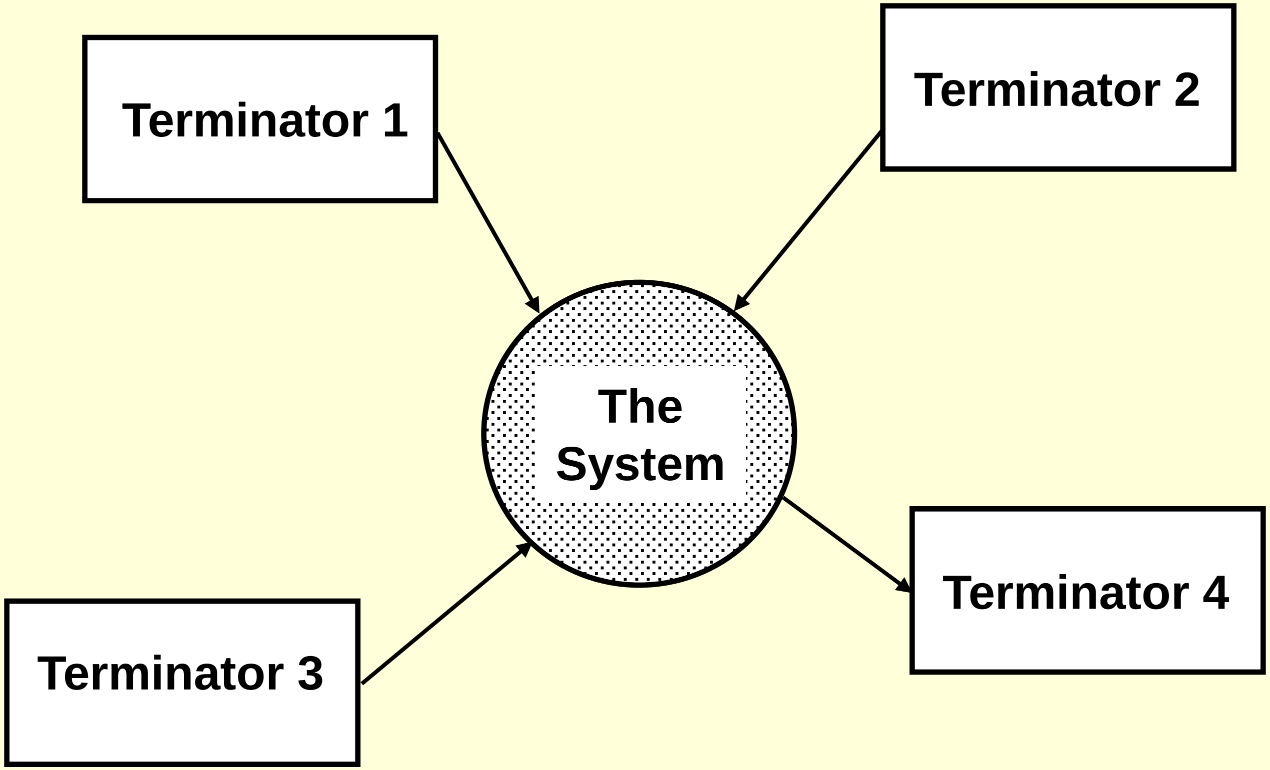
- **Process-oriented modeling**
  - Miscellaneous early methods
    - Flow charting
    - HIPO, IPO, behavioral diagrams
    - Structure diagram
  - Modern structured analysis
    - Yourdon-Demarco
    - Hatley-Pirbhai real time method now known as PSARE
- **Data oriented**
  - Table normalizing
  - IDEF-1X
  - DoDAF
- **Object oriented**
  - Early OOA
  - UML

Modern Structured Analysis
Modeling Sequence



Statement of Purpose
Scenario
Context Diagram

Modern Structured Analysis
Context Diagram - The Ultimate DFD



Modern Structured Analysis

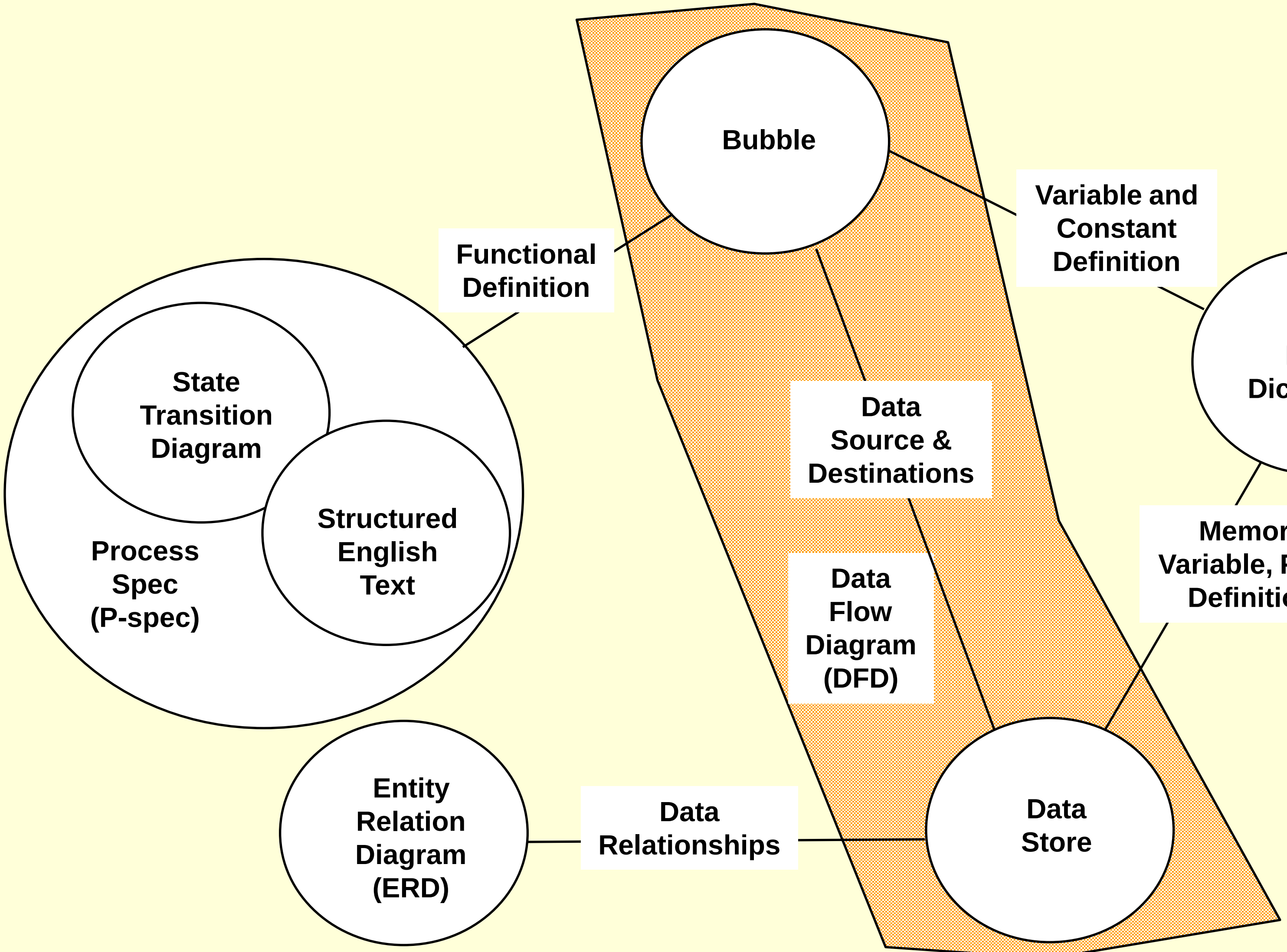
A Scenario in the Form of an Event List

Elevator System Event List	
1.	Passenger issues up summons request.
2.	Passenger issues down summons request.
3.	Elevator reaches summoned floor.
4.	Elevator not available for summons request.
5.	Elevator becomes available for summons.
6.	Passenger issues destination request.
7.	Elevator reaches requested destination.
8.	Elevator arrives at floor.
9.	Elevator departs floor.
10.	Elevator fails to move (goes out of service).
11.	Elevator returns to normal service.
12.	Elevator becomes overloaded.
13.	Elevator load becomes normal.

Phrase events from perspective of the environment

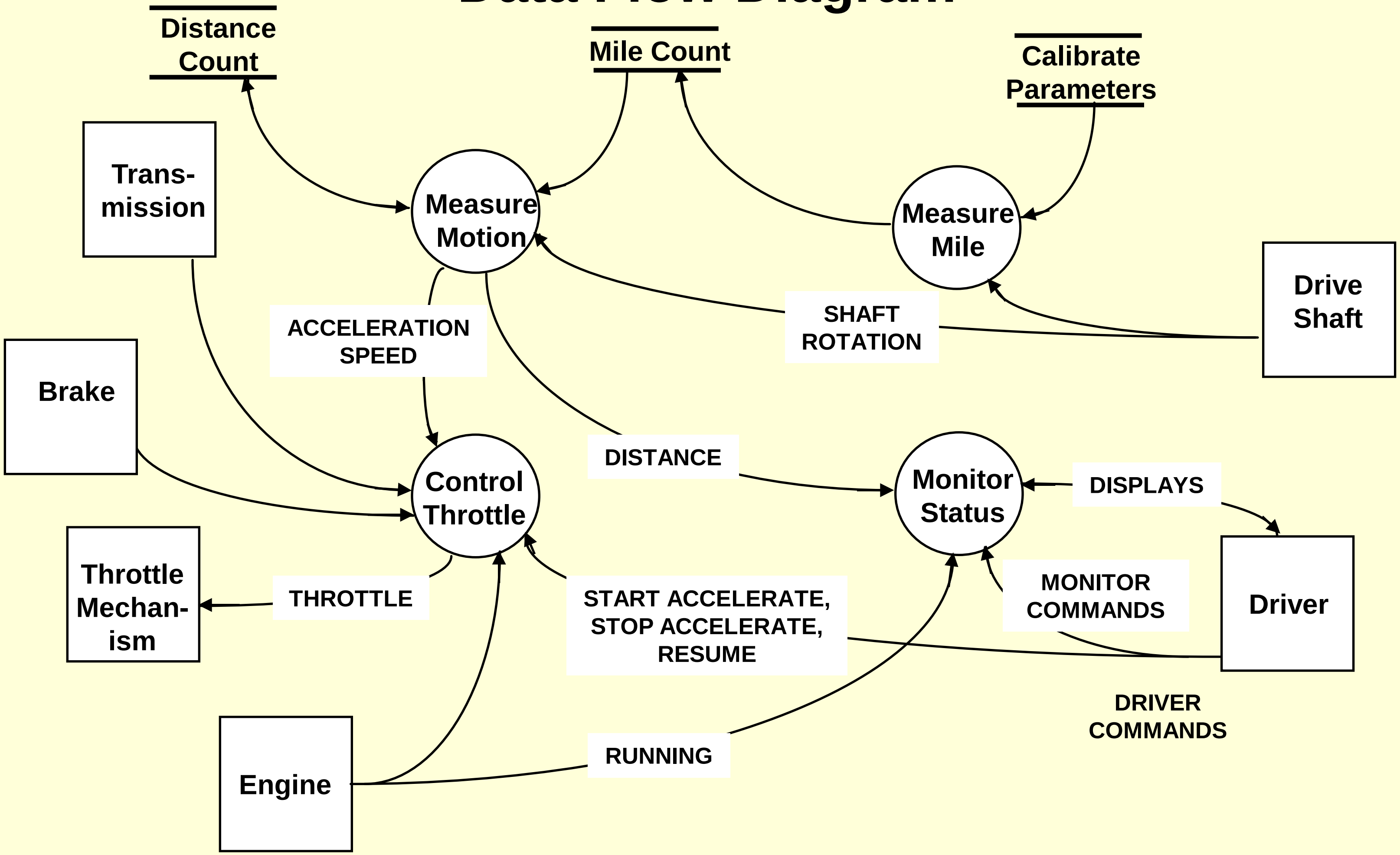
From Yourdon, "Modern Structured Analysis"

Modern Structured Analysis
Environmental Modeling Artifacts



Modern Structured Analysis

Data Flow Diagram



From Derek Hatley and Imitiaz Pirbhai, "Strategies For Real-Time System Specification", Dorset House, 1988

Modern Structured Analysis
Data Dictionary

- One data dictionary line item for each line or store on each DFD
- Name the data item and define it with mathematical precision

REQUIREMENTS (DATA) DICTIONARY

Name	Definition
ACCELERATION =	\Measured vehicle acceleration \Units: Miles per hour per second
ACTIVATE =	\Driver's cruse control activate command \2 Values: On, Off
DEFAULT COUNT =	\Constant = TBD; Default value of calibrated mile count \Units: Dimensionless

From Hatley, Pirbhai, "Strategies for Real-Time System Specification"

Modern Structured Analysis

Process Specification Examples

PSPEC 1.1 Measure Motion

For each pulse of SHAFT ROTATION

Add 1 to DISTANCE COUNT
then set:

DISTANCE = DISTANCE COUNT/MILE COUNT

At least once per second, measure pulse rate of SHAFT ROTATION in pulses per hour, and set:

SPEED = Pulse Rate/MILE COUNT

At least once per second, measure rate of change of SHAFT ROTATION pulses in pulses per hour, and set:

ACCELERATION = Rate of change/MILE COUNT

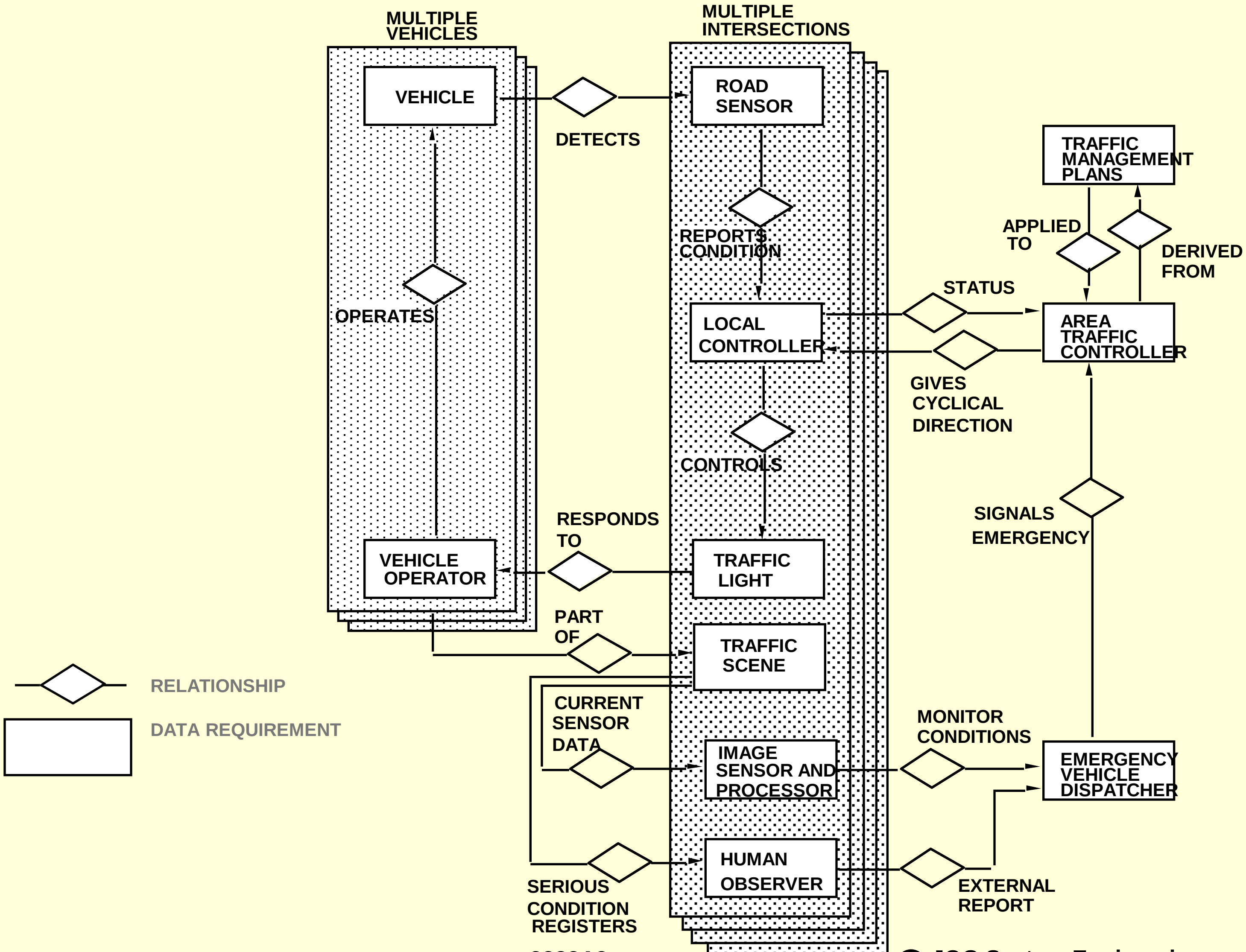
PSPEC 1.2 Measure Mile

Each time activated, start counting SHAFT ROTATION pulses

While LOWER LIMIT $\leq$ pulse count $\leq$ UPPER LIMIT
Set MILE COUNT = pulse count
Otherwise
Set MILE COUNT = DEFAULT COUNT

NOTE: All words in all-caps must be explained in dictionary.
From Hatley, Pirbhai, Strategies for Real-Time System Specification

Entity Relationship Diagram (ERD)

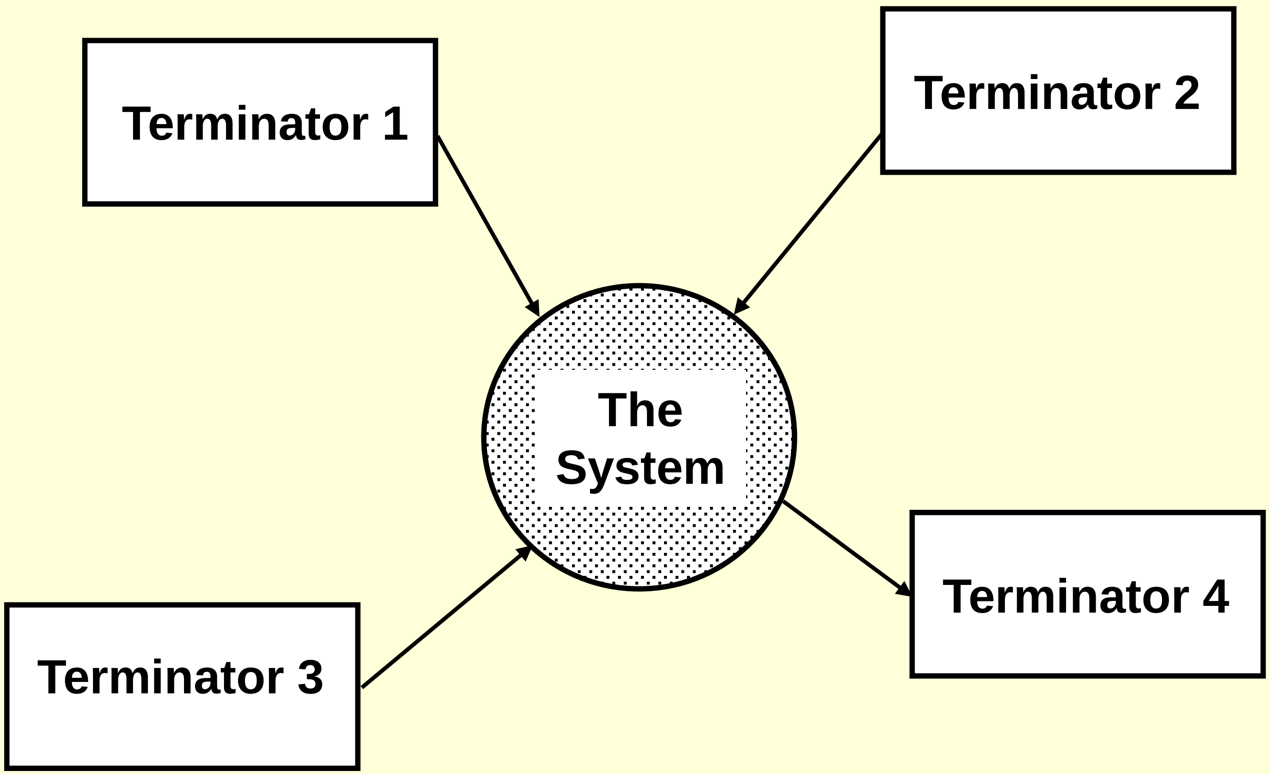


PSARE

- Extension for MSA to better deal with real time control
- Formerly known as Hatley Pirbhai
- PSARE = Process for System Architecture and Requirements Engineering
- Also extends MSA into systems work
- PSARE is closest to a universal architecture description framework of all existing models

PSARE

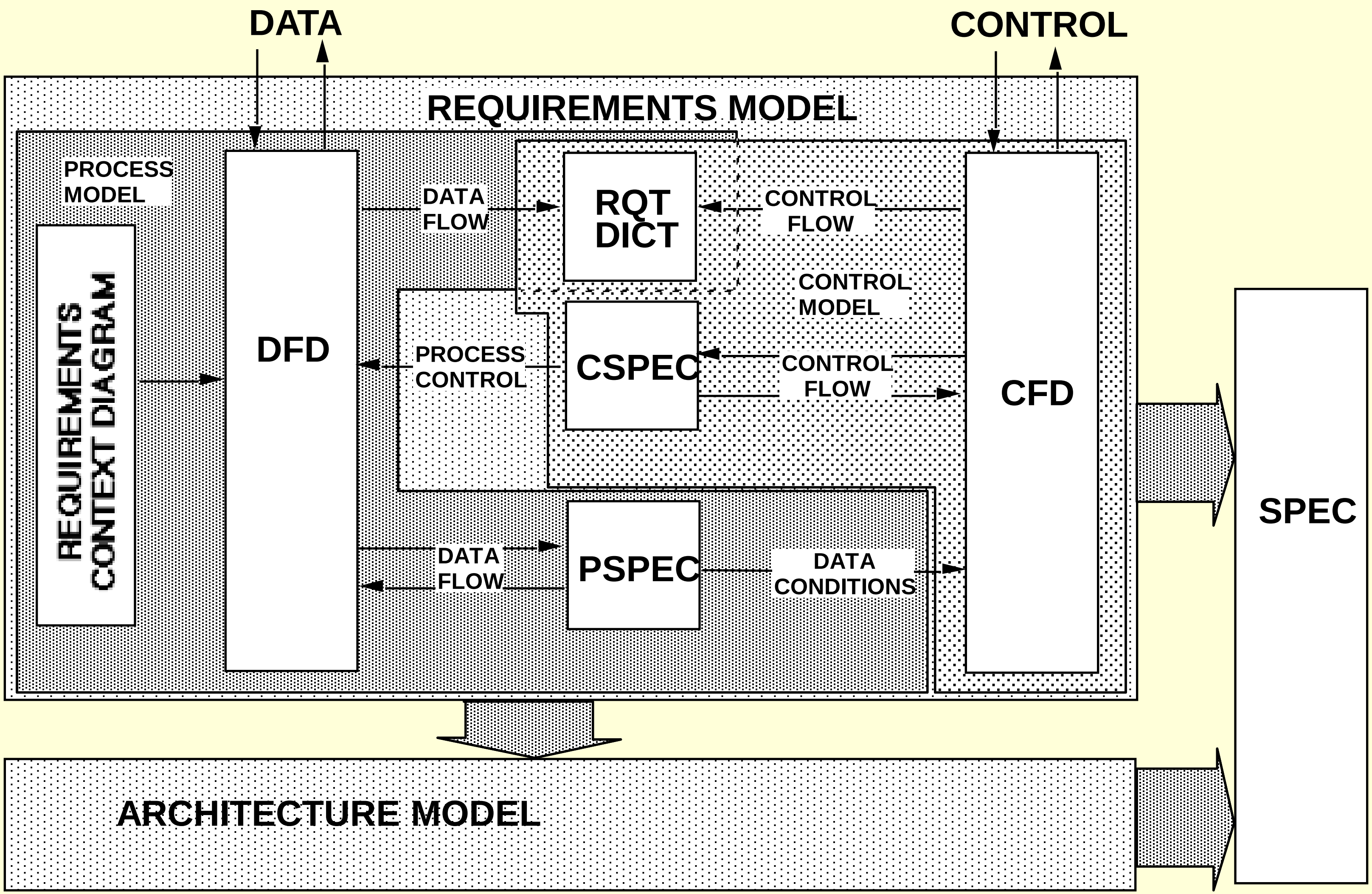
Also Includes the Context Diagram



PSARE Delta

- **DFDs illustrate information, energy, or material processing within the system.**
  - One DFD for each process at a particular level.
  - Lower tier DFDs expand on the information represented by the parent process.
  - As a whole, a set of DFDs represent requirements statements at increasing levels of detail.
  - Data stores retain information from a flow after the source ceases sending it.
  - DFD processes may deal with discrete or continuous data.
- **PSPECs define each process in terms of the relationship between input and output.**
  - Brief, concise narrative descriptions of the functions of a process at lowest level of decomposition.
  - Can contain, text, diagrams, or structured English.
  - Represents the requirements of the process and not the design.
  - They correlate with the process bubbles on the DFD at the lowest tier.
- **CFDs illustrates what the process must do under any given conditions.**
  - CFD bubbles mirror the DFD structure.
  - CFD flows can be shown on one plane using dashed lines
  - Data conditions are control signals generated within PSPECs through tests on data.
  - Process controls are generated from logic in CSPECs and activate/deactivate DFD processes.
- **CSPECs define behavior needed to control the system.**
  - Control behavior defined in terms of finite state machines (combinatorial or sequential).
  - Inputs are control flows from CFDs.
  - Outputs are process activators and control flows into CFDs.
- **REQUIREMENTS (or Data) DICTIONARY defines all data and control elements.**
  - One line item for each flow on every DFD and CFD.

