Massachusetts Institute of Technology

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Fundamentals of Systems Engineering

Prof. Olivier L. de Weck

Session 8

Systems Integration
Interface Management

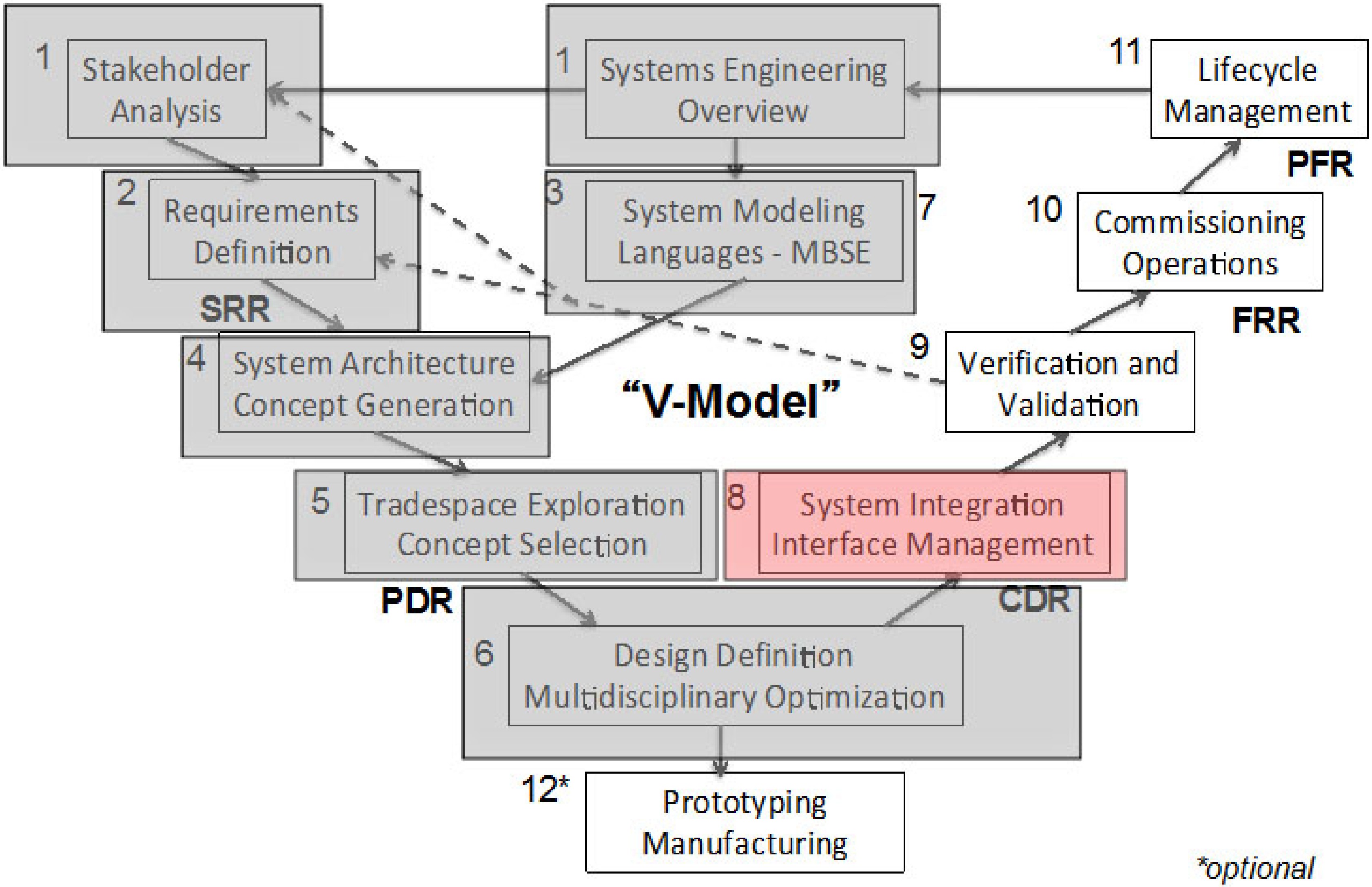
General Status Update

Assignment	*Topic*	*Weight*
A1 (group)	Team Formation, Definitions, Stakeholders, Concept of Operations (CONOPS)	12.5%
A2 (group)	Requirements Definition and Analysis Margins Allocation	12.5%
A3 (group)	System Architecture, Concept Generation	12.5%
A4 (group)	Tradespace Exploration, Concept Selection	12.5%
A5 (group)	Preliminary Design Review (PDR) Package and Presentation	20%
Quiz (individual)	Written online quiz	10%
Oral Exam (individual)	20' Oral Exam with Instructor 2-page reflective memorandum	10%

A4 is due today !