PSARE Requirements Model



PSARE

Requirements (Data) Dictionary

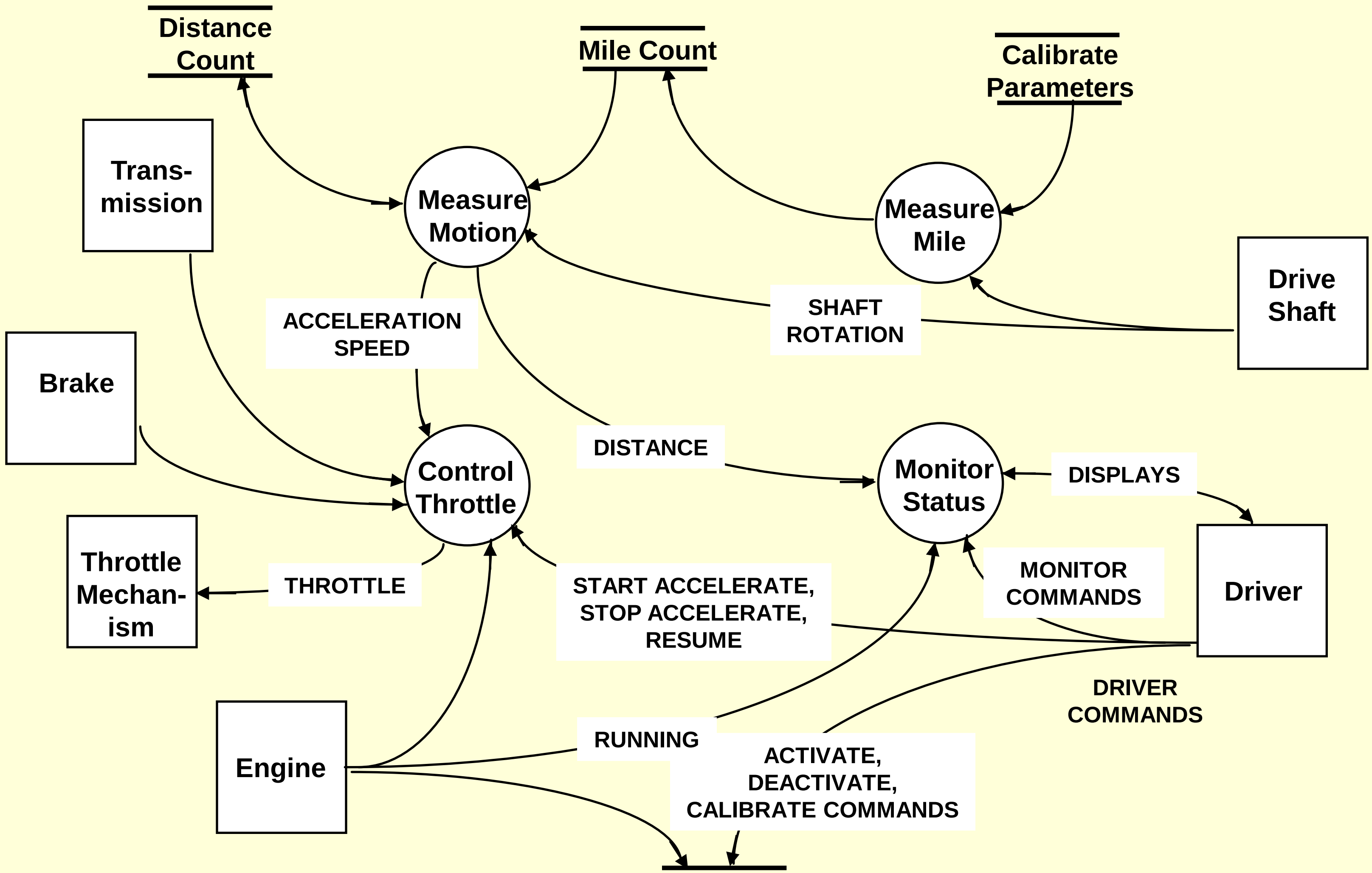
- One data dictionary line item for each line or store on each DFD
- Name the data item and precisely define it with mathematical precision

REQUIREMENTS (DATA) DICTIONARY

Name	Definition
ACCELERATION =	\Measured vehicle acceleration \Units: Miles per hour per second
ACTIVATE =	\Driver's cruse control activate command \2 Values: On, Off
DEFAULT COUNT =	\Constant = TBD; Default value of calibrated mile count \Units: Dimensionless

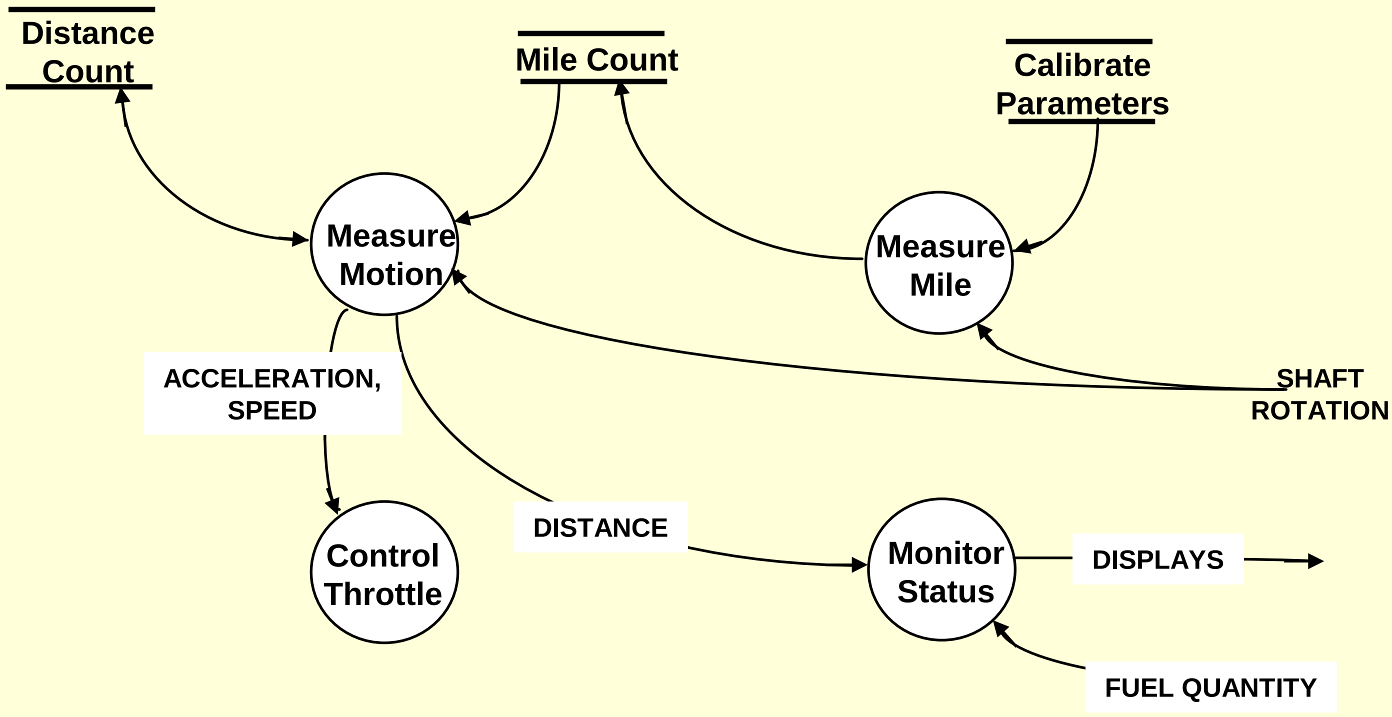
From Hatley, Pirbhai, "Strategies for Real-Time System Specification"

Combined DFD/CFD



From Derek Hatley and Imitiaz Pirbhai, "Strategies For Real-Time System Specification", Dorset House, 1988

PSARE
Isolated DFD



From Derek Hatley and Imitiaz Pirbhai, "Strategies For Real-Time System Specification, Dorset House", 1988

PSARE

Process Specification Examples

PSPEC 1.1 Measure Motion

For each pulse of SHAFT ROTATION

Add 1 to DISTANCE COUNT then set:

DISTANCE = DISTANCE COUNT/MILE COUNT

At least once per second, measure pulse rate of SHAFT ROTATION in pulses per hour, and set:

SPEED = Pulse Rate/MILE COUNT

At least once per second, measure rate of change of SHAFT ROTATION pulses in pulses per hour, and set:

ACCELERATION = Rate of change/MILE COUNT

PSPEC 1.2 Measure Mile

Each time activated, start counting SHAFT ROTATION pulses

While LOWER LIMIT $\leq$ pulse count $\leq$ UPPER LIMIT

Set MILE COUNT = pulse count

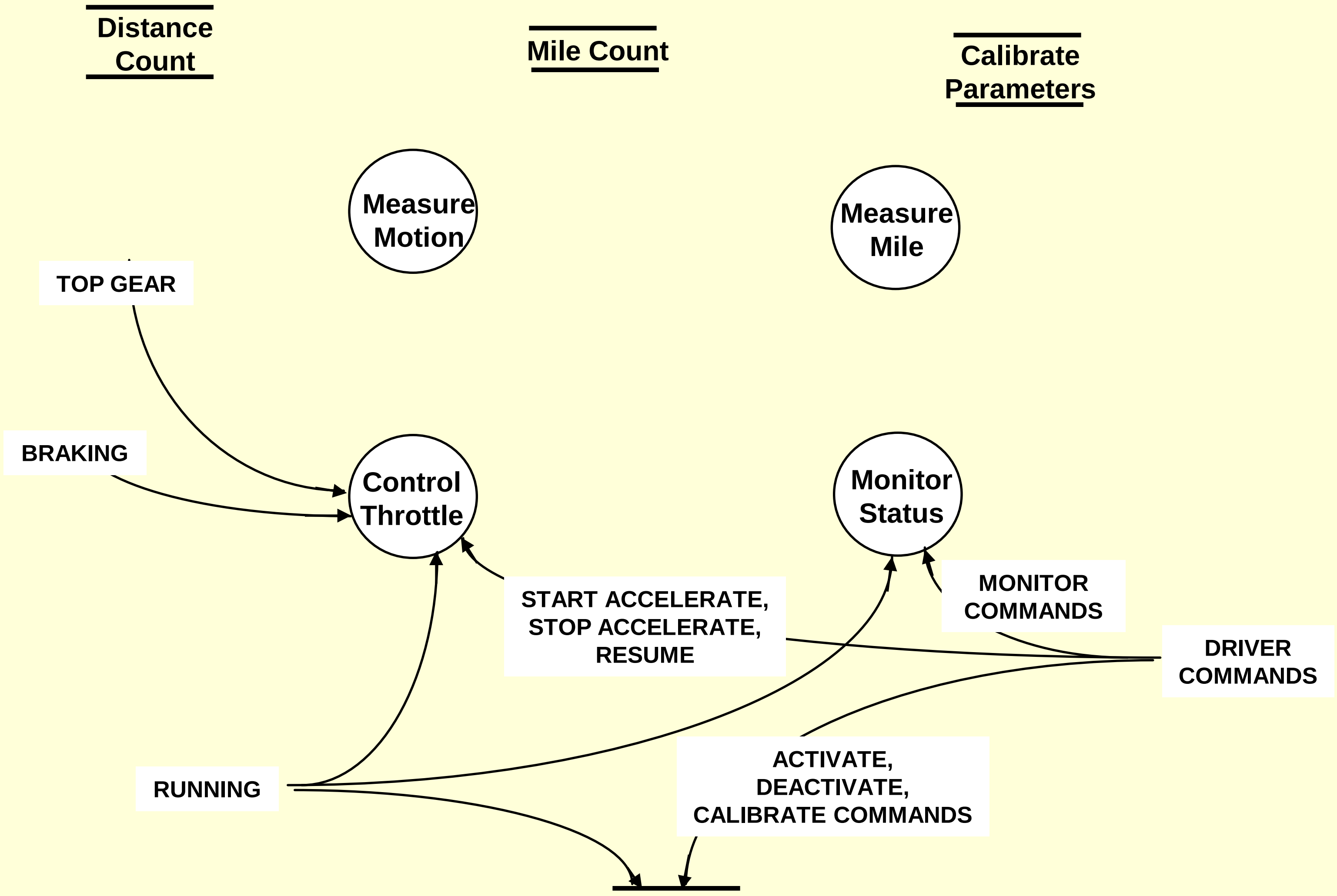
Otherwise

Set MILE COUNT = DEFAULT COUNT

NOTE: All words in all-caps must be explained in dictionary.

From Hatley, Pirbhai, "Strategies for Real-Time System Specification"

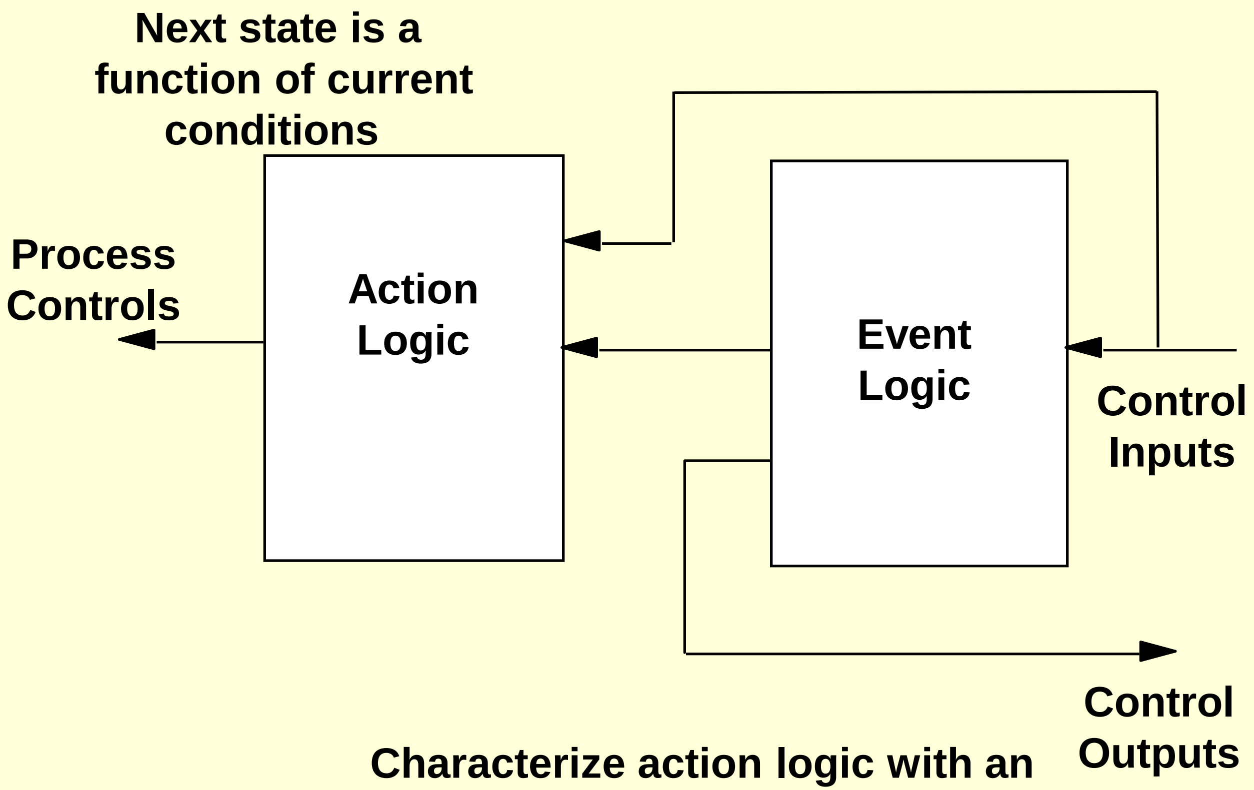
PSARE
Isolated CFD



From Derek Hatley and Imitiaz Pirbhai, "Strategies For Real-Time System Specification", Dorset House, 1988

PSARE

C-Specs - Generic Combinational Machine

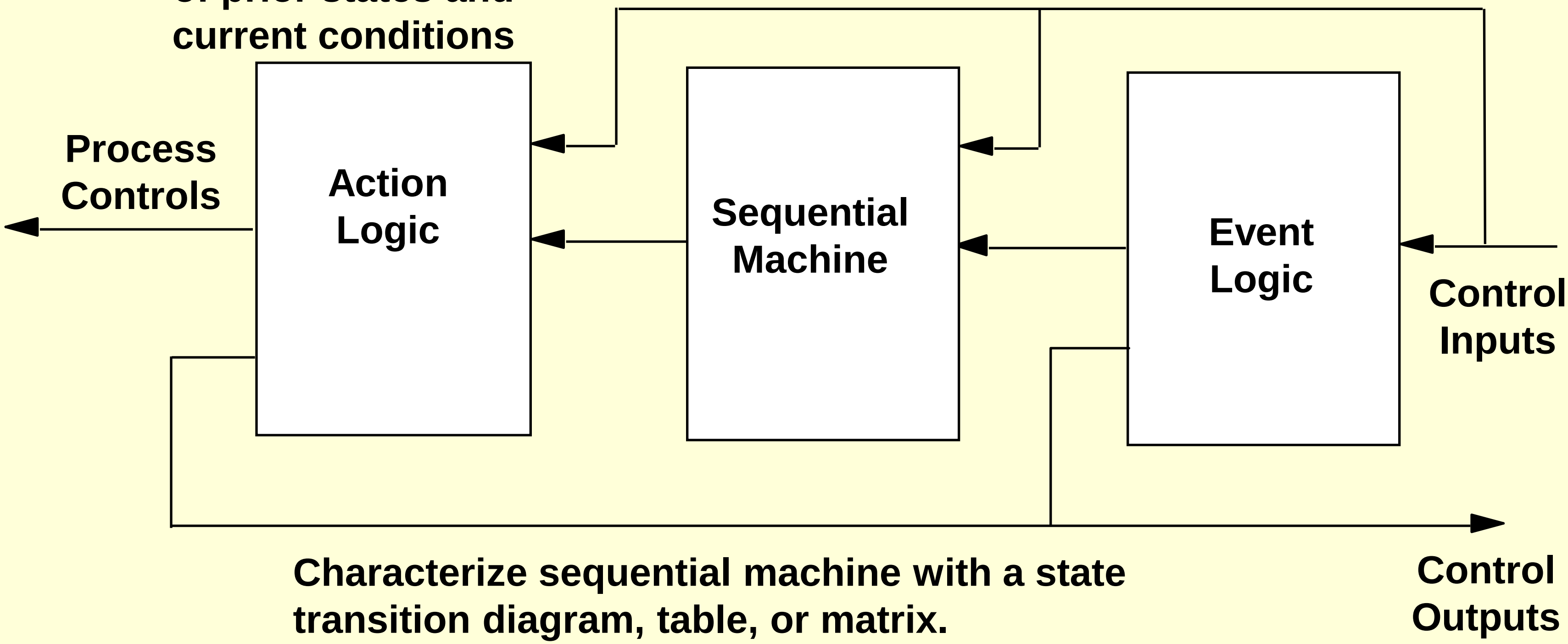


Characterize action logic with an activation table and the event logic with a decision table.

PSARE

CSpec - Generic Sequential Machine

Next state is a function
of prior states and
current conditions

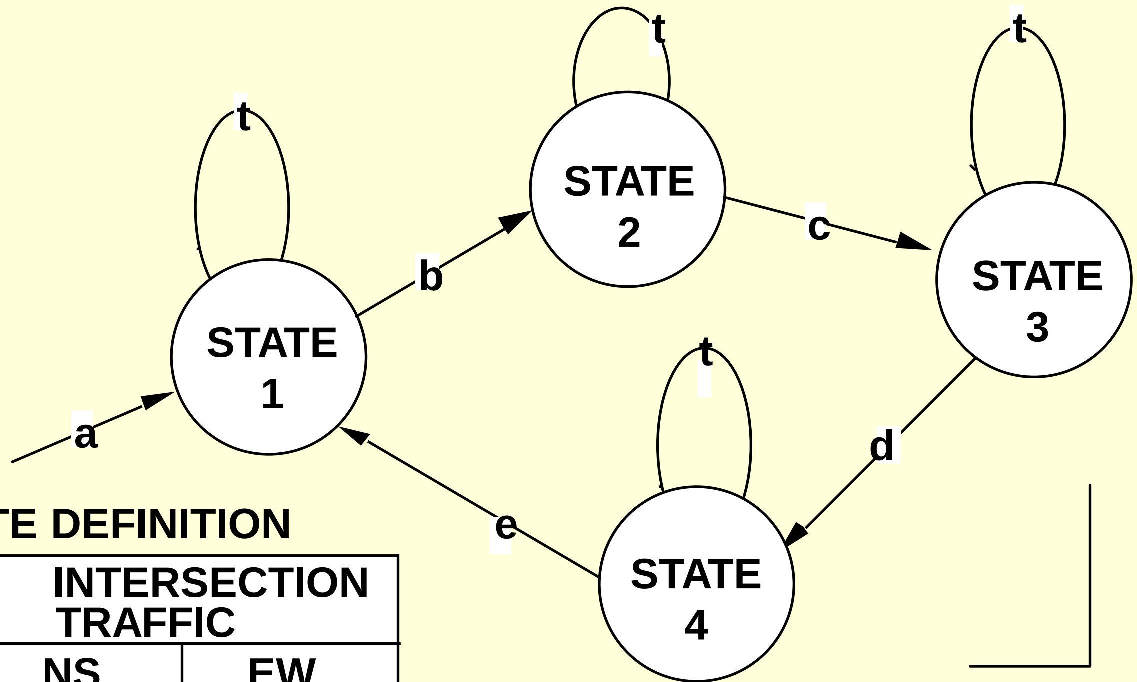


Characterize sequential machine with a state transition diagram, table, or matrix.

Characterize action logic with an activation table and the event logic with a decision table.

PSARE

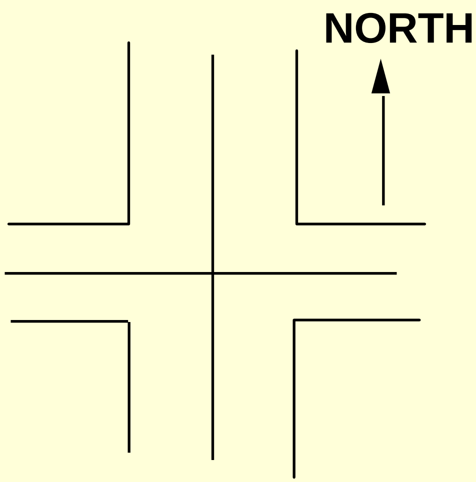
State Transition Diagram



STATE DEFINITION

STATE	INTERSECTION TRAFFIC NS	EW
1	GREEN	RED
2	ORANGE	RED
3	RED	GREEN
4	RED	ORANGE

TRANSITION RULES
a=Power On
b=State 1 AND 120t
c=State 2 AND 10t
d=State 3 AND 60t
e=State 4 AND 10t
t=timing pulse @ 60Hz



REAL WORLD SITUATION

PSARE

Decision Table

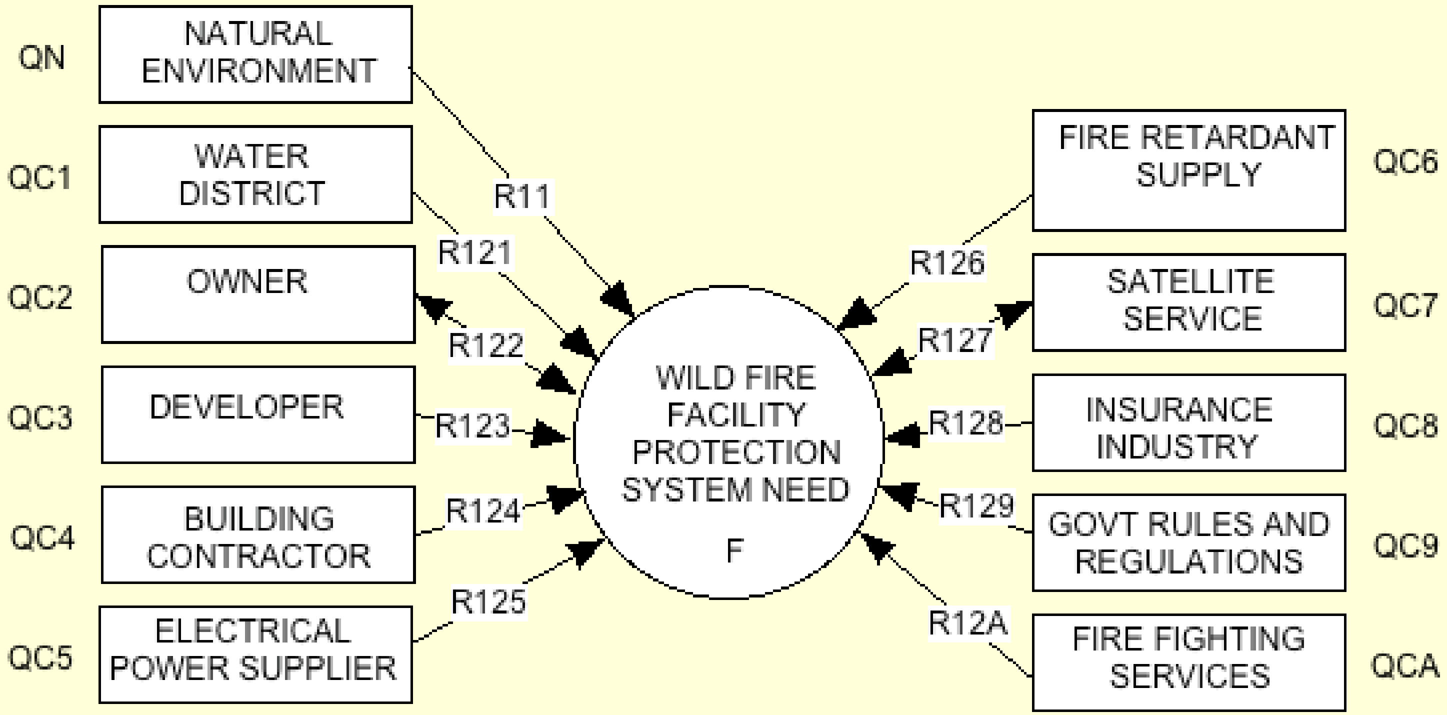
- Define inputs
- Determine all possible values of each input
- Determine output results desired for each input condition
- Review for impossible input combinations
- Review for combinations that can be grouped

MODE	INPUTS: TEMP	INPUTS: PRESS	OUTPUTS: HEATER	OUTPUTS: PUMP
IDLE	HIGH	HIGH	OFF	OFF
		LOW	OFF	OFF
	LOW	HIGH	OFF	OFF
		LOW	OFF	OFF
AUTO 1	HIGH	HIGH	OFF	OFF
		LOW	OFF	ON
	LOW	HIGH	ON	OFF
		LOW	ON	ON
AUTO 2	HIGH	HIGH	OFF	OFF
		LOW	OFF	ON
	LOW	HIGH	ON	OFF
		LOW	ON	OFF

ONLY FOUR POSSIBLE OUTPUTS:
Both Pump And Heater Off
Both Heater And Pump On
Heater On, Pump Off (May Be Dangerous)
Heater Off, Pump On

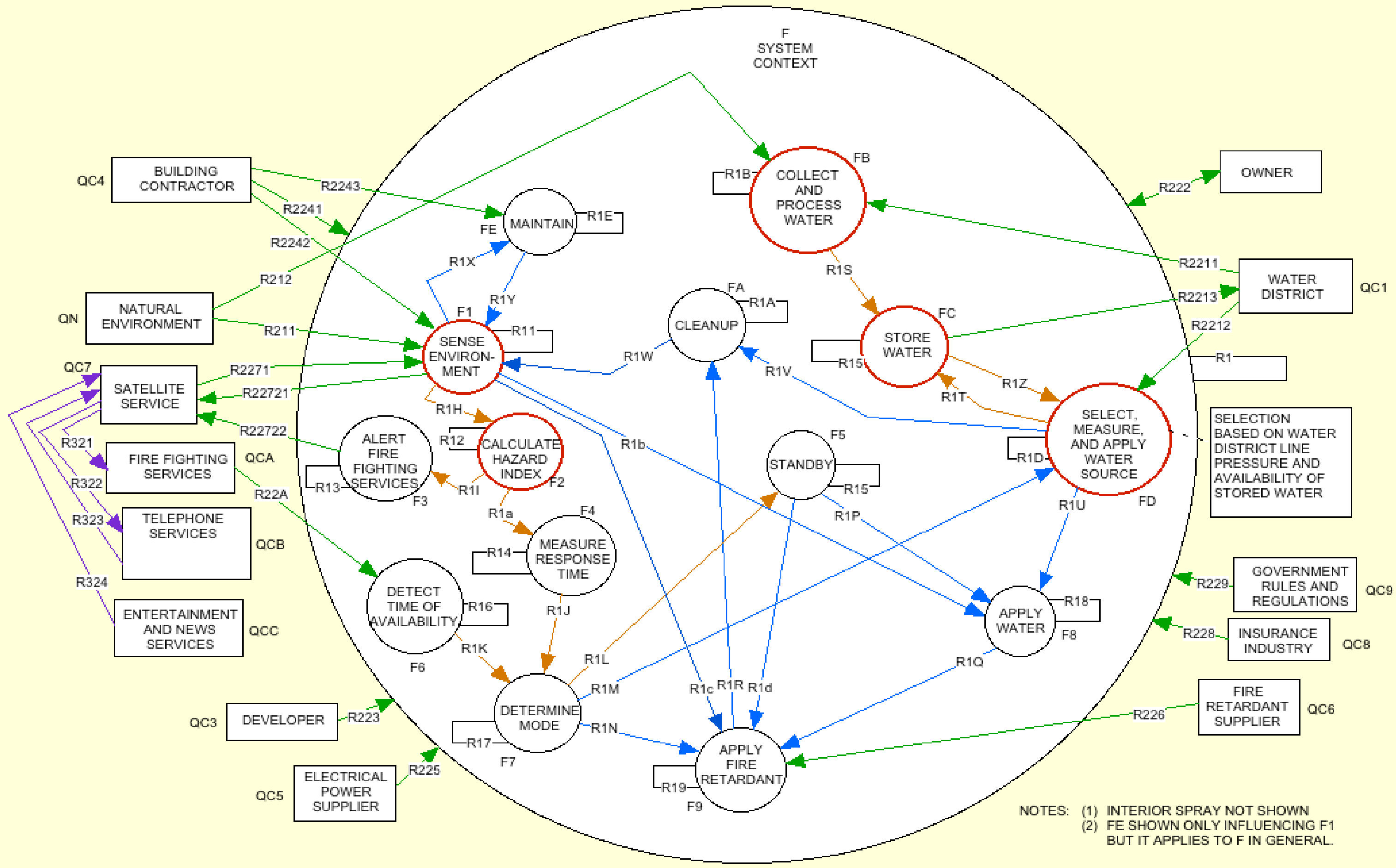
PSARE

Sample System Analysis - The Context Diagram



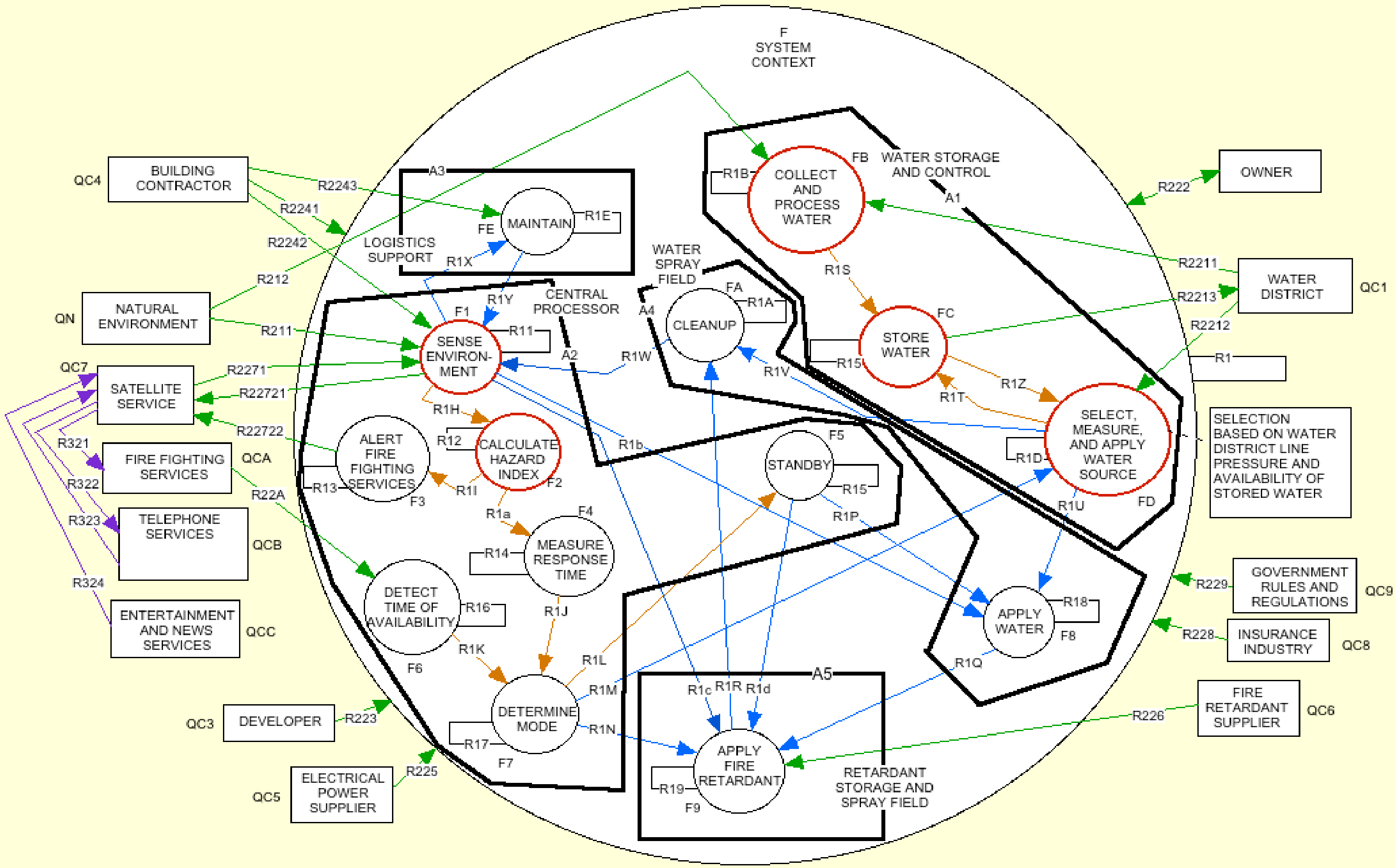
PSARE

Sample System Analysis – Context Diagram Expansion



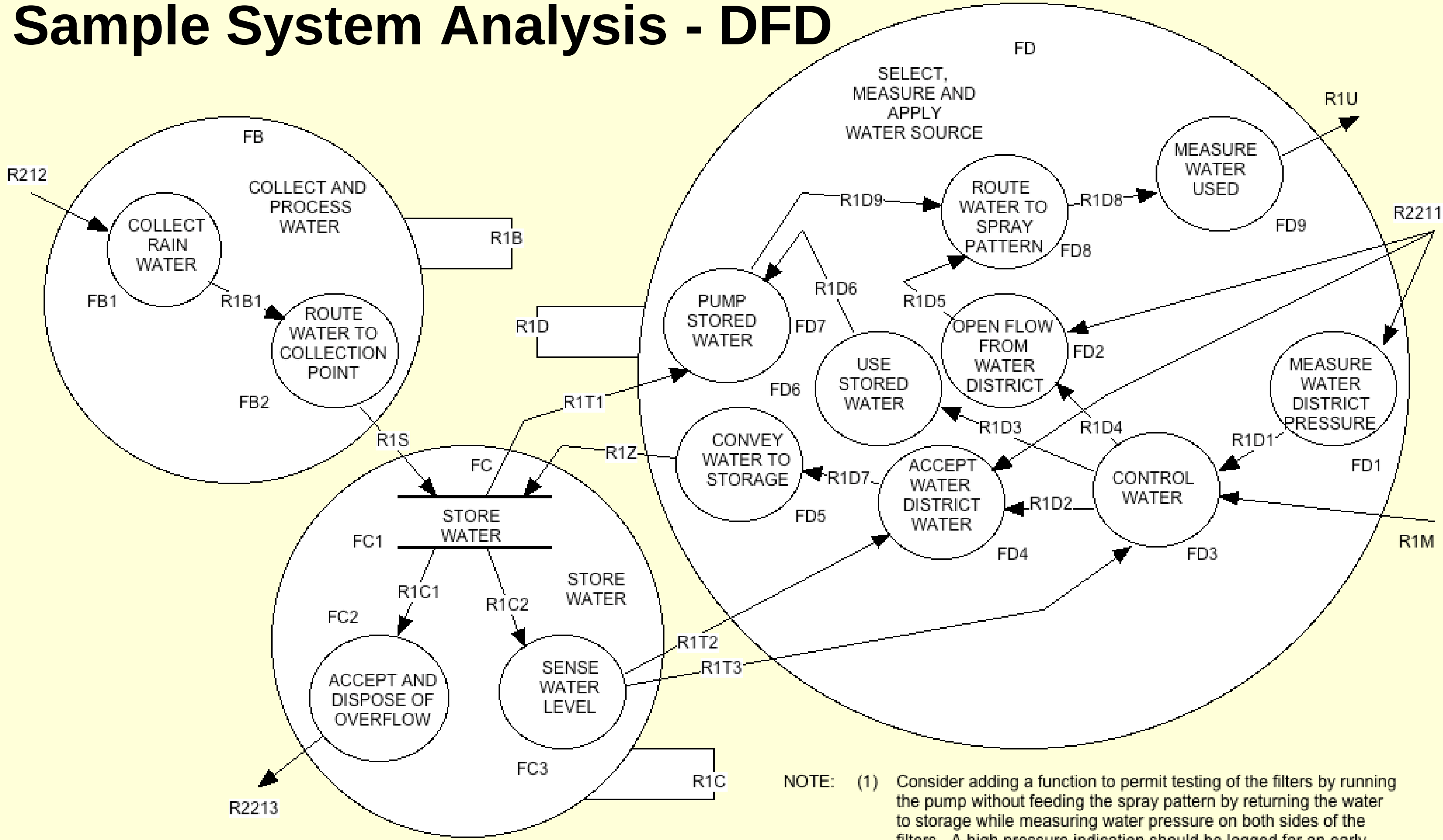
PSARE

Sample System Analysis - Super Bubbles



PSARE

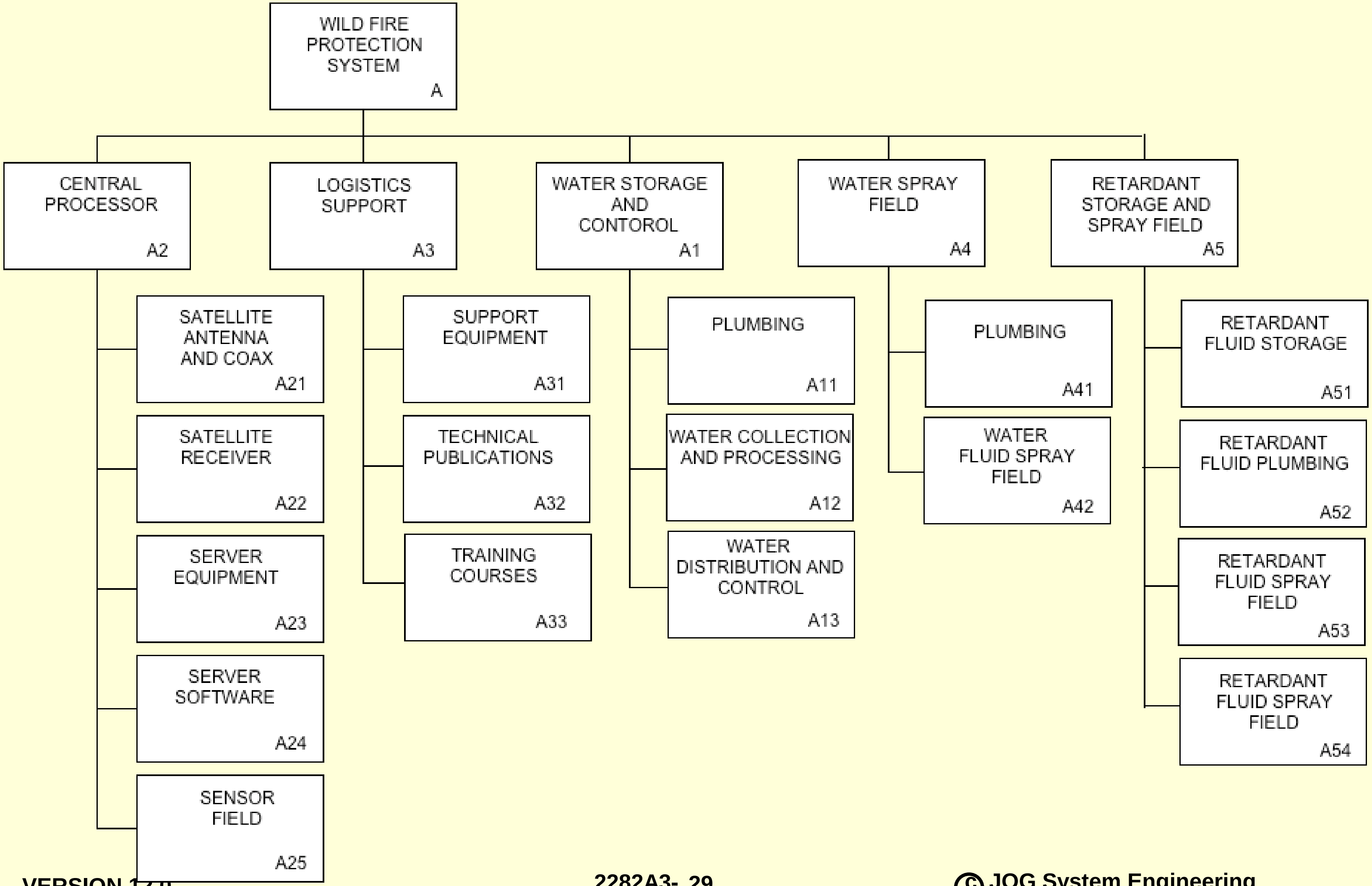
Sample System Analysis - DFD



NOTE:
(1) Consider adding a function to permit testing of the filters by running the pump without feeding the spray pattern by returning the water to storage while measuring water pressure on both sides of the filters. A high pressure indication should be logged for an early maintenance response.
(2) Consider adding a function and relationship that directs the water flow from function FD8 onto the facility from the direction of the fire rather than from all directions.
(3) Water used must be measured before the spray pattern rather than after.

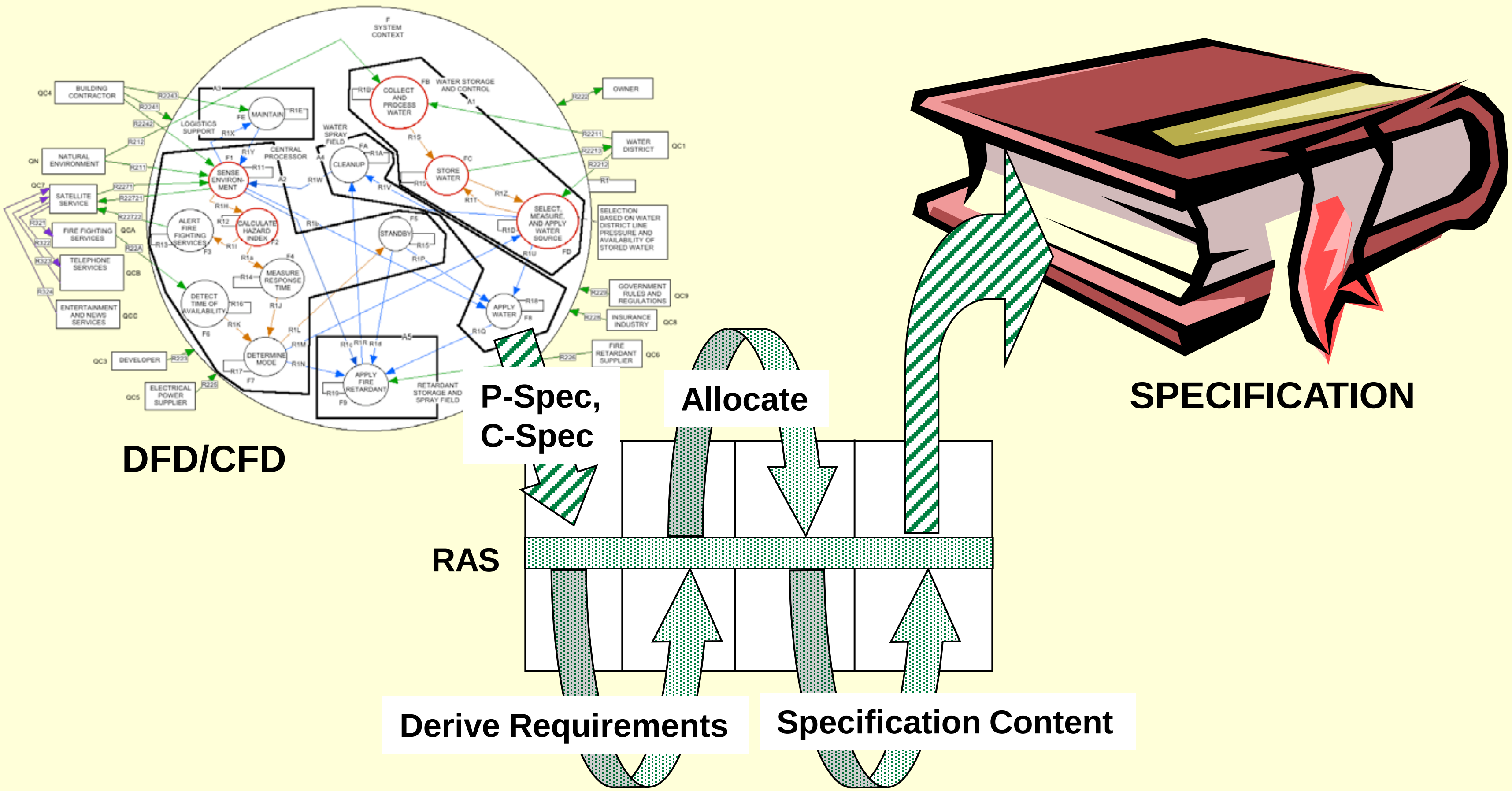
PSARE

Common Product Entity Structure



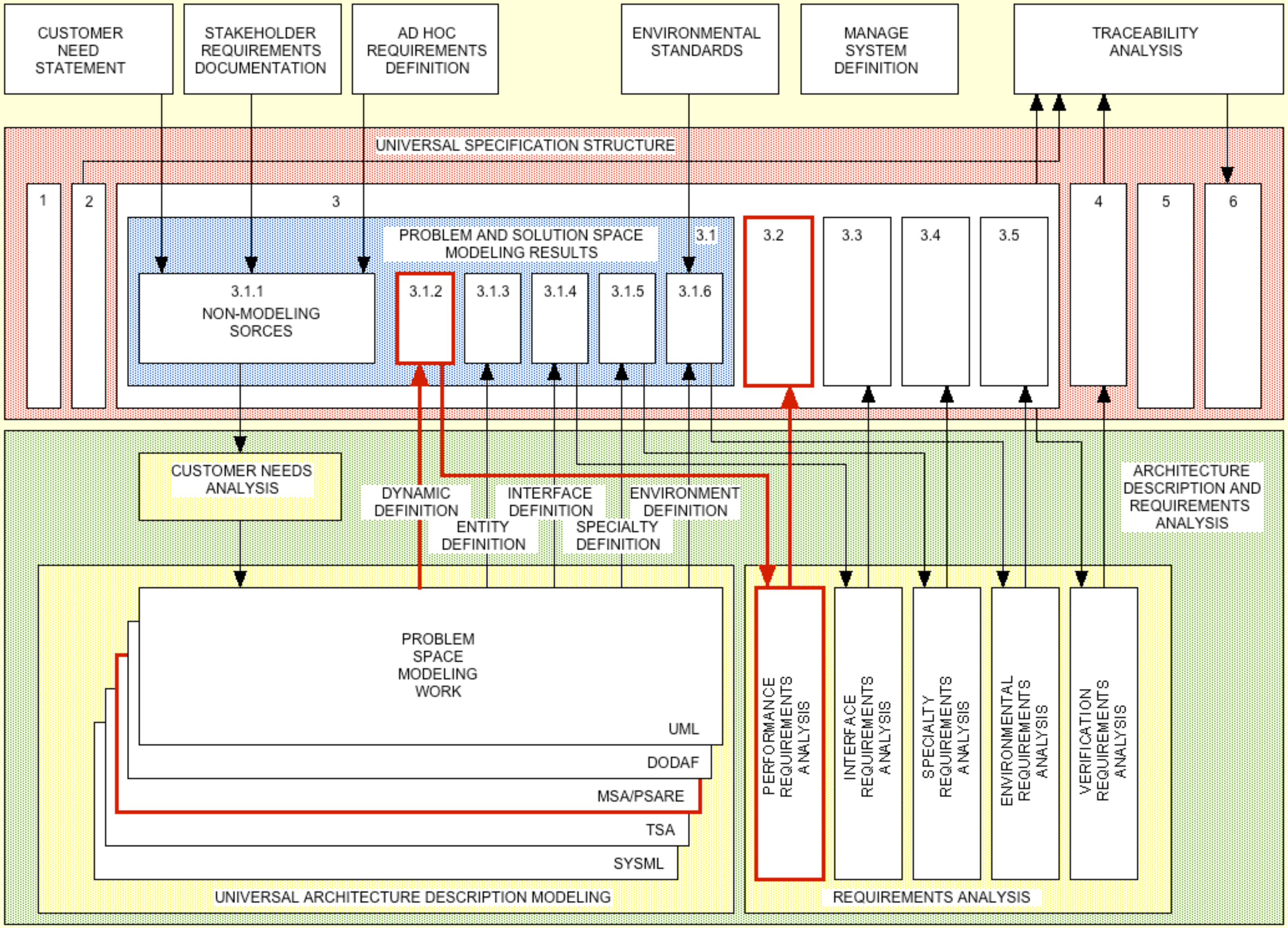
PSARE

The Requirements Dictionary is the RAS



PSARE

Deriving Specification Content



Continue Reading About MSA and PSARE

- Tom DeMarco, "Structured Analysis and System Specification",1979
- Edward Yourdon and Larry Constantine, "Structured Design", Prentice hall, 1979
- Victor Weinberg, "Structured Analysis", Yourdon Press, 1980
- Edward Yourdon, "Modern Structured Analysis", Yourdon Press, 1989
- Derek Hatley, Peter Hruschka, and Imtiaz Pirbhai, Process For System Architecture and Requirements Engineering", Dorset House, 2000

JOG SYSTEM ENGINEERING
GRAND SYSTEMS DEVELOPMENT TRAINING PROGRAM
TUTORIAL

UNIVERSAL ARCHITECTURE DESCRIPTION FRAMEWORK

UML-SysML OVERVIEW AND UADF CONSTRUCT

UML-SysML UADF Components

- **Hardware Models**
  - **Traditional Structured Analysis**
  - **SysML**
- **Computer Software Models**
  - **Process-oriented analysis**
    - **Flow charting**
    - **Modern Structured Analysis**
    - **PSARE**
  - **Data-oriented analysis**
    - **Table normalizing**
    - **IDEF-1X**
  - **Object-oriented analysis**
    - **Early models**
    - **UML**
  - **DoD architecture framework**

Benefits of SysML/UML UADF

- Top-Down
- Respect for Sullivan (dynamic opening)
- Seamless HW/SW switch
- Thoroughly modern
- Well supported by tool companies