The "V-Model" of Systems Engineering

16.842/ENG-421 Fundamentals of Systems Engineering

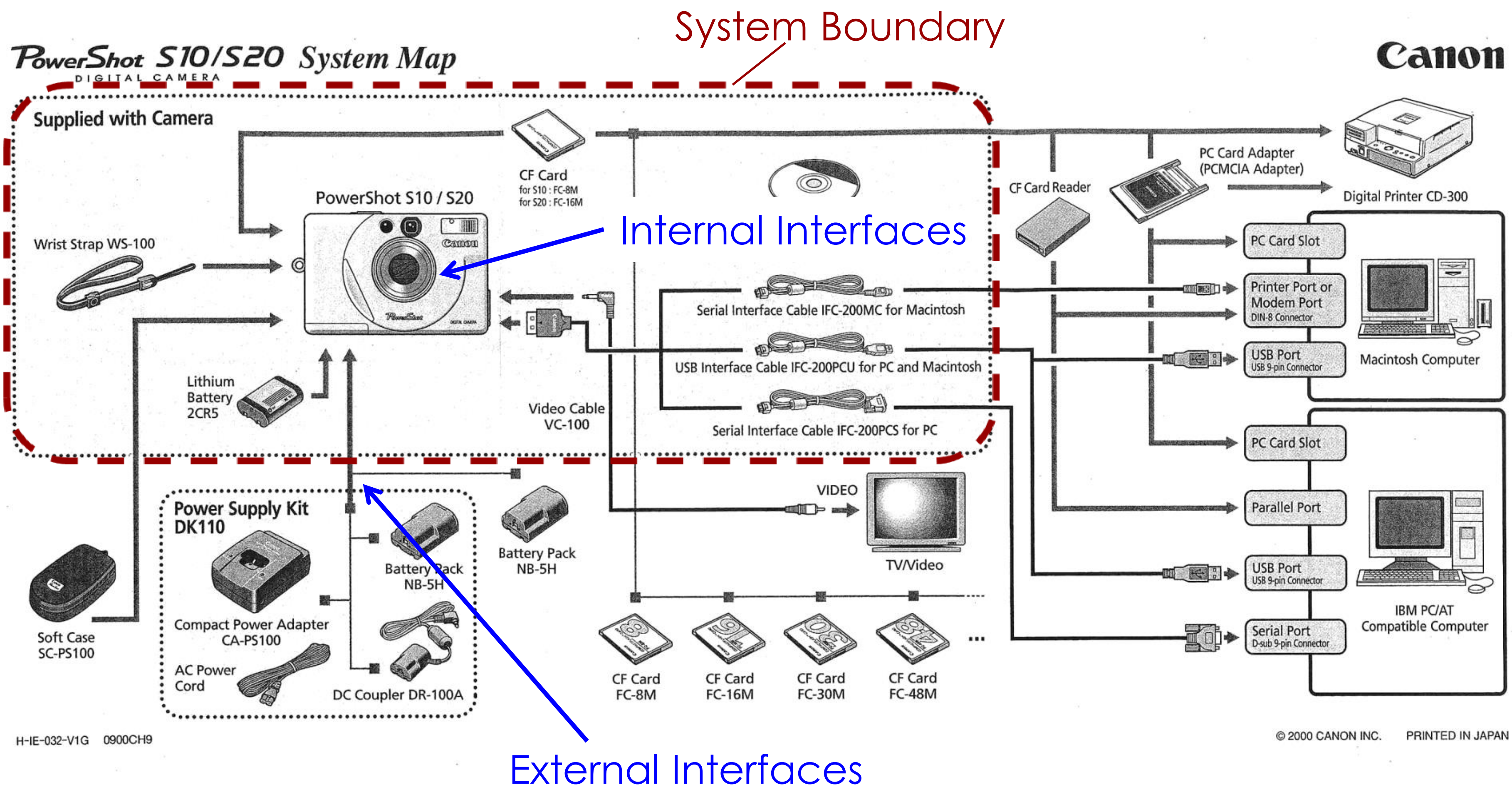


Numbers indicate the session # in this class

Outline

- Why is interface management important?
  - System failures due to interfaces
  - Working with partners and suppliers
- Interface Management
  - Types of Interfaces
  - Design Structure Matrix (DSM)
  - Interface Control Documents (ICD) – NASA Approach
- System Integration
  - Sequencing of Integration