UML and TSA

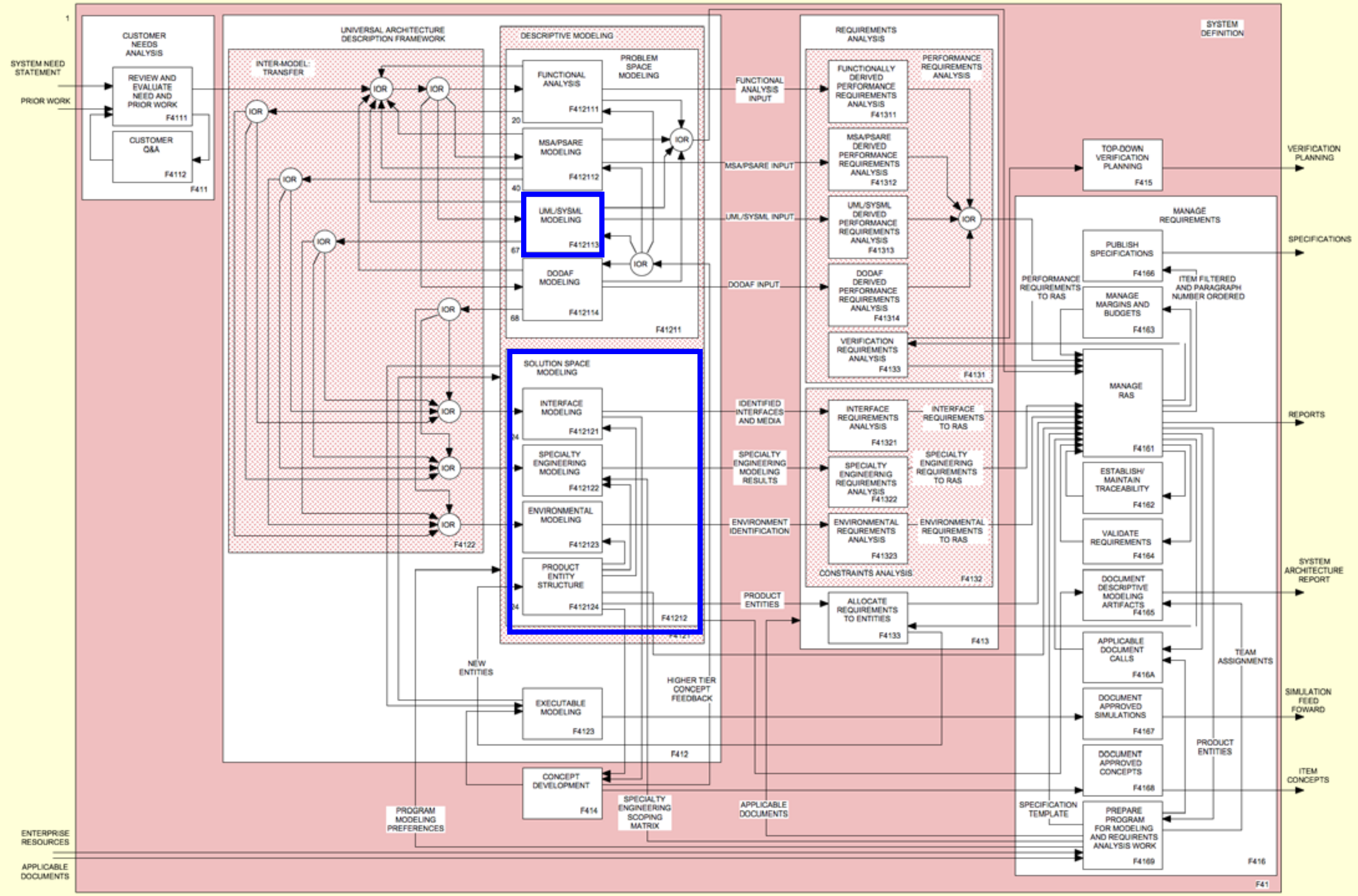
Not Too Far Apart Actually

UNIFIED MODELING LANGUAGE (UML)

STATIC REPRESENTATION			DYNAMIC REPRESENTATION				
					INTERACTION DIAGRAMS		
DEPLOY-MENT DIAGRAM	COMPONENT DIAGRAM	OBJECT & CLASS DIAGRAMS	STATE CHART	USE CASE DIAGRAM	COMMUNI-CATION DIAGRAM	SEQUENCE DIAGRAM	ACTIVITY DIAGRAM
PRODUCT ENTITY BLOCK DIAGRAM			STATE DIAGRAM	SCHEMATIC BLOCK DIAGRAM		TIMELINE DIAGRAM	FUNCTIONAL FLOW DIAGRAM
PHYSICAL FACET			BEHAVIORAL FACET				FUNCTION-AL FACET

TRADITIONAL STRUCTURED ANALYSIS (TSA)

SysML/UML UADF With Common Solution Space Models



The Diagrams of UML 2.0

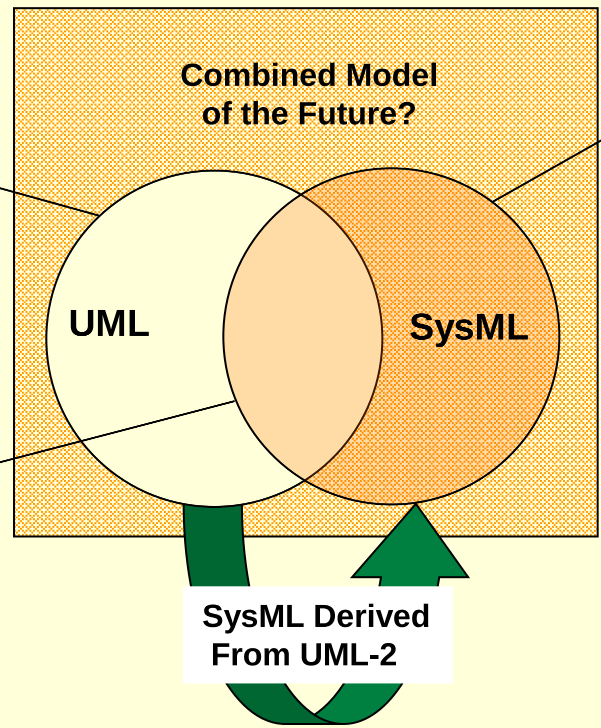
- For modeling dynamic parts of the system
  - ✓ Use case diagram
  - ✓ Sequence diagram
  - – Timing diagram
  - ✓ Communication diagram (renamed in 2)
  - ✓ State diagram
  - ✓ Activity diagram
  - – Interaction overview diagram (2)
- For modeling static parts of the system
  - ✓ Object and class diagrams
  - ✓ Component diagram
  - ✓ Deployment diagram
  - – Composite structure diagram (2)
  - – Package diagram (2)

(2) = added in UML 2.0
✓ Diagrams discussed in tutorial

The Diagrams in UML and SysML

UML-2 Only

Object/Class Diagram
Component Diagram
Deployment Diagram
Communication Diagram
Interaction Overview Diagram
Composite Structure Diagram



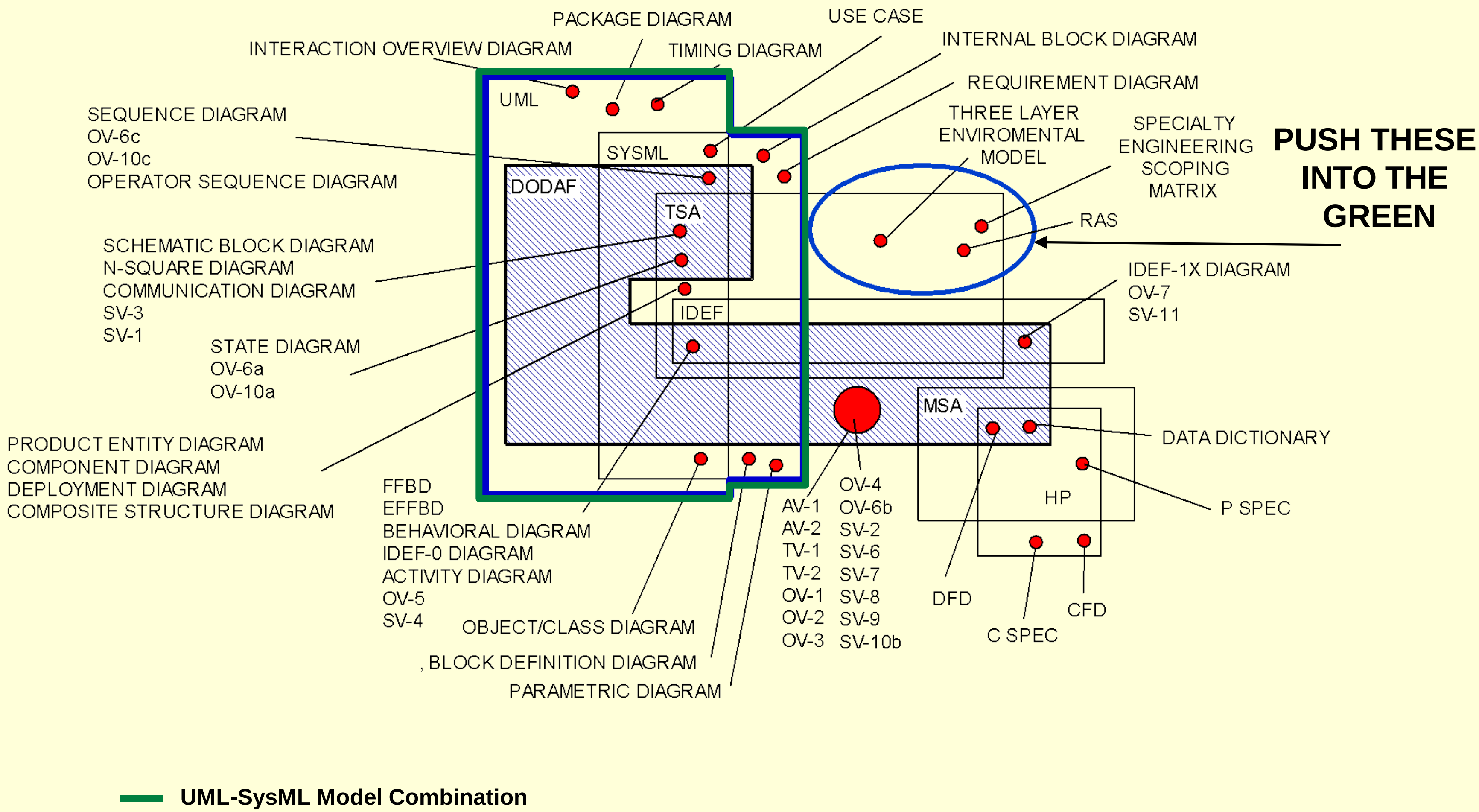
SysML Only

Requirement Diagram
Parametric Diagram
Block Definition Diagram
Internal Block Diagram

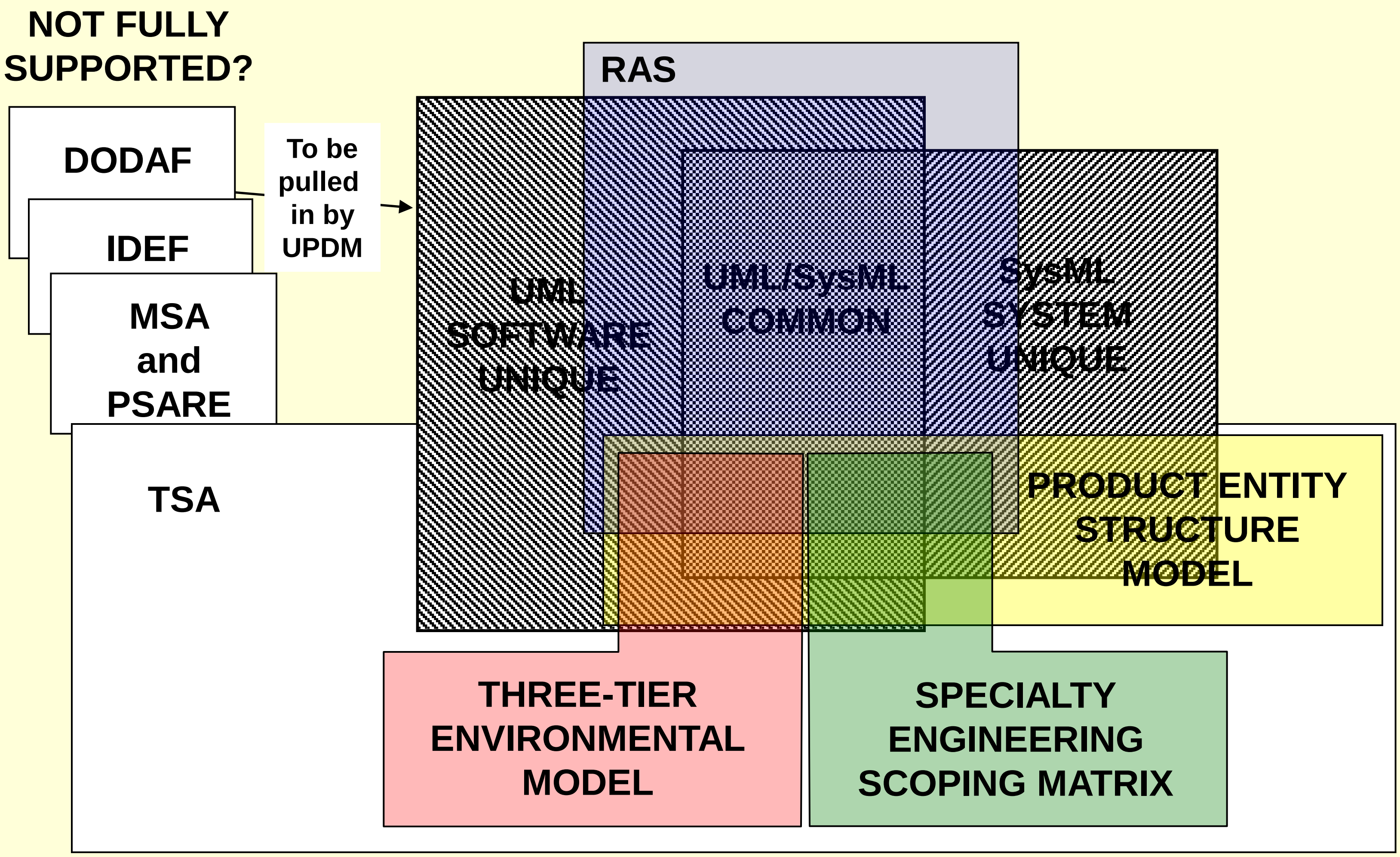
Common Diagrams

Use Case Diagram
Activity Diagram
State Diagram
Sequence Diagram
Package Diagram
Timing Diagram

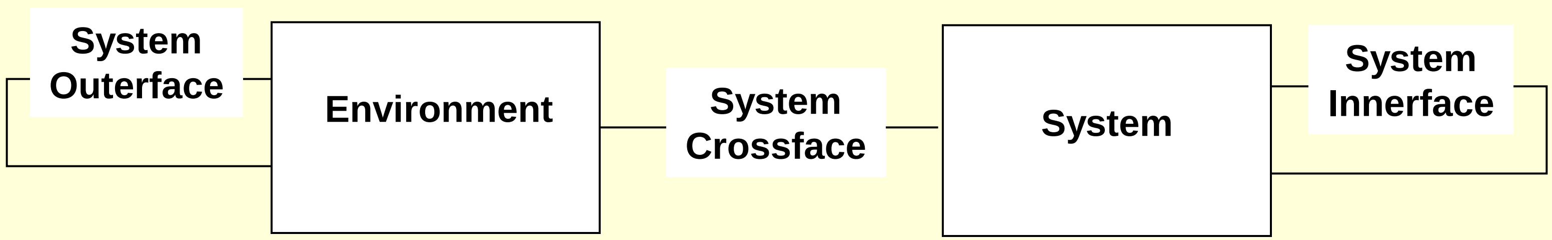
Toward a Universal Model
Using UML-SysML Modeling Artifacts



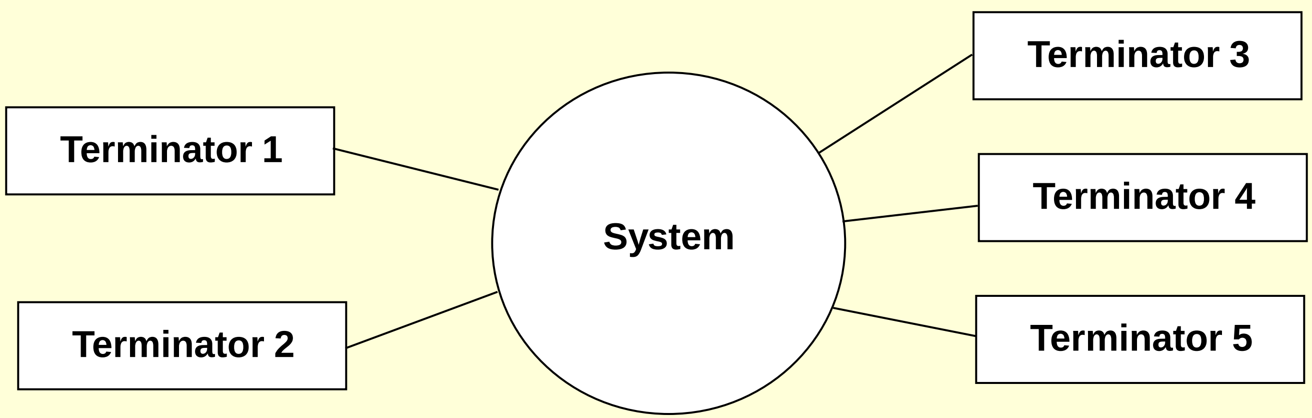
A Universal Model for the Future?



Two Older Inside-Outside Views of Systems



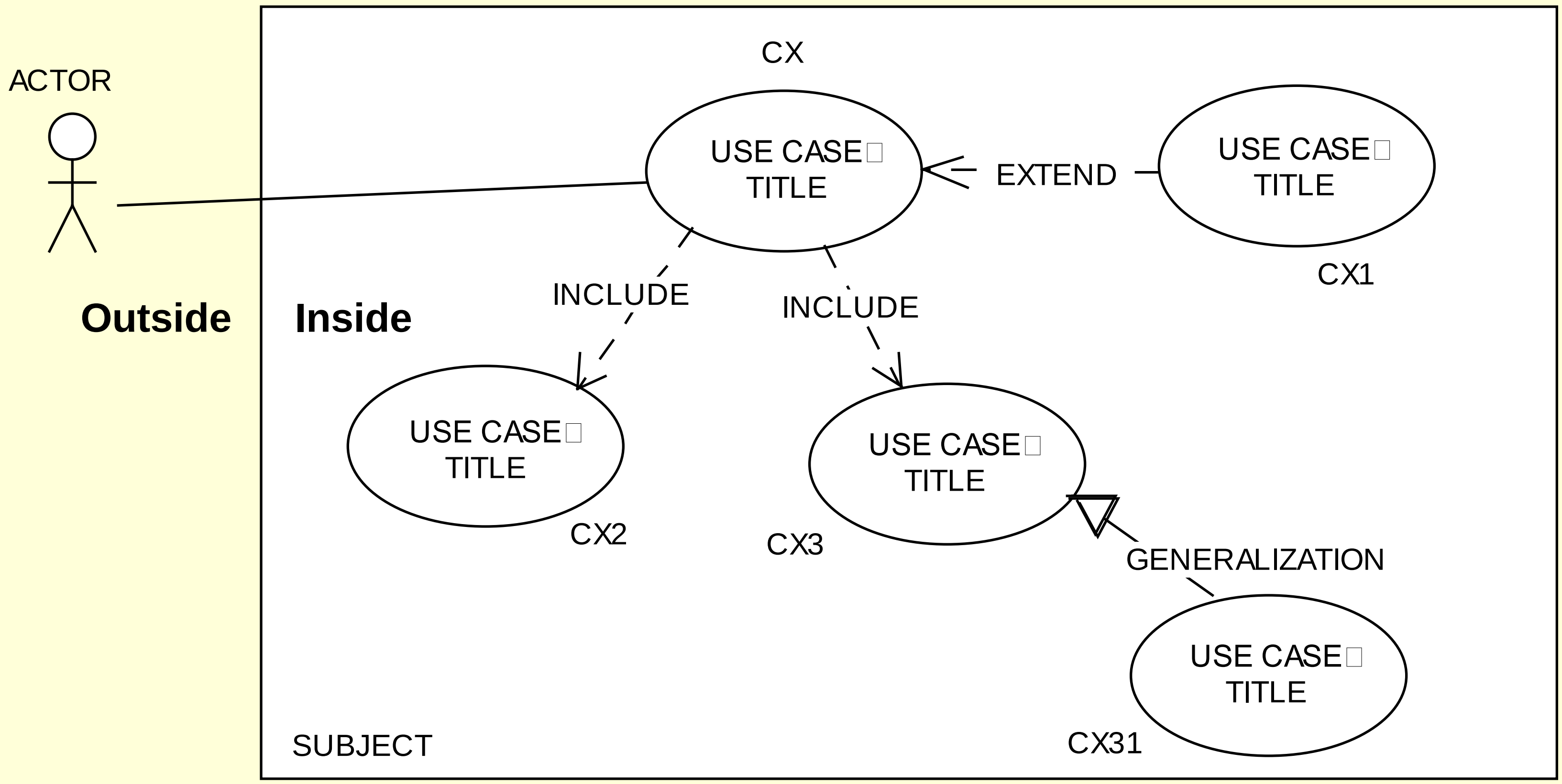
The Traditional Structured Analysis View of a System



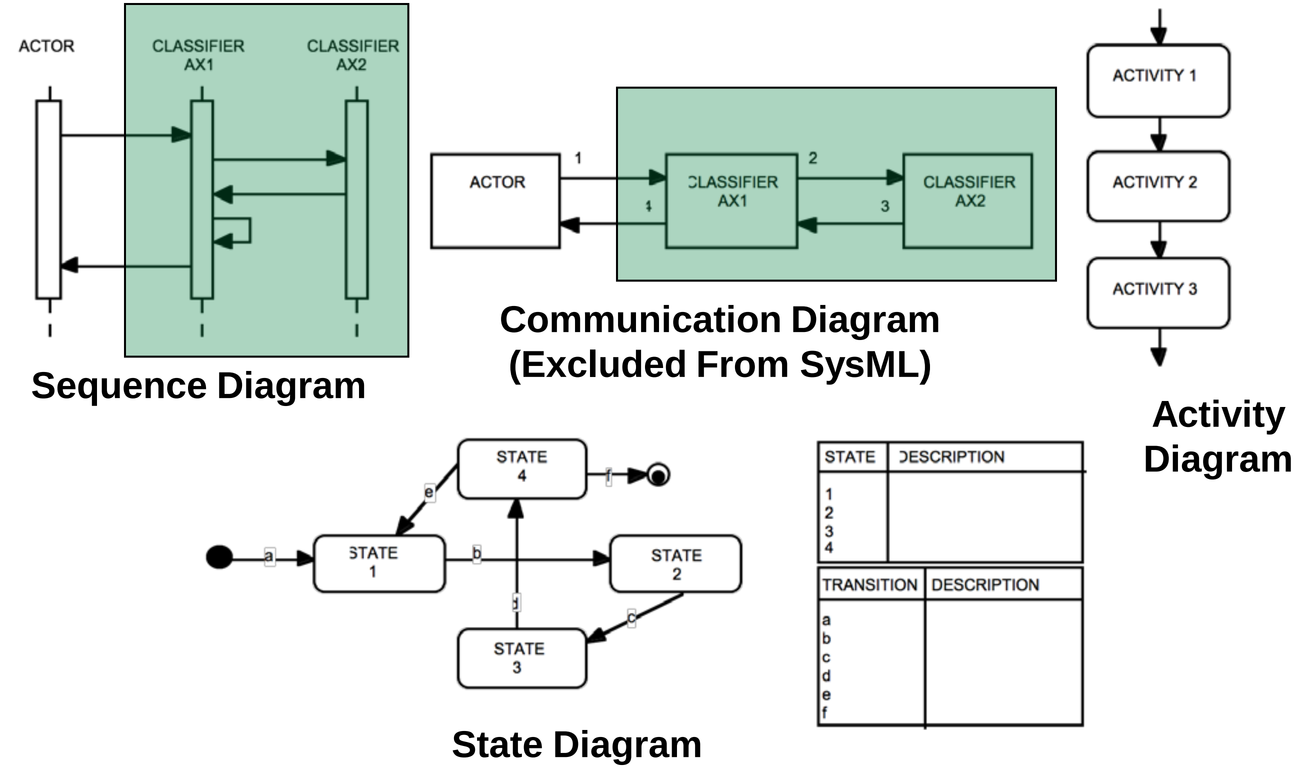
The Modern Structured Analysis View of a System
The Context Diagram

Use Cases

The UML-SysML View of a System



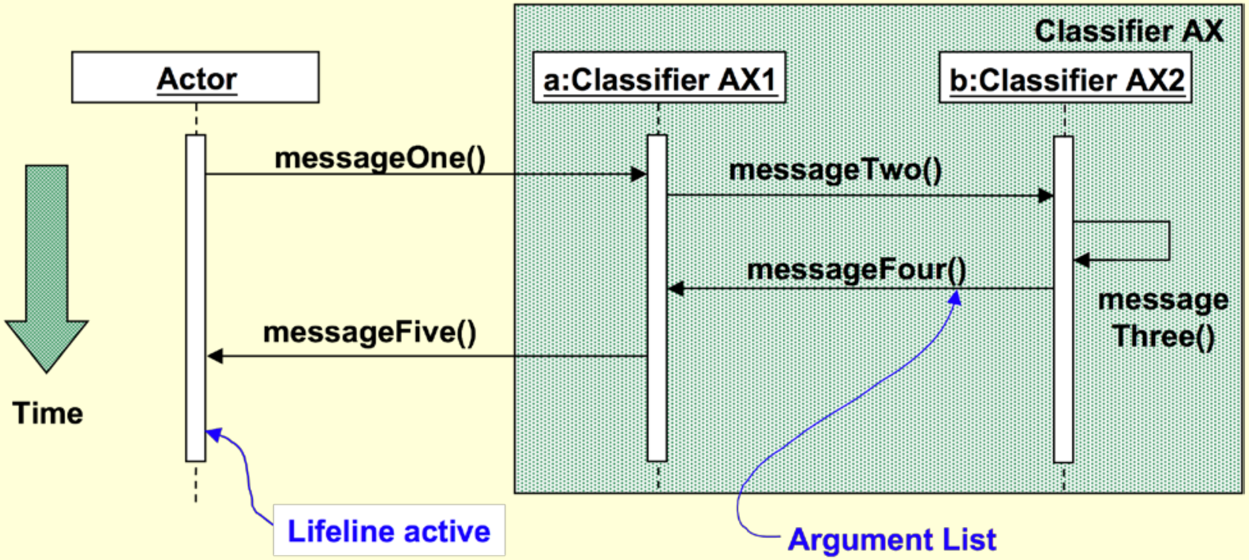
UML-SysML Dynamic Modeling Artifacts



STATE	DESCRIPTION
1	
2	
3	
4	

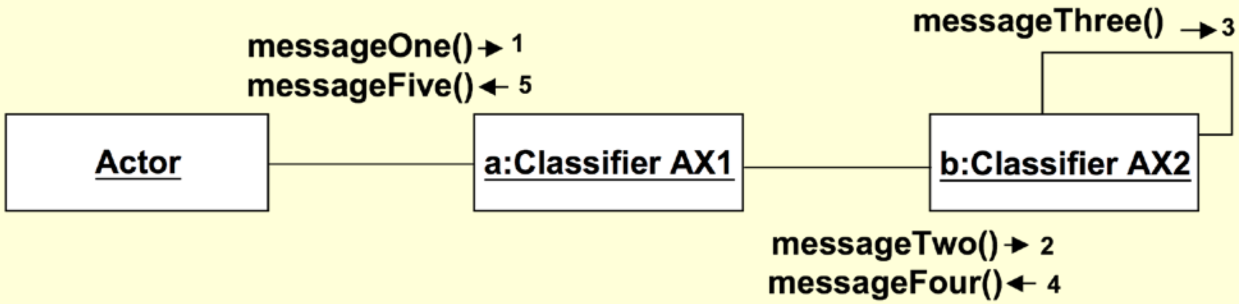
TRANSITION	DESCRIPTION
a	
b	
c	
d	
e	
f	

Sequence Diagram



Covers messages between entities but can't handle material or energy relationships in UML. Can do in SysML.