  - Role of Standards

System Interfaces – internal vs. external



Source: Canon (Powershot S10)

Interface-induced Failures

- Much effort is spent on designing individual parts of a system
  - Functionality, tolerances, mean-time-between-failure (MTBF)
  - Interfaces are often neglected and can be the “weak points”
    - Bottlenecks, Structural failures, Erroneous function calls

Merging from side road to main road (Russia 2007)

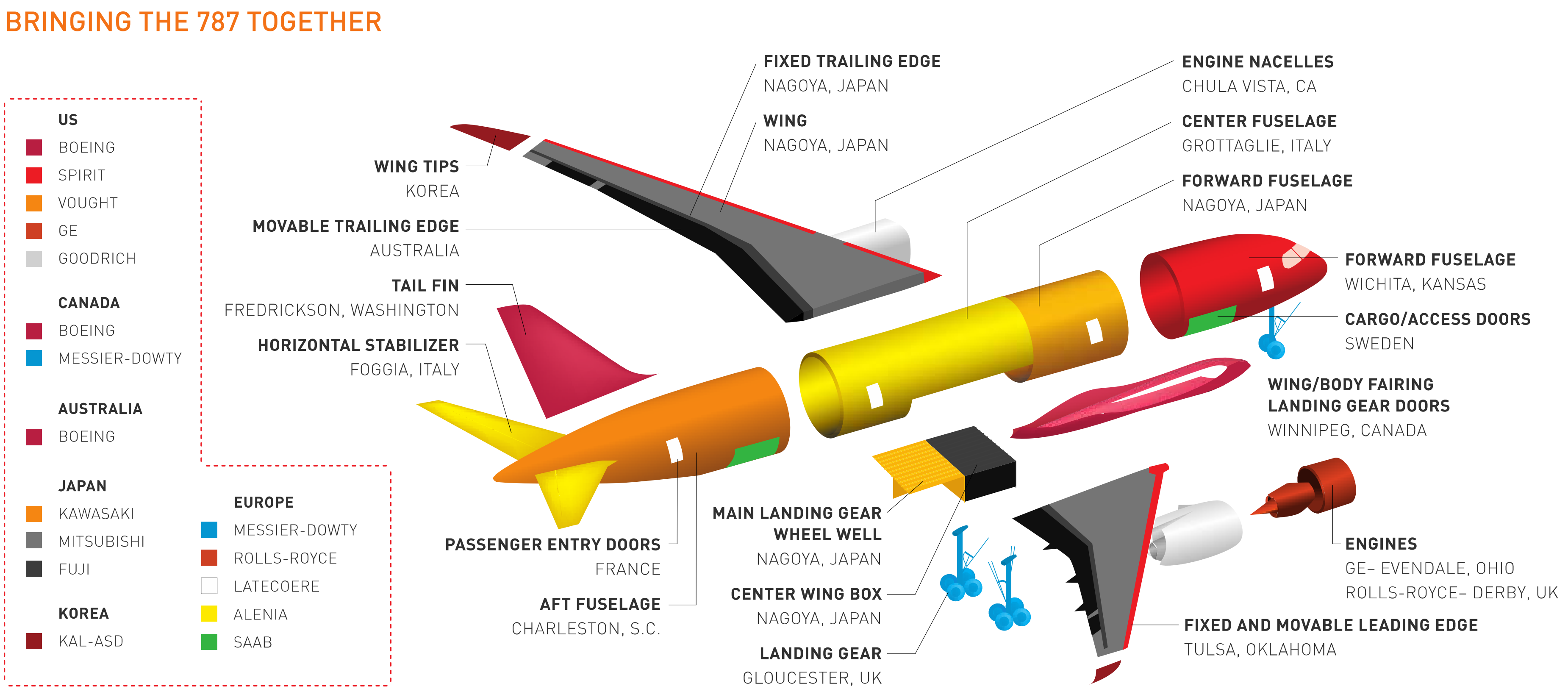
Torque Failure (MIT 2005)