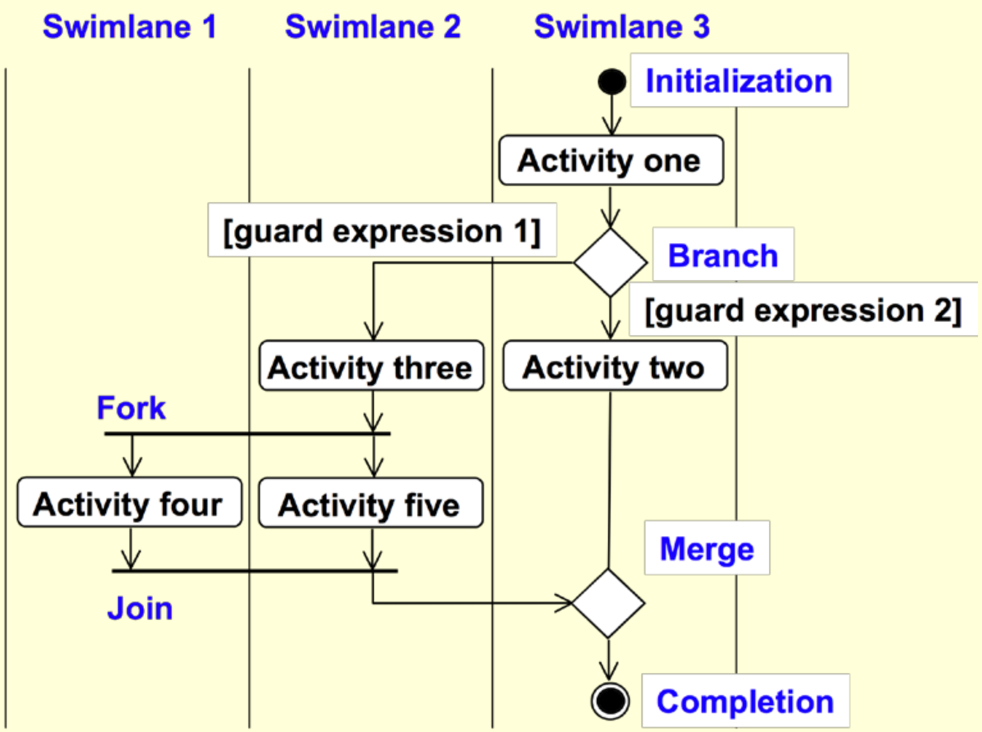
Communication Diagram



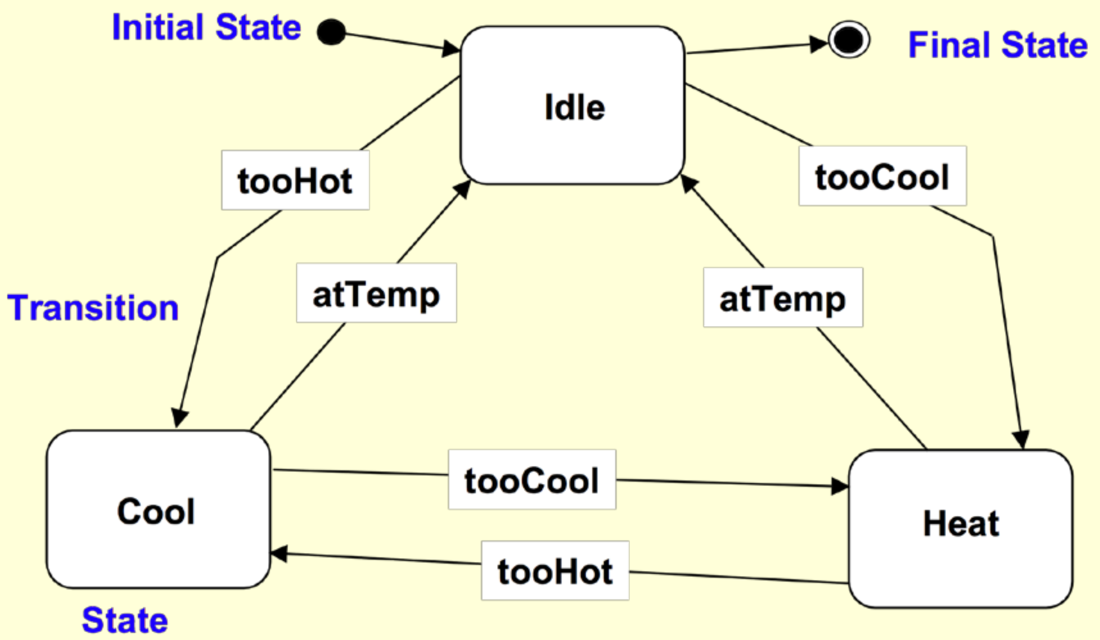
- Actually identical to sequence diagram content.
- Note used in SysML where it is replaced with block diagrams

Activity Diagram

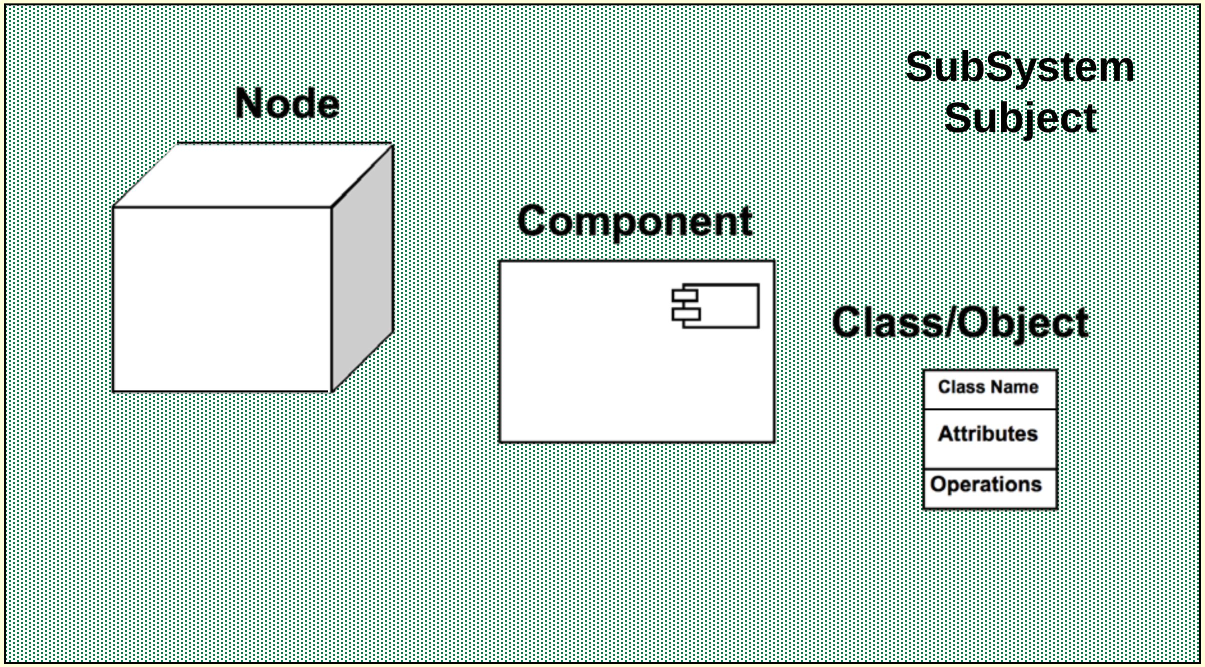
The old flow chart lives!



State Diagram
For a Bang-Bang Thermal Control System



UML Static Modeling Artifacts
Organization of the Design Solution

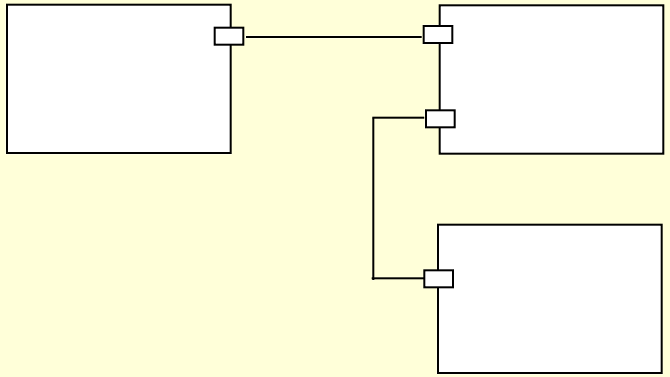


Replaced by block diagrams in SysML

SysML Blocks

Static Elements of the System

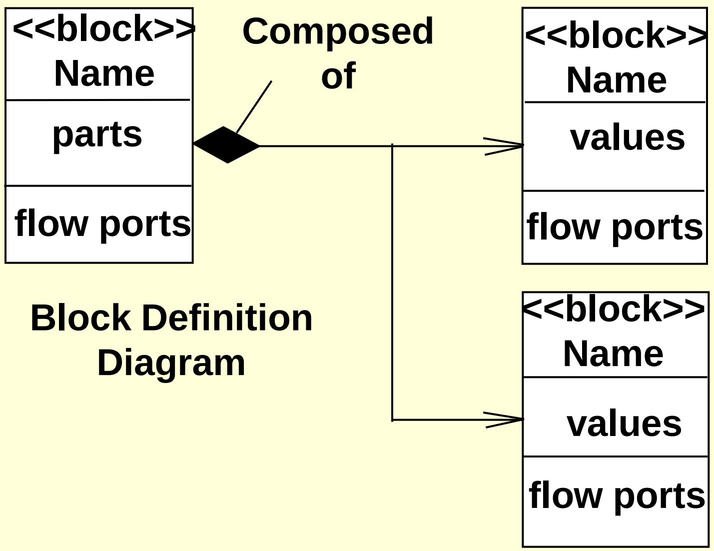
- Internal Block Diagram
- Exposes the internal interactions inside of a block
- The interconnecting lines can represent relationships as diverse as the glue holding the wiper sensor to windshield or an interface command channel



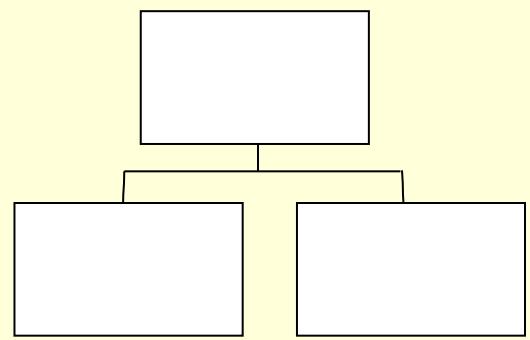
- Object, component, and node used in UML – why different?

SysML Blocks

Static Elements of the System

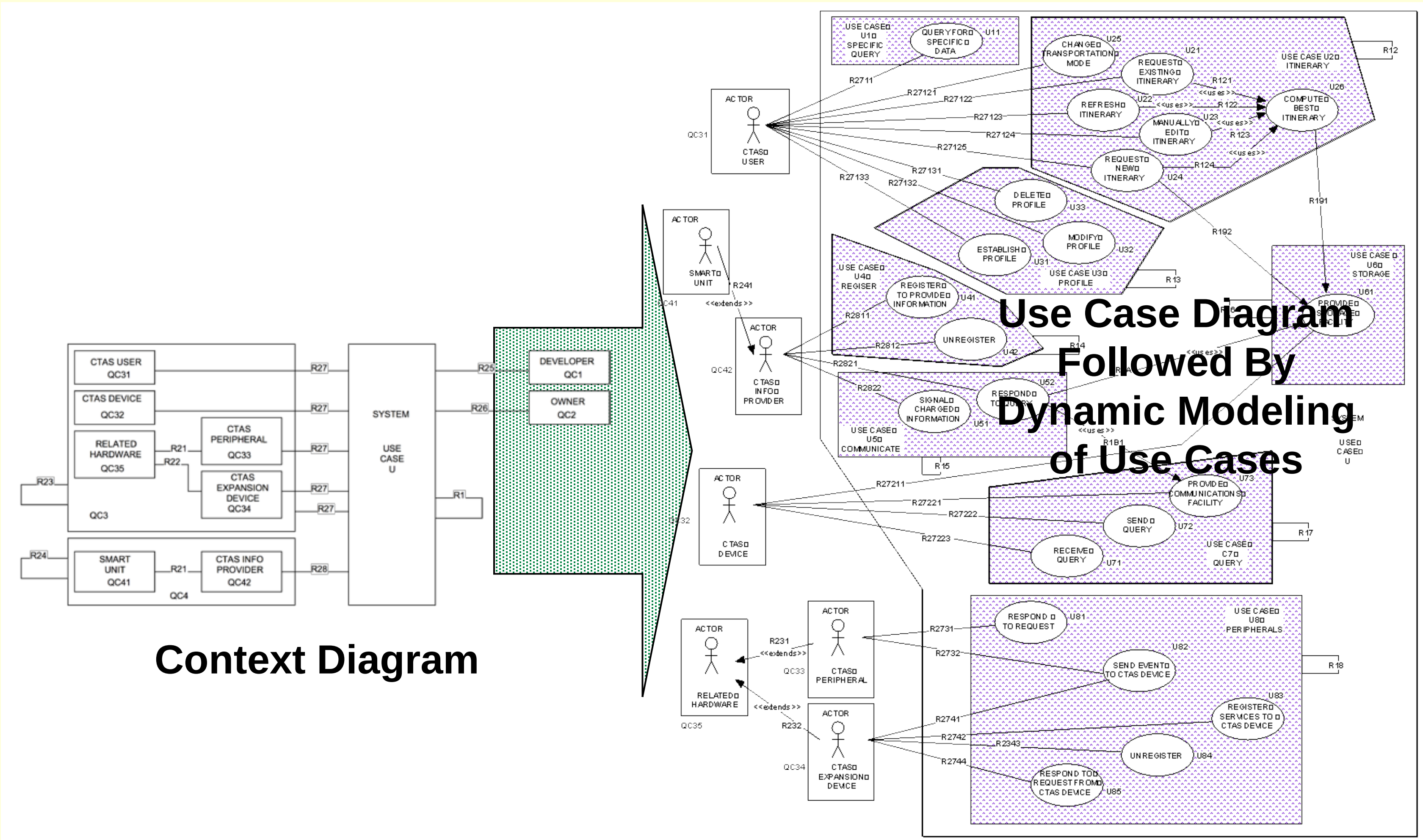


Block Definition Diagram

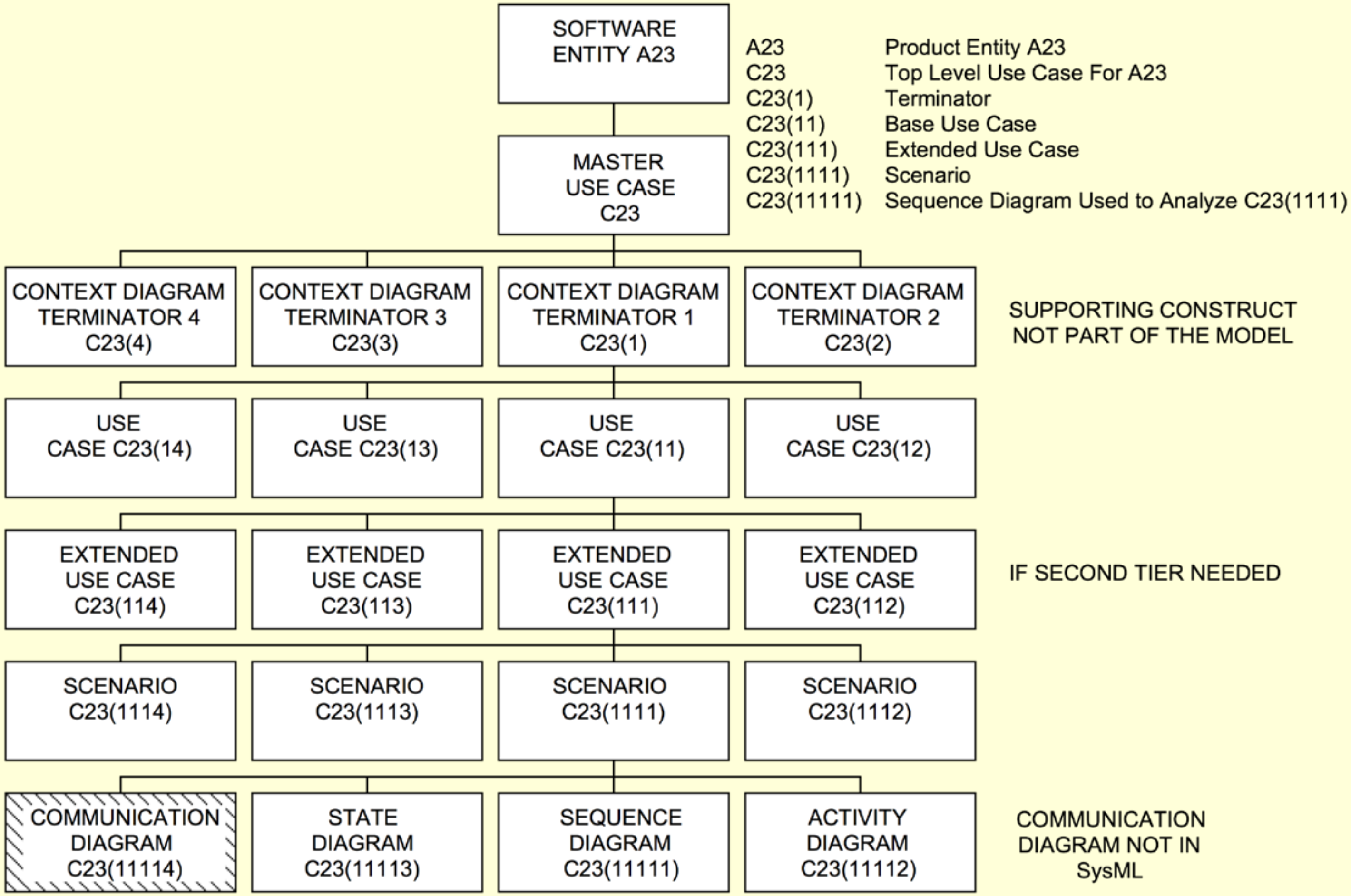


Equivalent Product Entity Block Diagram

SysML/UML Modeling
Use Case Analysis Example

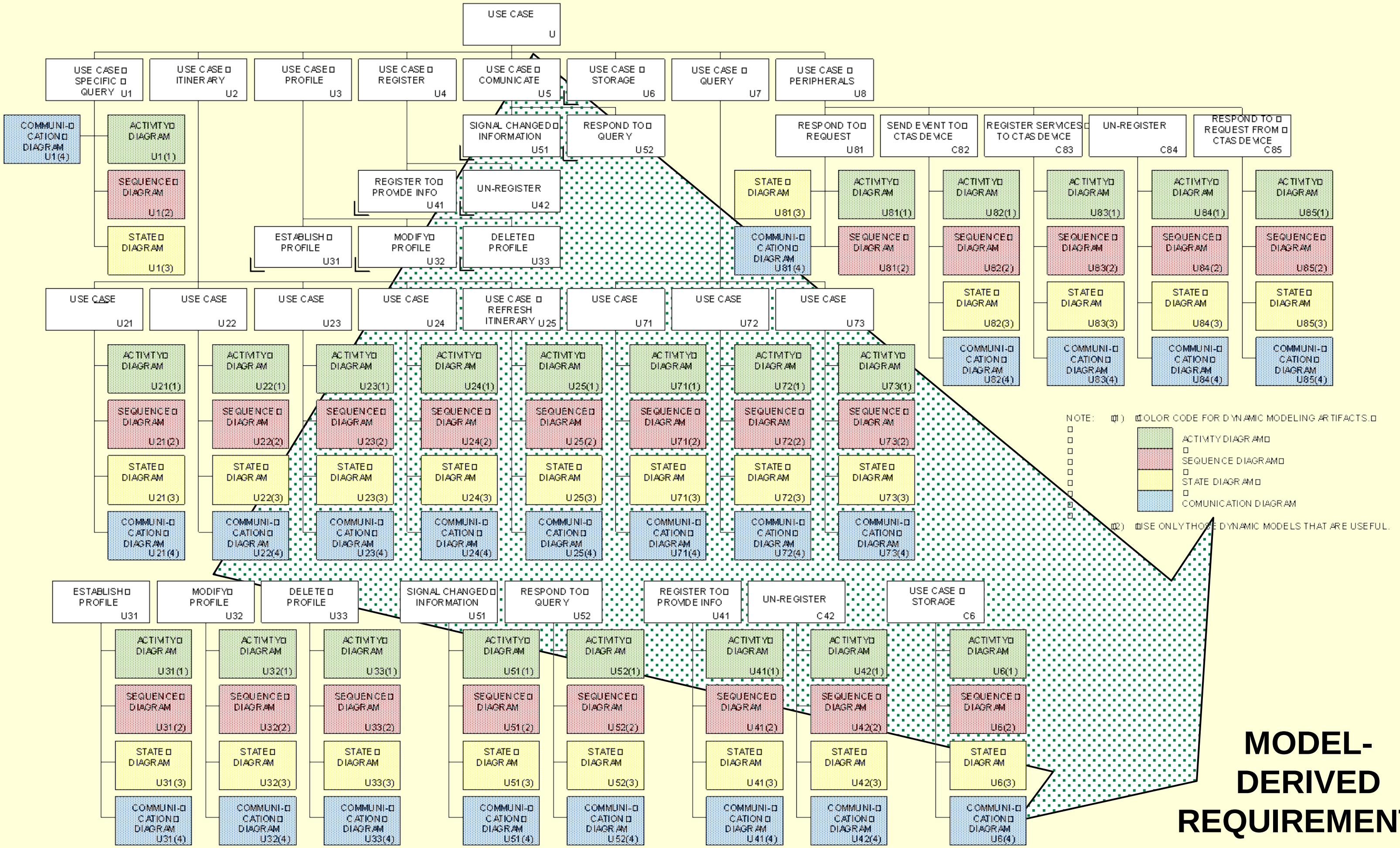


Hierarchical Structure for UML-SysML Analysis



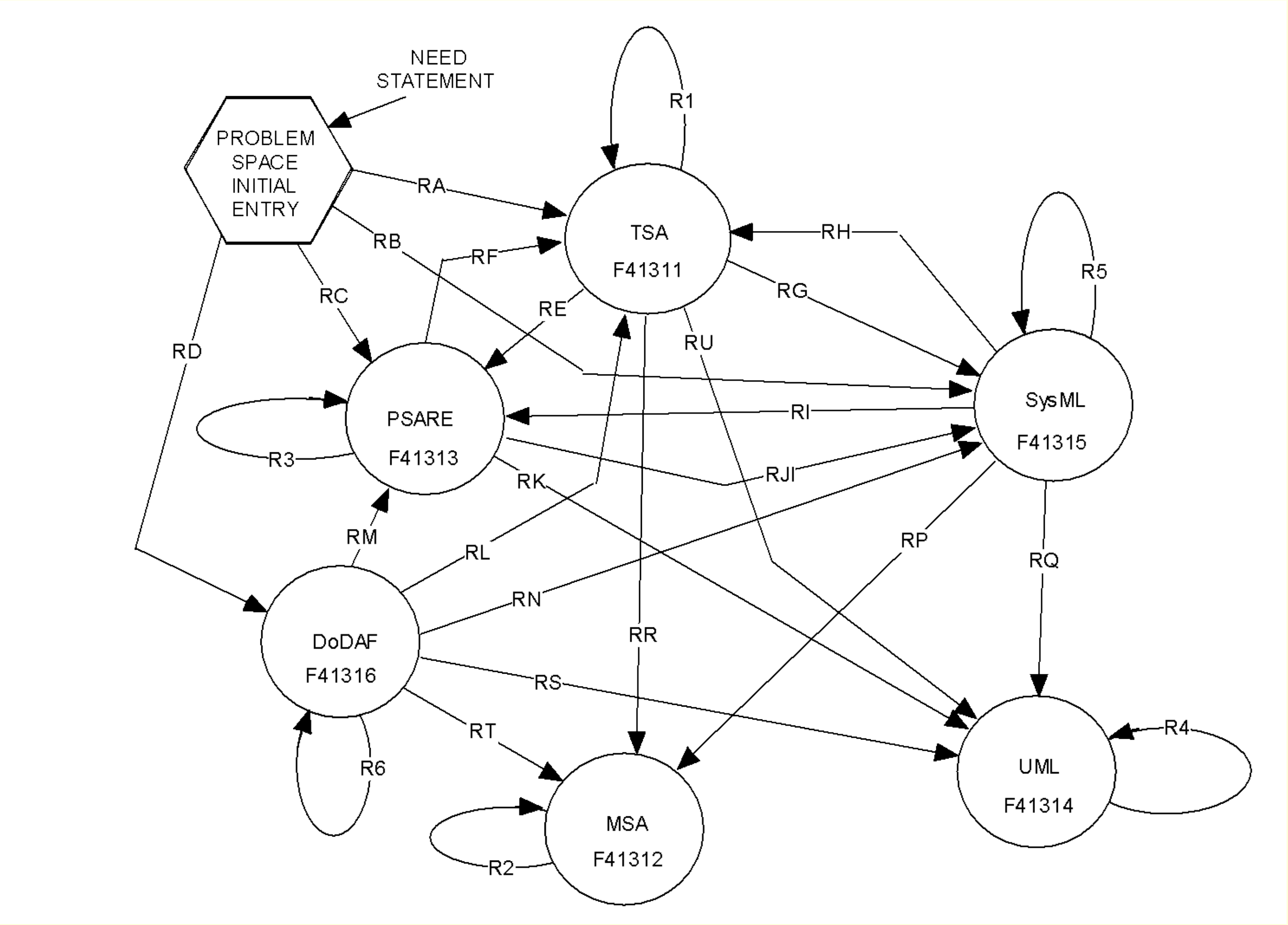
SysML/UML Modeling

Dynamic Modeling Artifacts Example

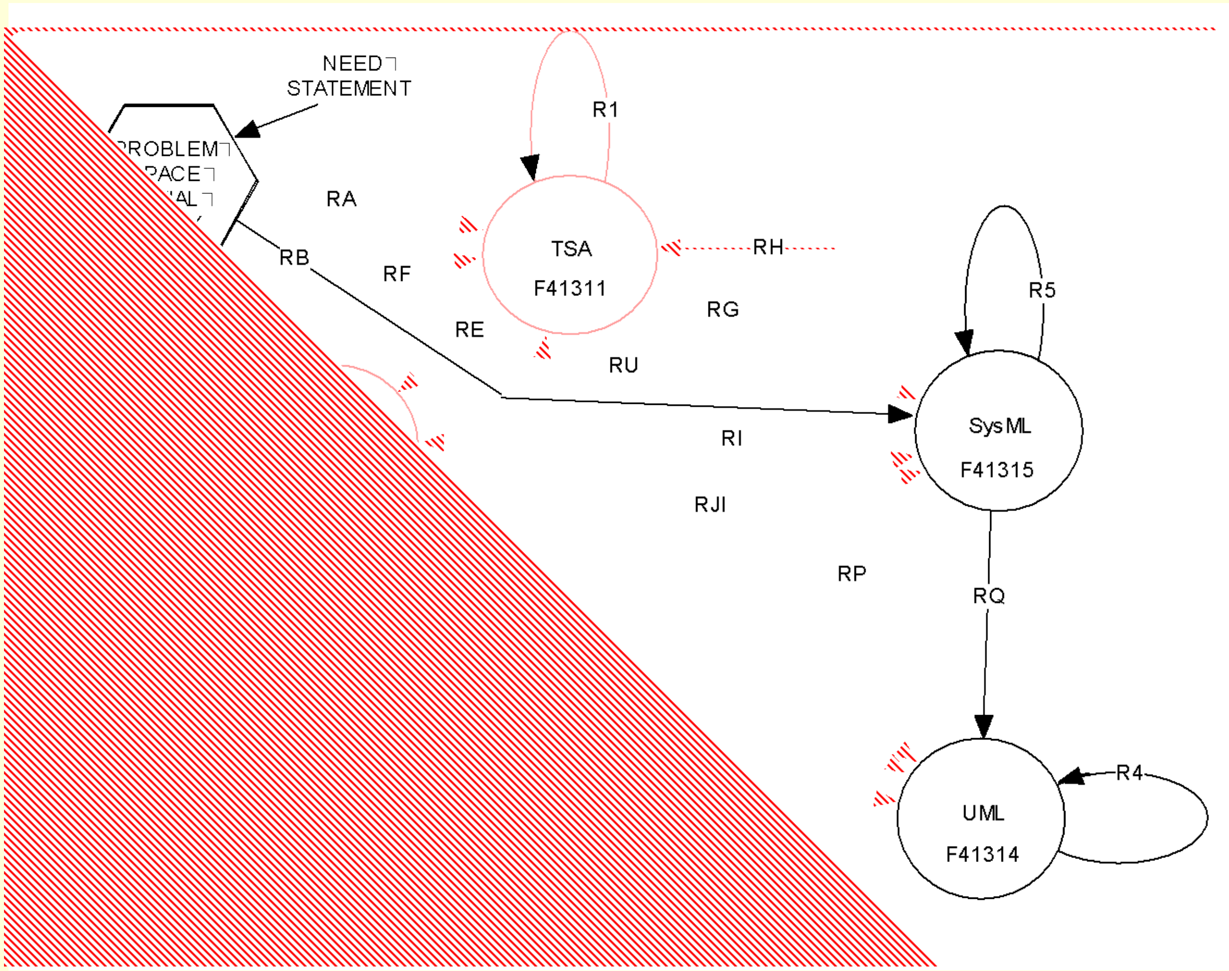


MODEL-DERIVED REQUIREMENTS

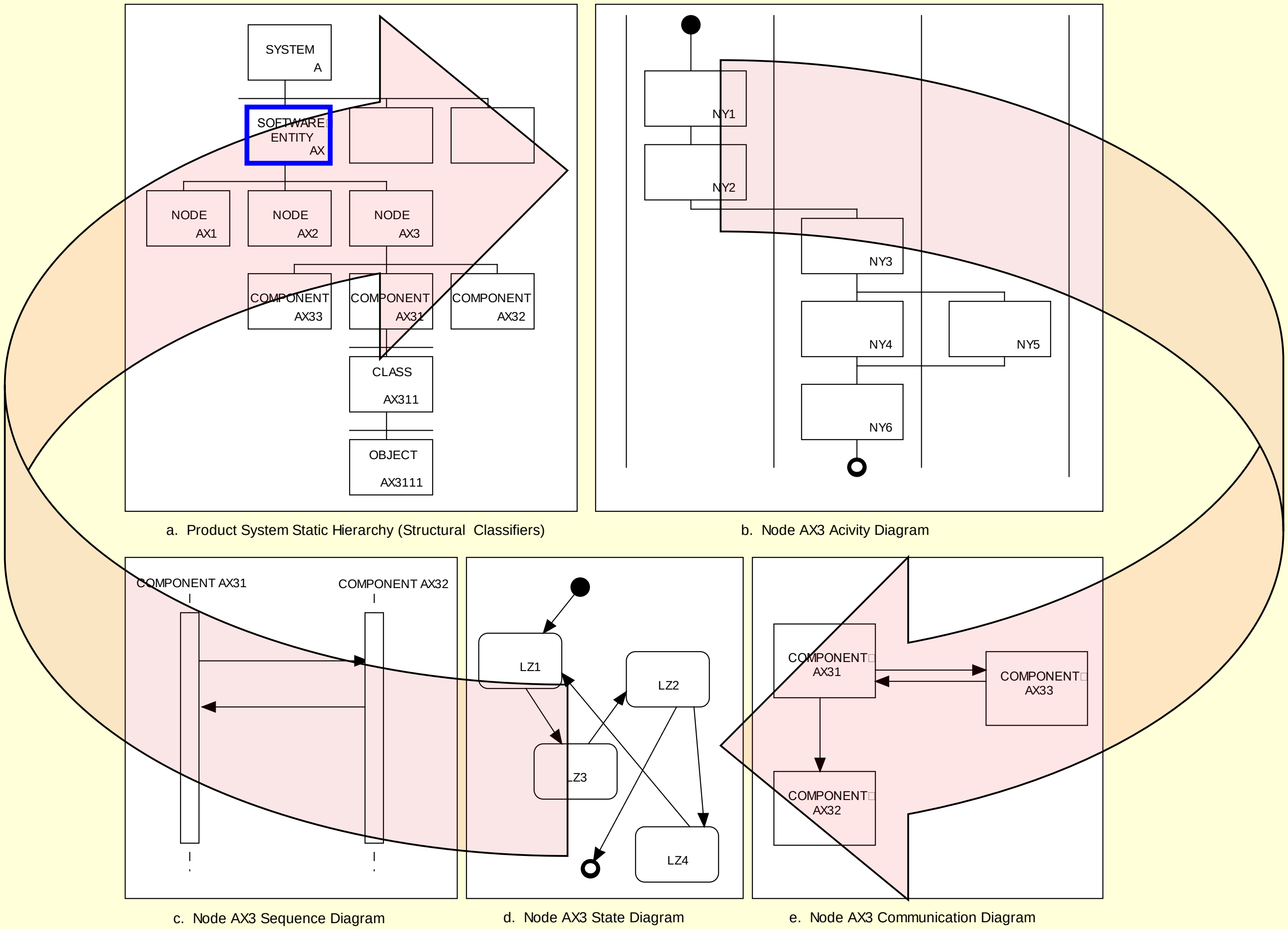
All Possible Inter-Model Transfers



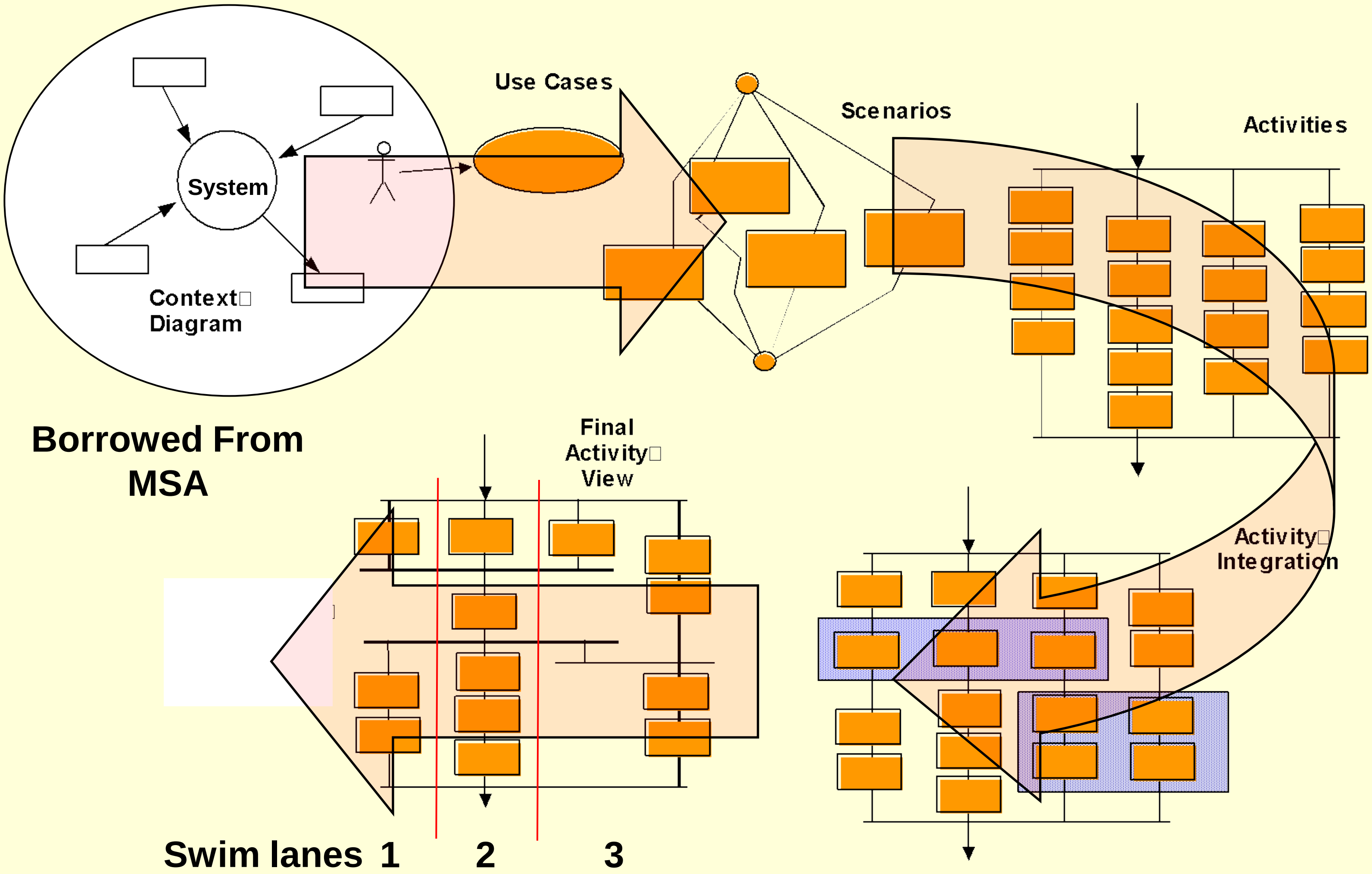
UADF Inter-Model Transfers With a SysML/UML UADF



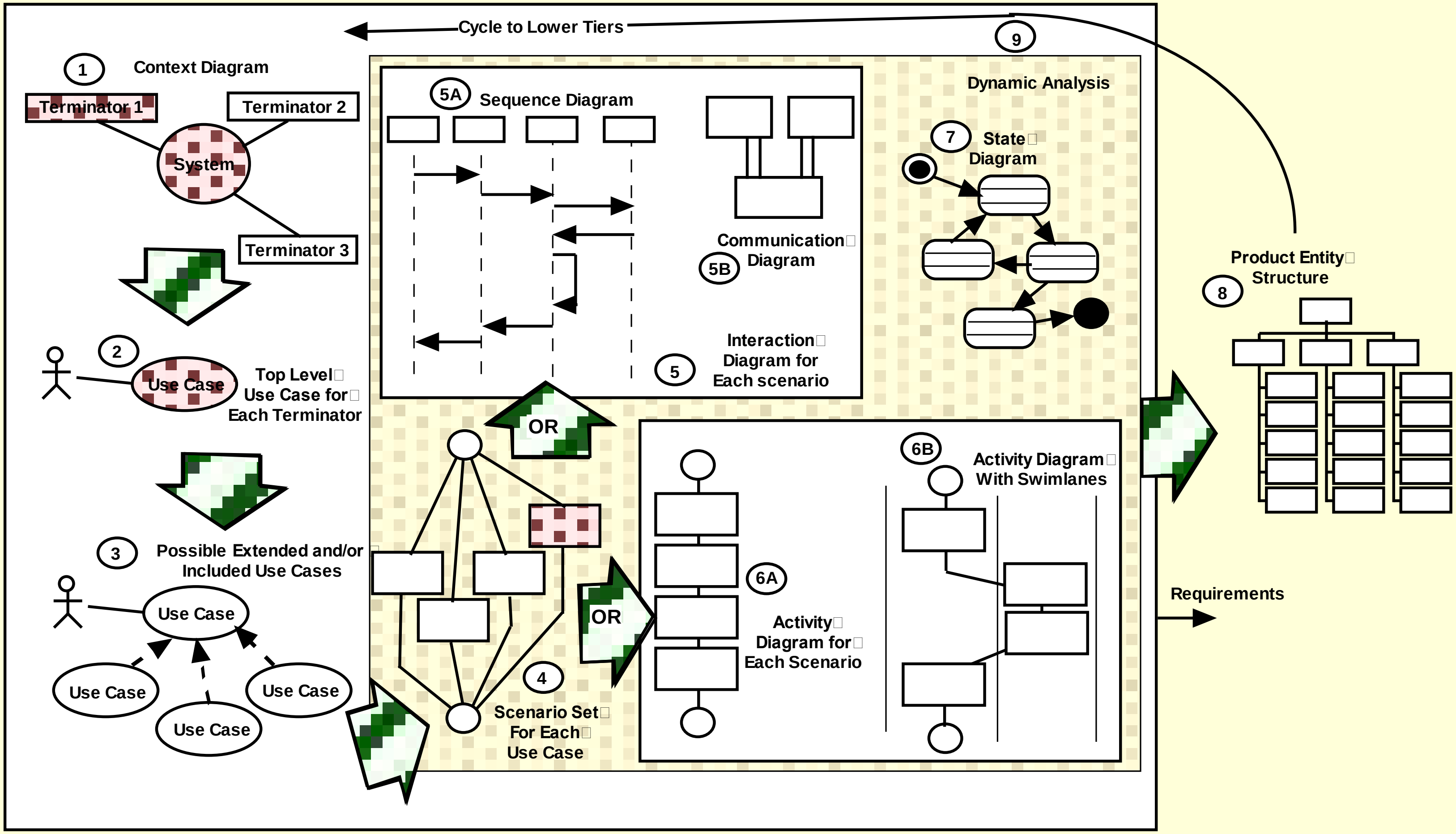
UML- SysML Cyclical Analysis



Entity Identification Using UML-SysML

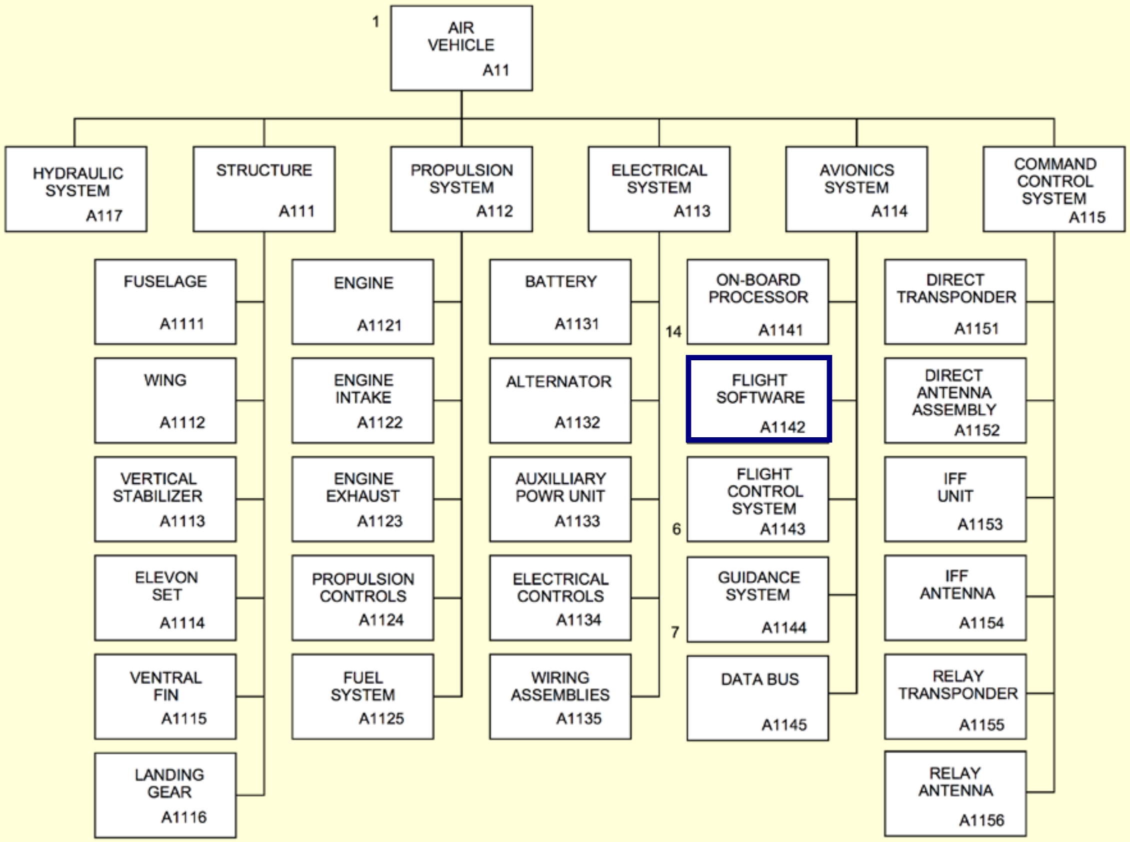


UML-SysML Dynamic Modeling Overview

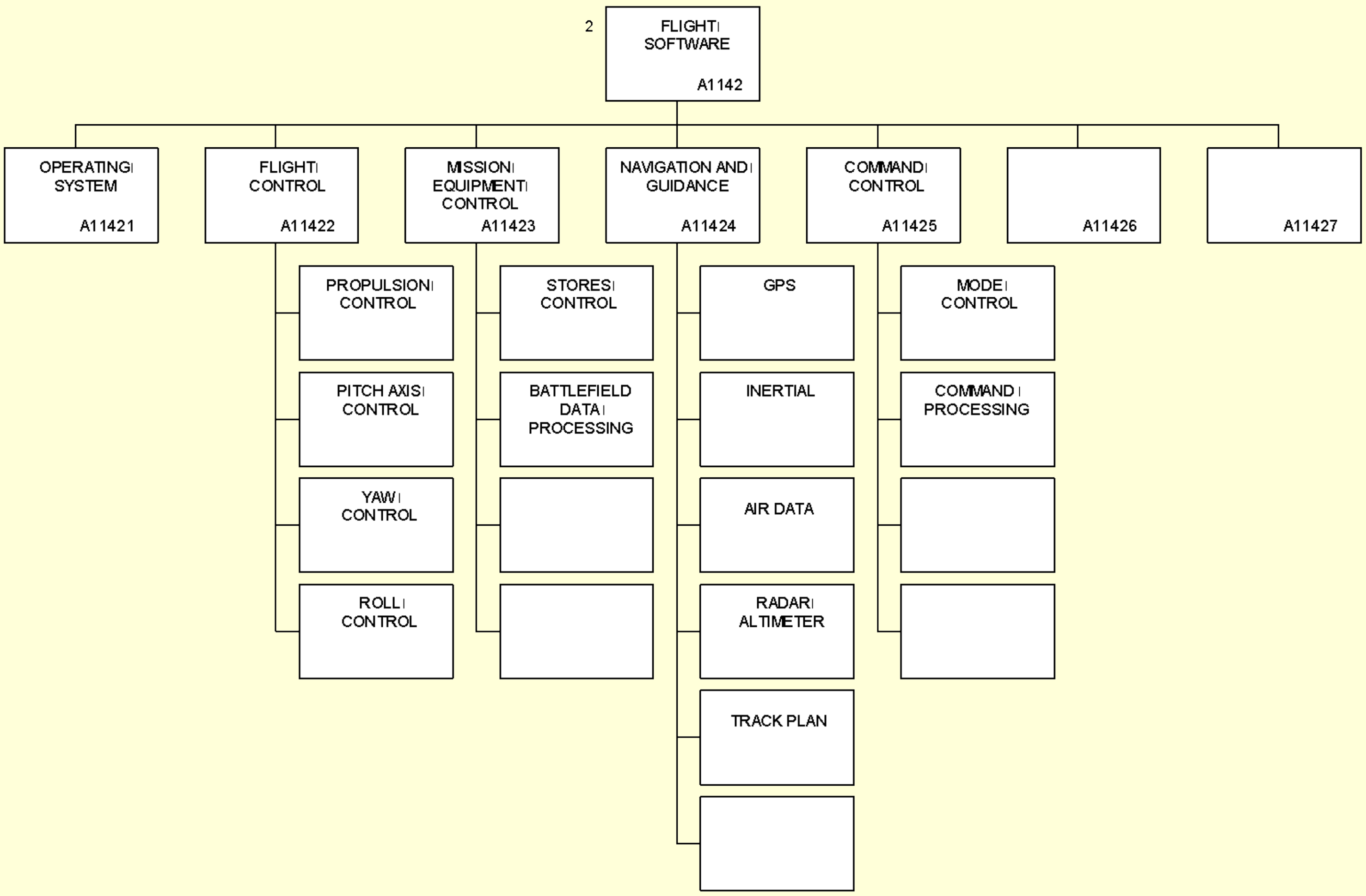


Combined Product Entity Structure

Entity identification flows from sequence, activity, or communication diagramming work

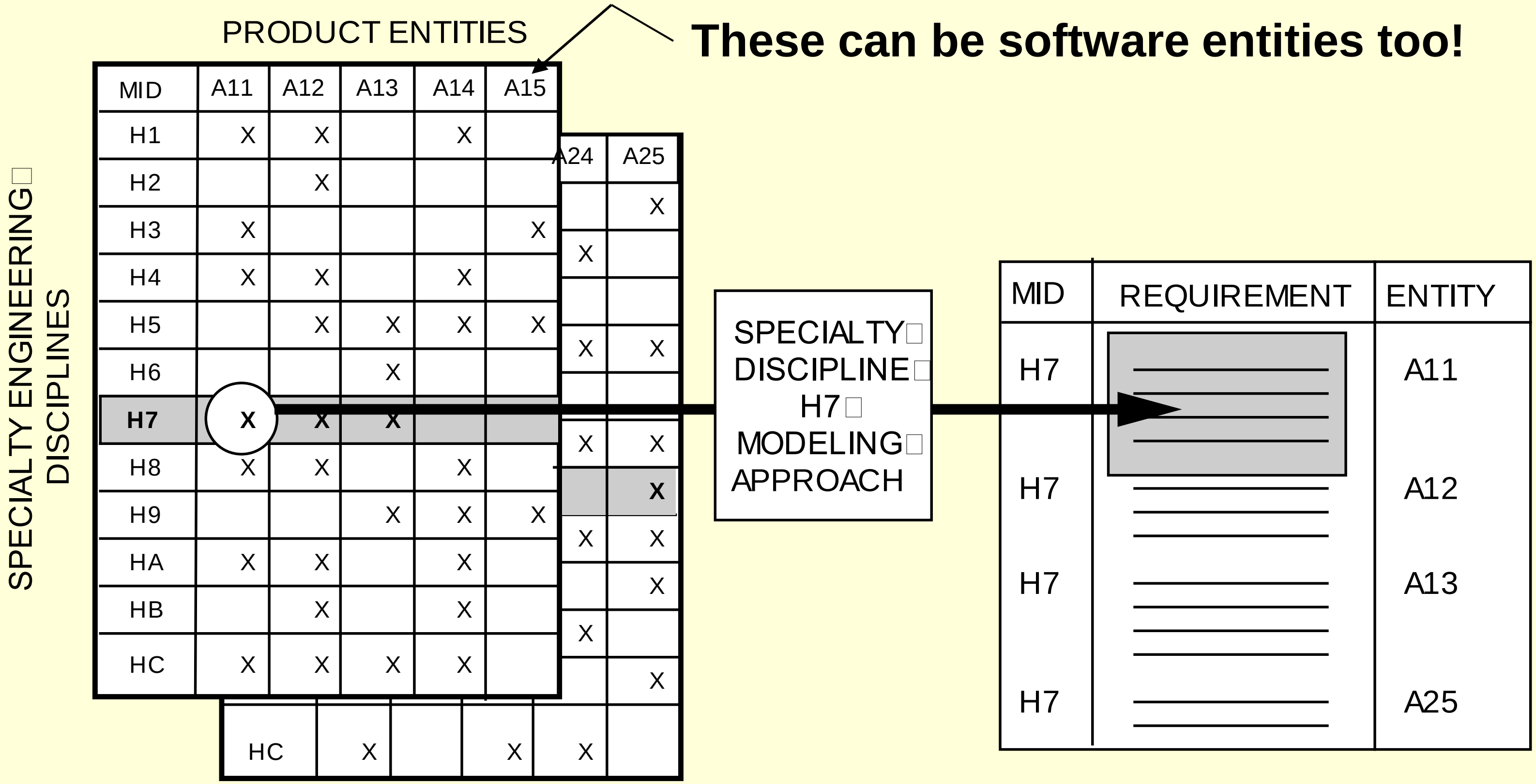


Possible Software Expansion



Specialty Engineering Modeling

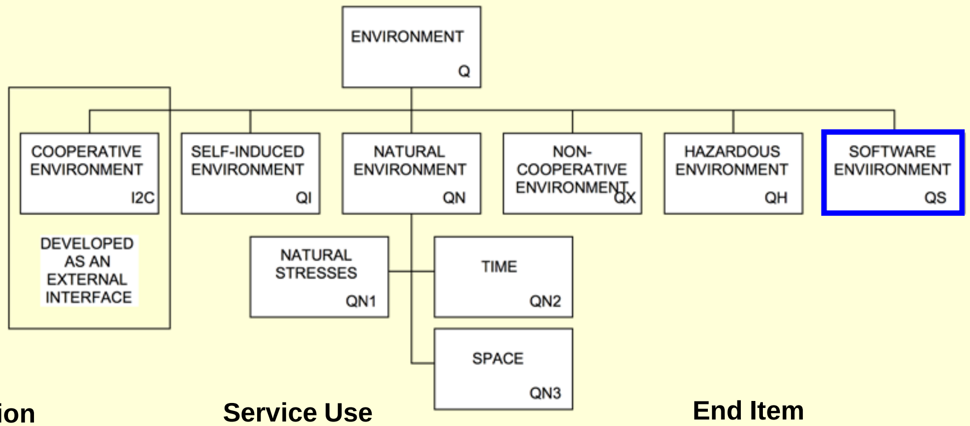
Once we know what the entities are we can investigate them from a specialty engineering perspective

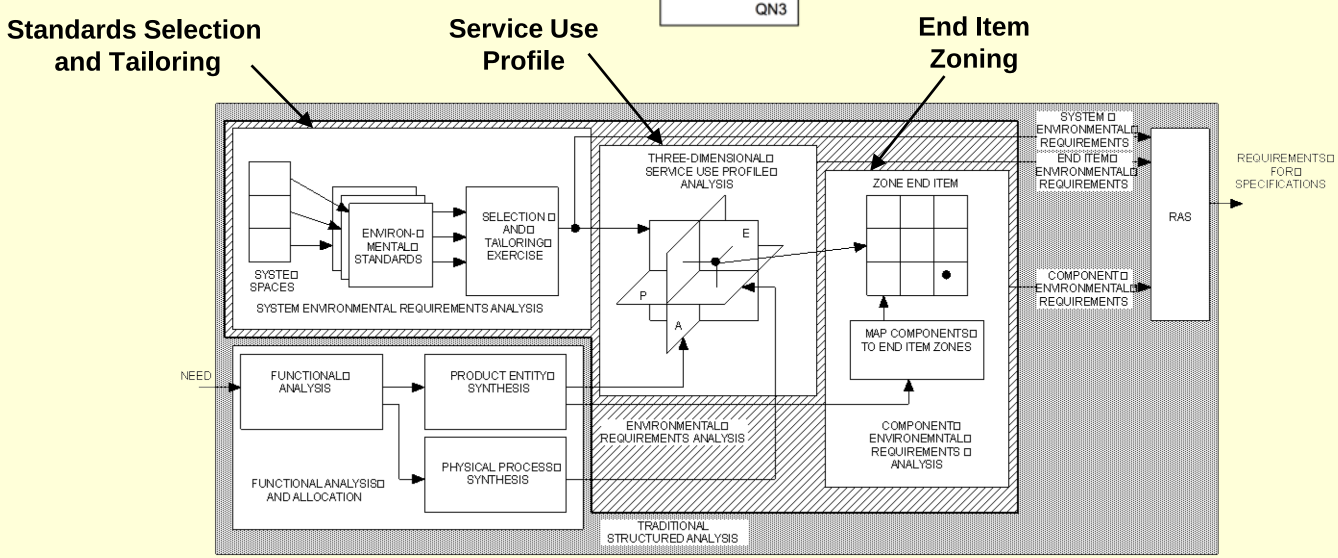


a. Specialty Engineering Scoping Matrix

b. Requirements Analysis Sheet (RAS)

Environmental Modeling





Computer Software Environmental Factors

- Language
- Compiler
- Machine Structure
- Memory
- Clock Speed

JOG SYSTEM ENGINEERING
GRAND SYSTEMS DEVELOPMENT TRAINING PROGRAM
TUTORIAL

UNIVERSAL ARCHITECTURE DESCRIPTION FRAMEWORK

THE FUTURE

What Will the Future Look Like?

- A single model for the problem space - no matter the specific product - will be developed in hardware or software
- Requirements embedded in problem space models encouraging requirements compliance in design models with the specifications appearing in the form of models
- A connected series of models for design
- Inter-model effects observable directly rather than individual human interpretation of effects followed by conversation and action - can we do this?
- Verification linkage through models
- Eventual connection between the problem space modeling and CAD-CAM models.
- A business process model coordinated with engineering modeling

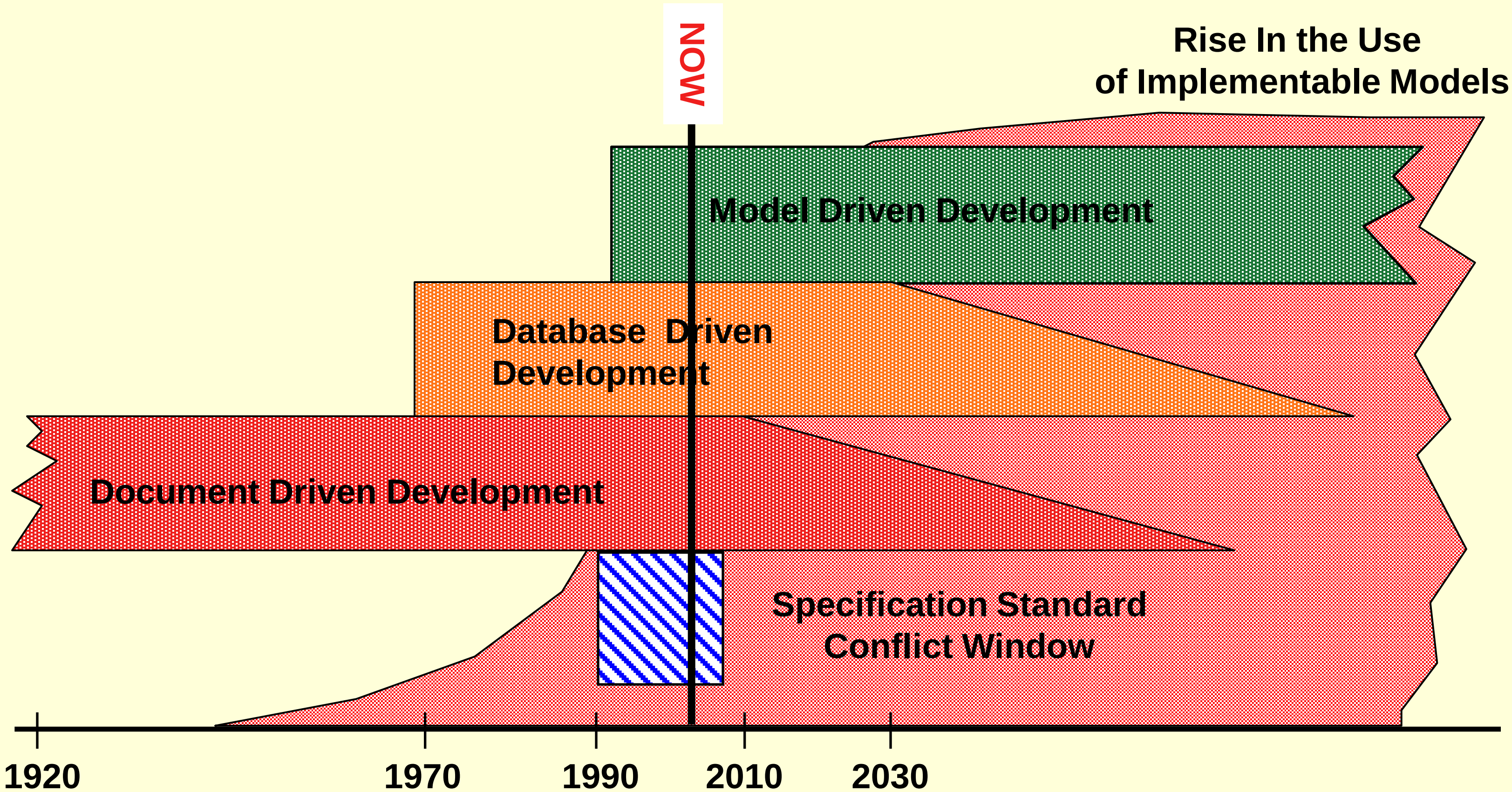
Model-Driven Challenges

- Will it be possible for managers to avoid whiplash due to the speed of the analytical process?
- Can we provide adequate exposure of the on-going and dynamic modeling work to encourage sound management of the development process?
- Will it really be possible to build models that fully express the problem space essential characteristics (requirements) while permitting a solution space larger than a single solution?

The Computer Network Becomes a Team Member in Good Standing

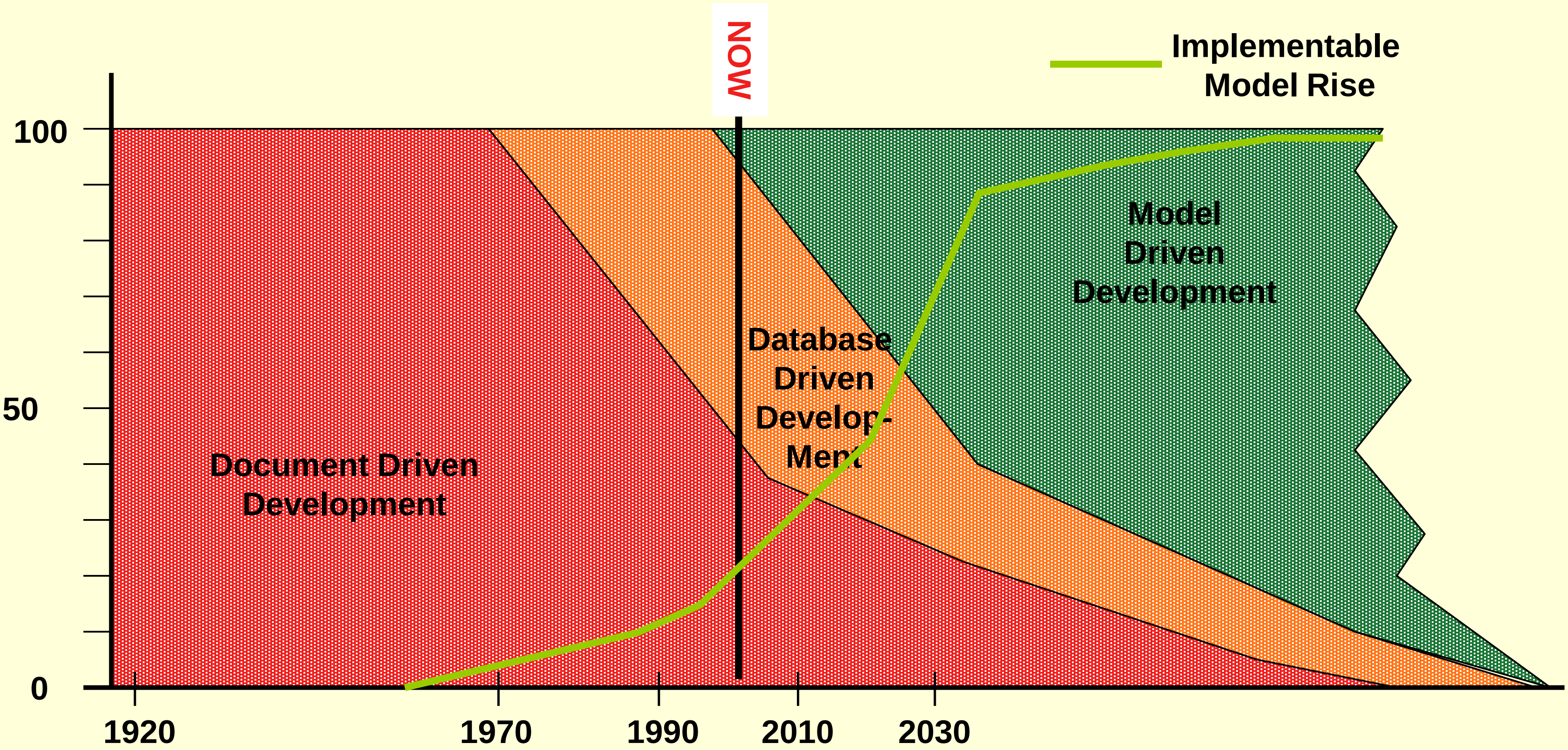
Will there be room for human emotion in the development process? I hope so!

Development Evolution Timeline, Driving Methods Staging



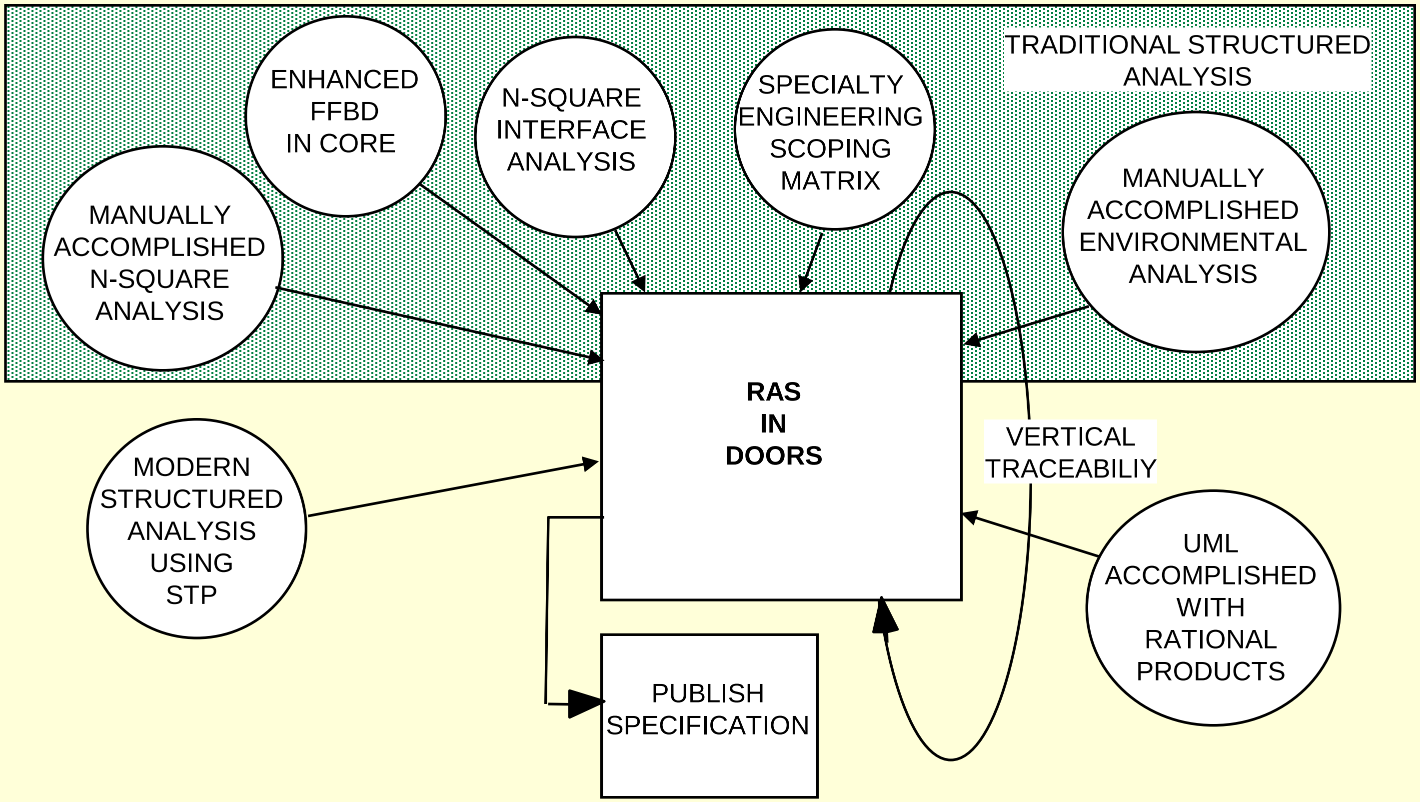
05-15-2002 DATA UNSUBSTANTIATED

Development Evolution Timeline, Program Percentages?

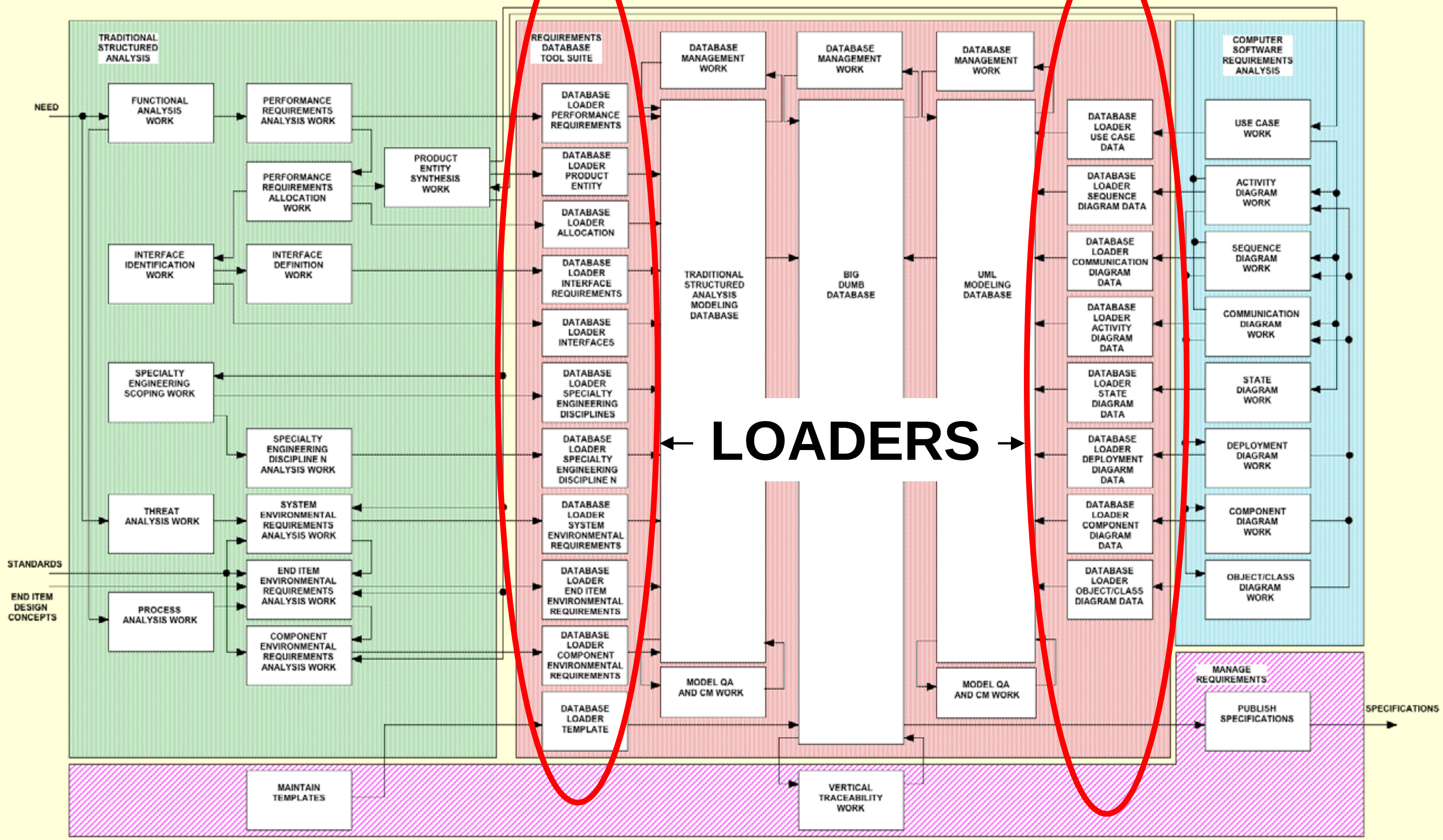


05-15-2002 DATA UNSUBSTANTIATED

Our Current Best Toolbox?

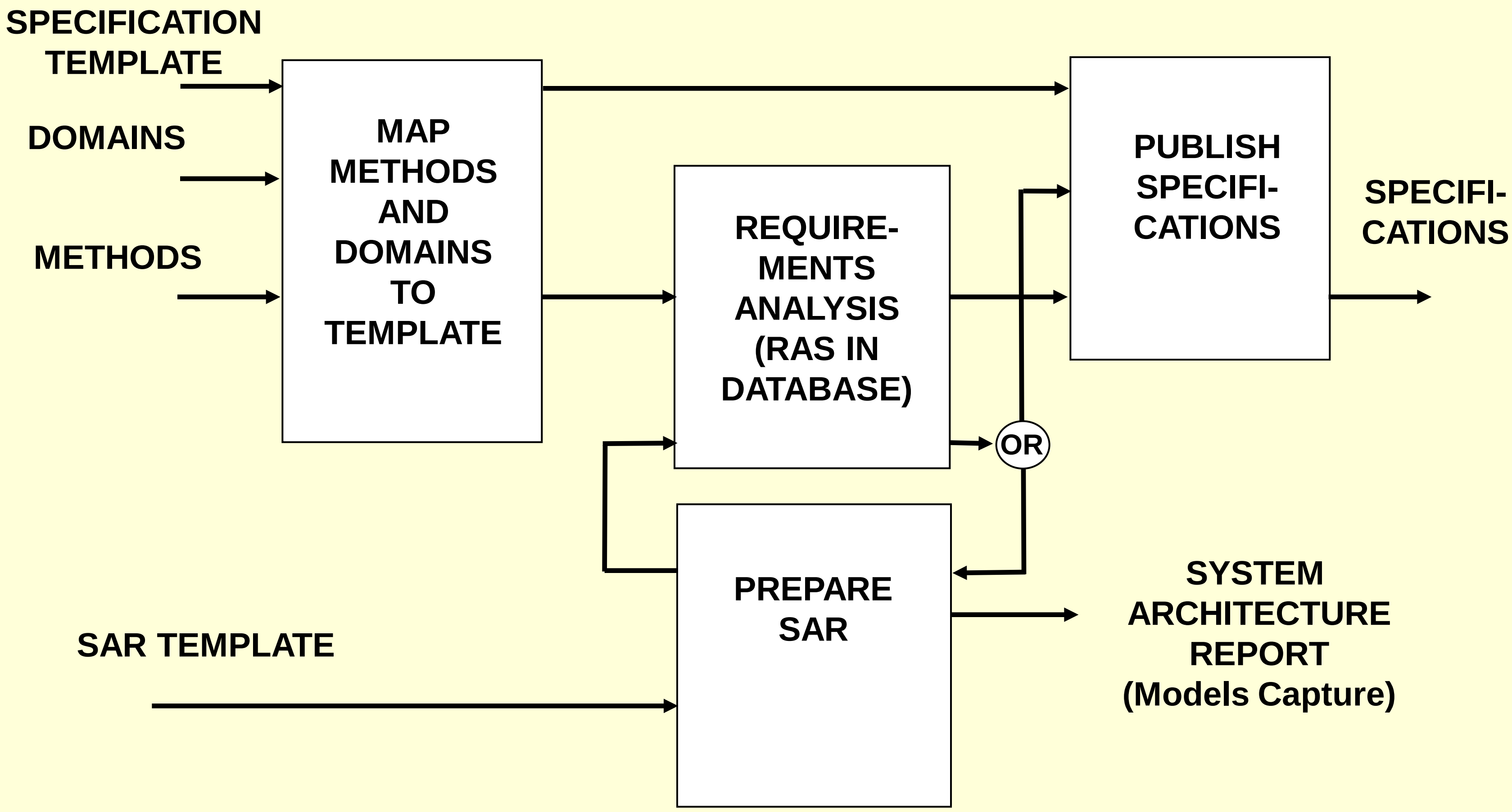


Possible Interim Tools Suite



Specification Generator

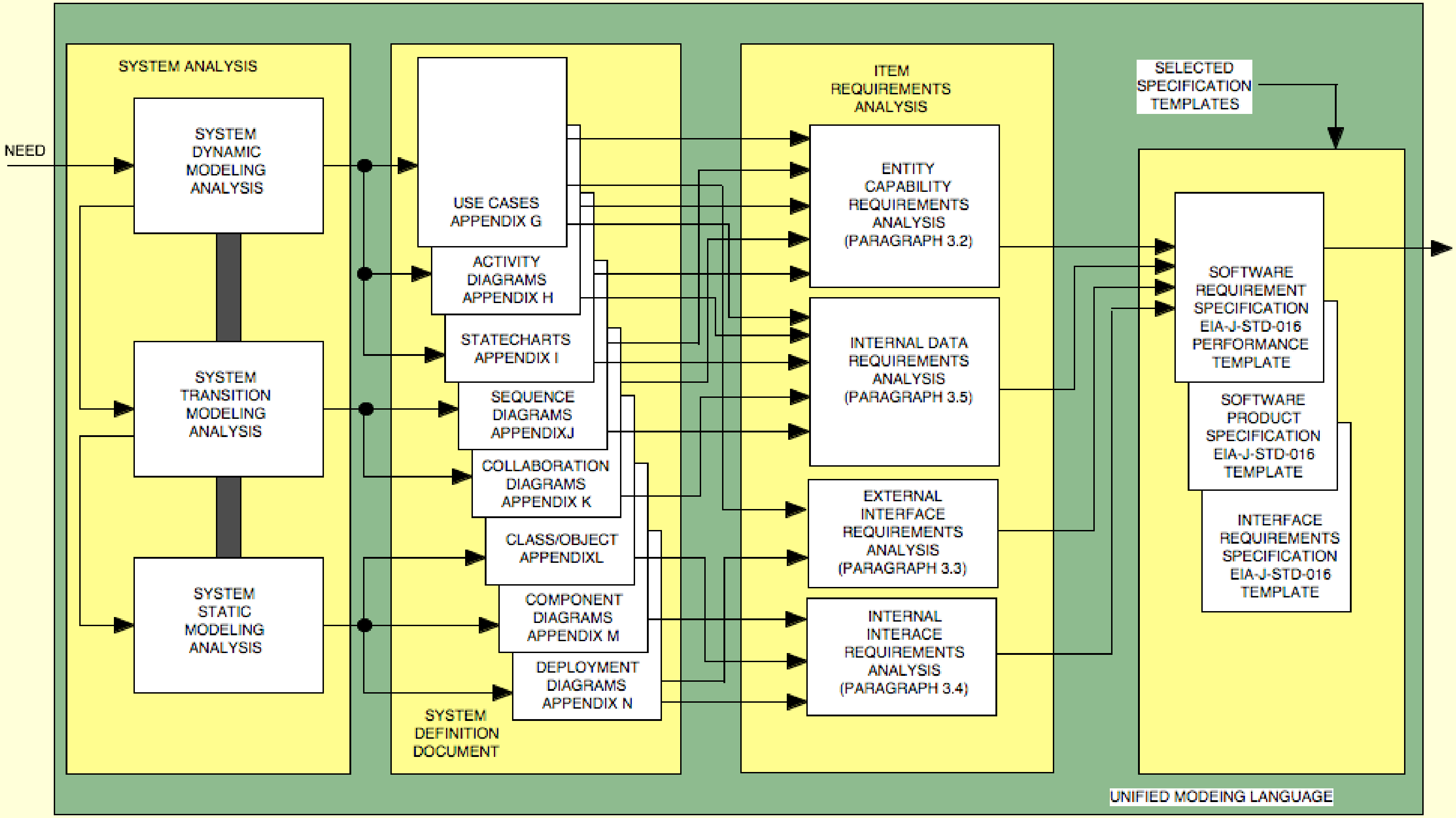
The Same Machinery Used for TSA



Three Ways to Capture the Modeling

- Within specification paragraph 3.1.3
- In a system architecture report (SAR) referenced in paragraph 3.1.3
- Within the computer tool used to accomplish the modeling work with a reference in paragraph 3.1.3 to the tool content

SAR Organization For UML-SysML



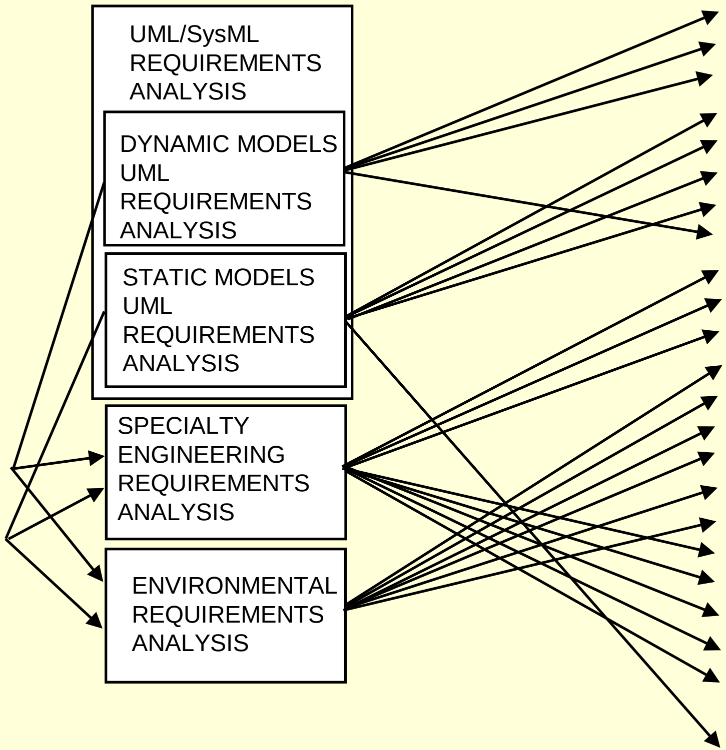
Combined RAS

We need a set of MID codes that we can use to couple modeling structures to requirements derived from them.

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	XR67	Storage Volume < 10 ISO Vans	A1	Sensor Subsystem		
H	Specialty Engineering Disciplines			A	Product System		
H11	Reliability	EW34	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	FYH4	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	6GHU	Mean Time to Repair < 0.2 Hours	A1	Sensor Subsystem	3.1.6	Maintainability
H12	Maintainability	U9R4	Mean Time to Repair < 0.4 Hours	A11	Cable	3.1.6	Maintainability
H12	Maintainability	J897	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 Innerface			A1			
I181	Aggregate Signal Feed Source Impedance	E37H	Aggregate Signal Feed Source Impedance= 52 ohms ± 2 ohms	A1	Sensor Subsystem		
I181	Aggregate Signal Feed Load Impedance	E37I	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	6D74	-40 degrees F< Temperature < +140 degrees F	A	Product System		
QX	Non-Cooperative Environmental Stresses			A	Product System		

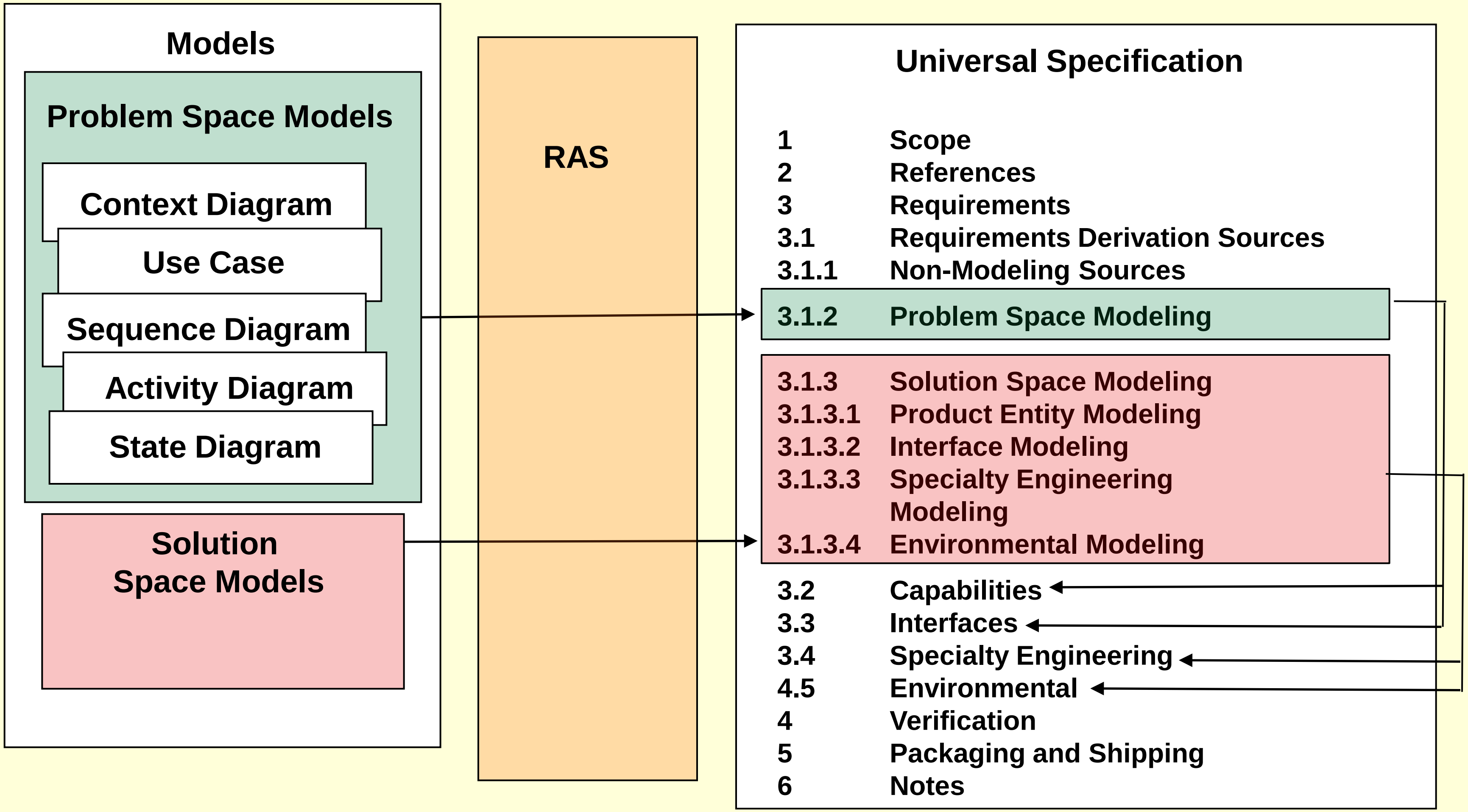
UML Model/SRS Template Correlation

If You Must Use MIL-STD-498/EIA J STD-016

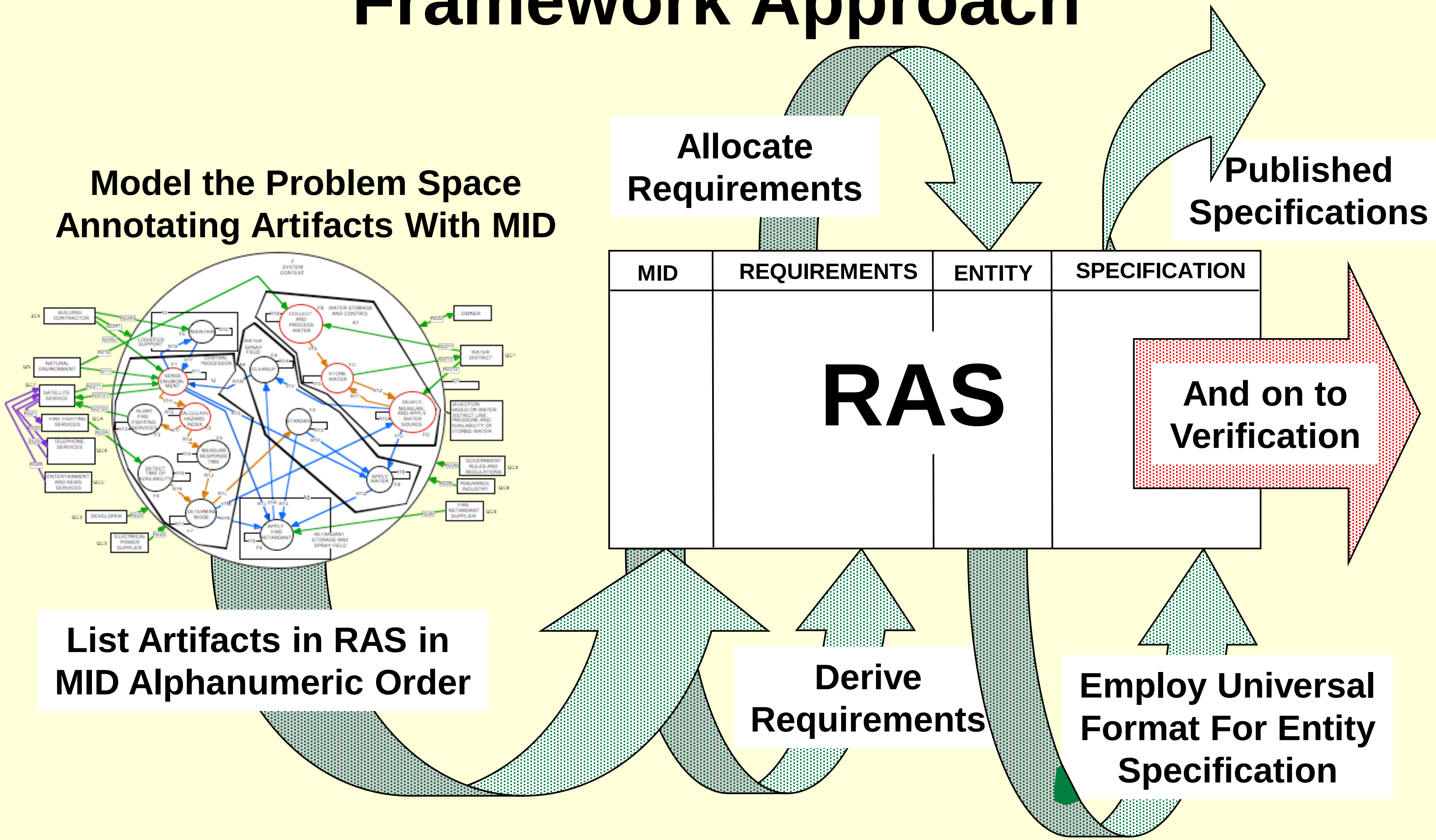


3	Requirements
3.1	Required States and Modes
3.2	CSCI Capability Requirements
3.2.x	(CSCI Capability)
3.3	CSCI External Interface Requirements
3.3.1	Interface Identification and Diagram
3.3.x	(Project Unique Interface Identifier)
3.4	CSCI Internal Interface Requirements
3.5	CSCI Internal Data Requirements
3.6	Adaptation Requirements
3.7	Safety Requirements
3.8	Security and Privacy Requirements
3.9	CSCI Environment Requirements
3.10	Computer Resource Requirements
3.10.1	Computer Hardware Requirements
3.10.2	Computer Hardware Resource Utilization Requirements
3.10.3	Computer Software Requirements
3.10.4	Computer Communications Requirements
3.11	Software Quality Factors
3.12	Design and Implementation Constraints
3.13	Personnel-Related Requirements
3.14	Training-Related Requirements
3.15	Logistics-Related Requirements
3.16	Other Requirements
3.17	Packaging Requirements
3.18	Precedence and Criticality of Requirements

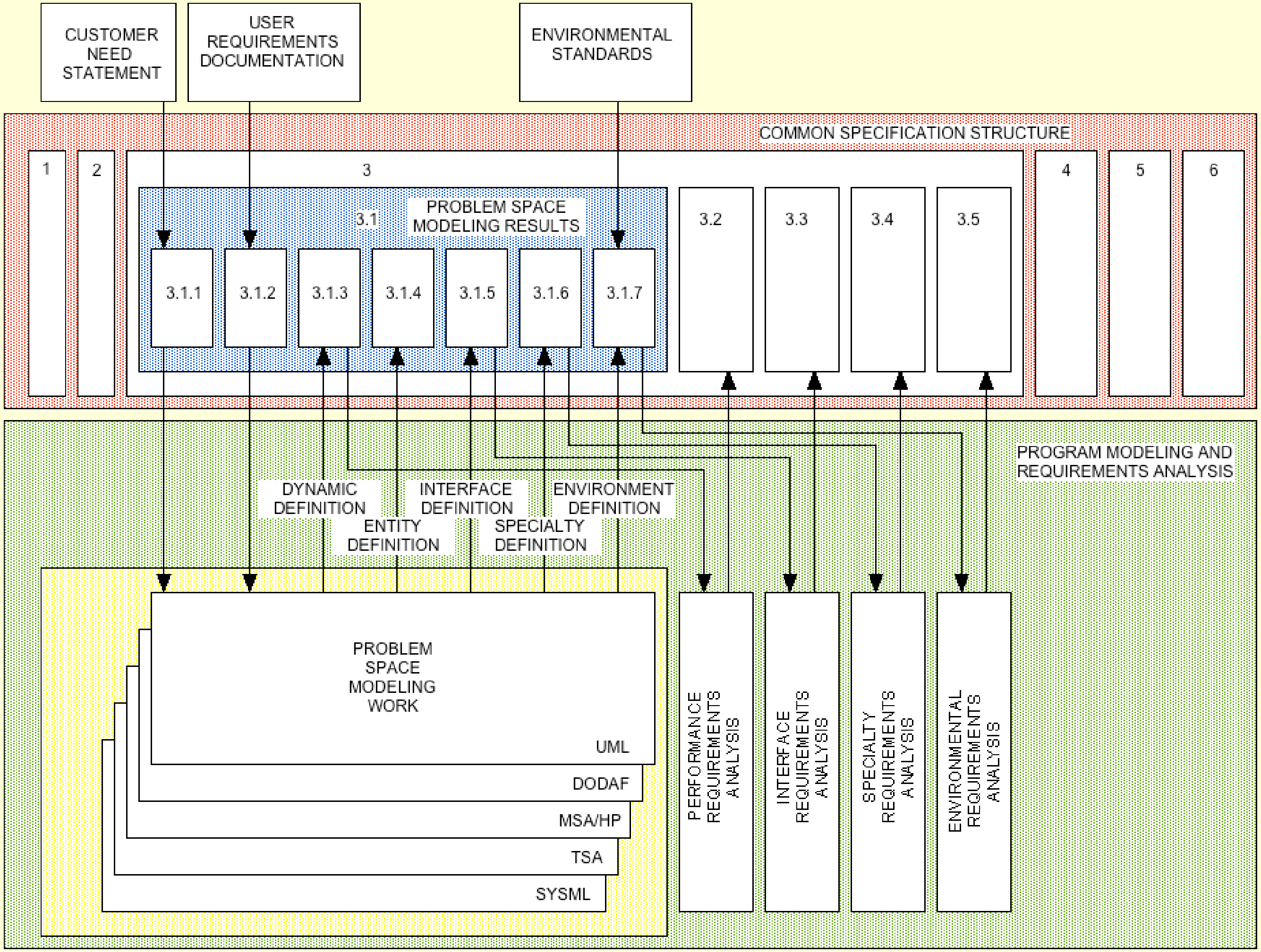
Model Results Flow Into Specifications Content Through the RAS



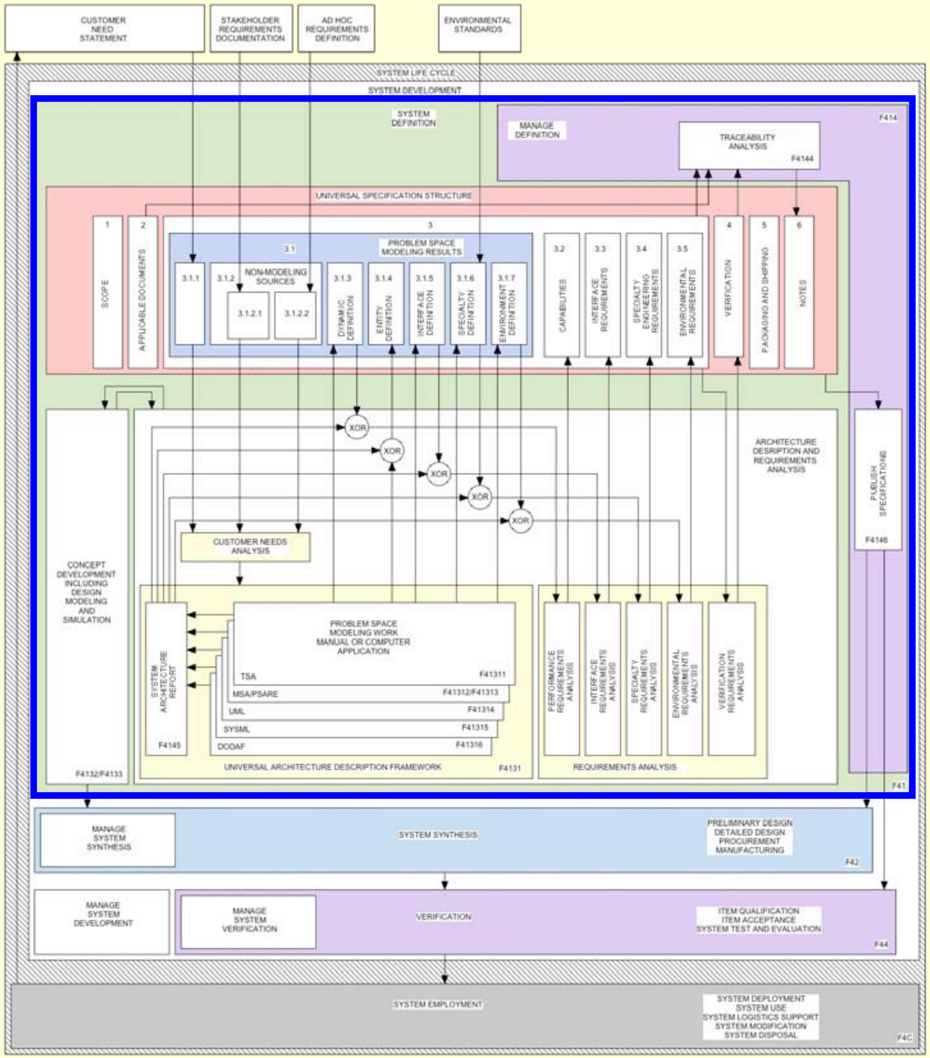
Universal Architecture Description Framework Approach



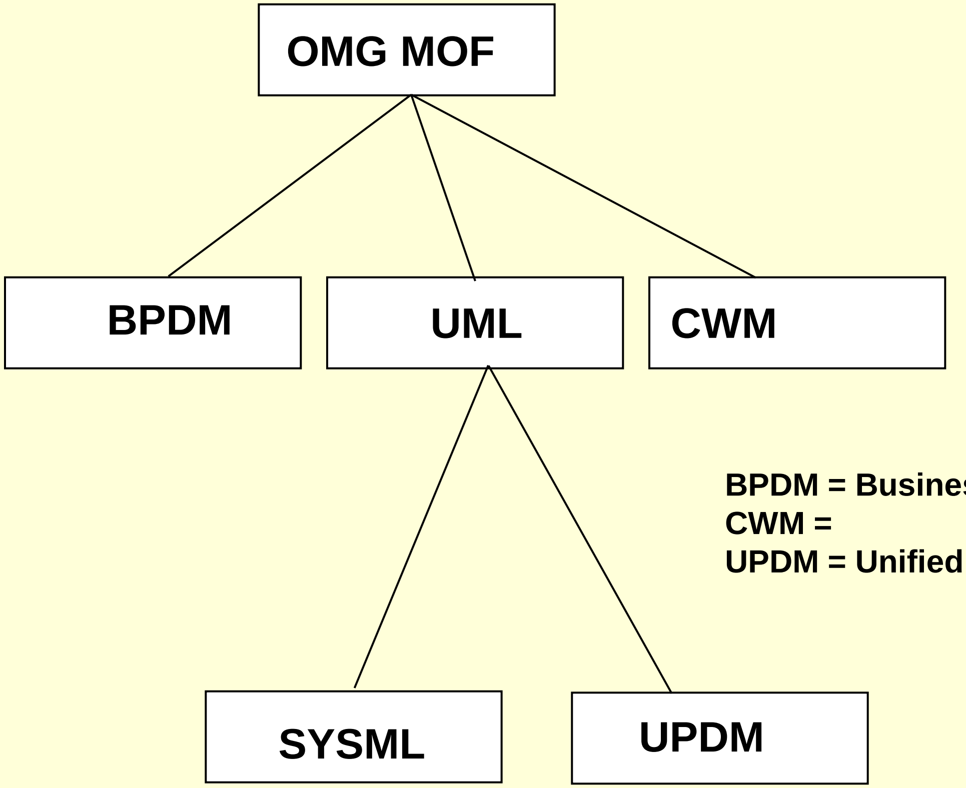
Building Universal Specifications



It Fits Into a Grander Structure



Model Convergence On the Road to Enterprise Architecting



BPDM = Business Process Data Model
CWM =
UPDM = Unified Profile For DODAF MoDAF

Action Items For You as a System Engineer

- 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 simplifying your combined set of models into a universal framework
- Work toward correlating the SW and HW development work patterns so as to encourage more effective integration
- Join INCOSE/NDIA working groups that deal with the issues covered in this tutorial and offer your ideas.

Tasks For a Development Organization

- Select a set of models for your UADF
- Train your people to apply those models to problem space
- Perfect inter-model traceability and integration skills as well as coordination of modeling and concept development work
- Insist on requirements being derived from models
- Apply a universal specification format
- Capture the work products that result from modeling work in a configuration manageable form
- Get on the tools treadmill