Ariane 501
Accident report (1996)

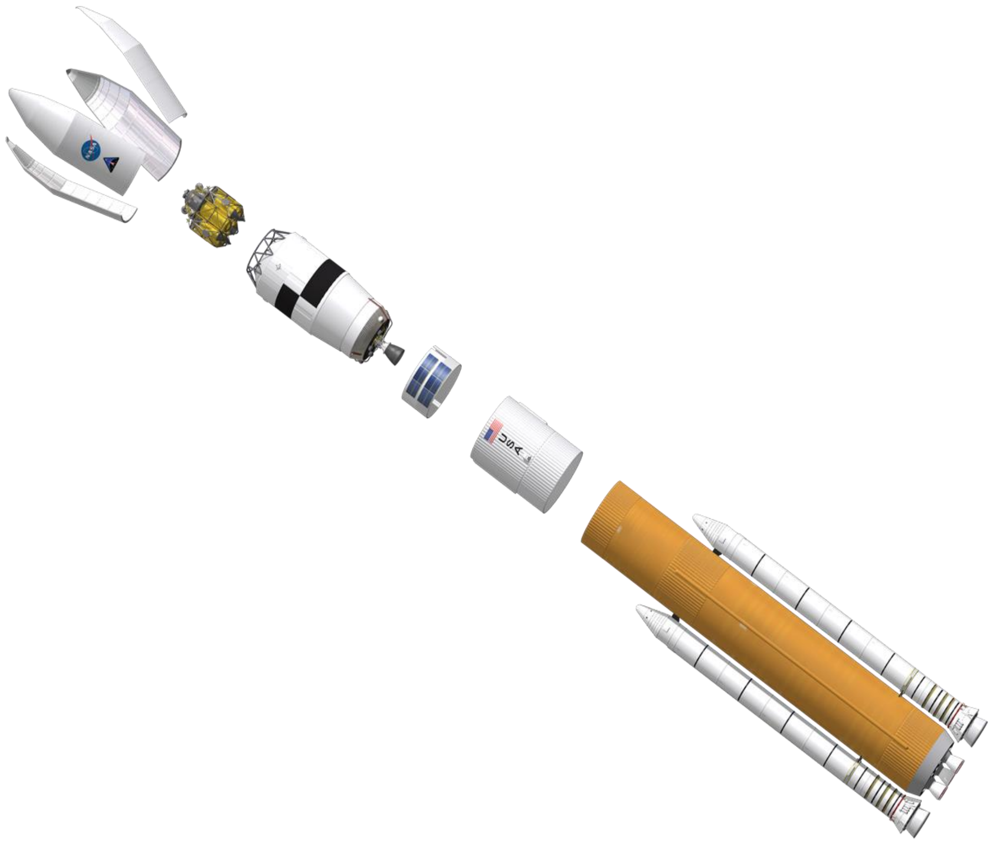
As a result of its failure, the active inertial reference system transmitted essentially diagnostic information to the launcher's main computer, where it was interpreted as flight data and used for flight control calculations. On the basis of those calculations the main computer commanded the booster nozzles, and somewhat later the main engine nozzle also, to make a large correction for an attitude deviation that had not occurred.

Working with Suppliers

Complex engineered systems are increasingly designed (and built) by geographically distributed teams, requiring careful definition of interfaces



Interface Management Importance



- Complex systems have many interfaces
  - Common interfaces reduce complexity
  - System architecture drives the types of interfaces to be utilized in the design process
  - Clear interface identification and definition reduces risk
  - Most of the problems in systems are at the interfaces.
  - Verification of all interfaces is critical for ensuring compatibility and operation

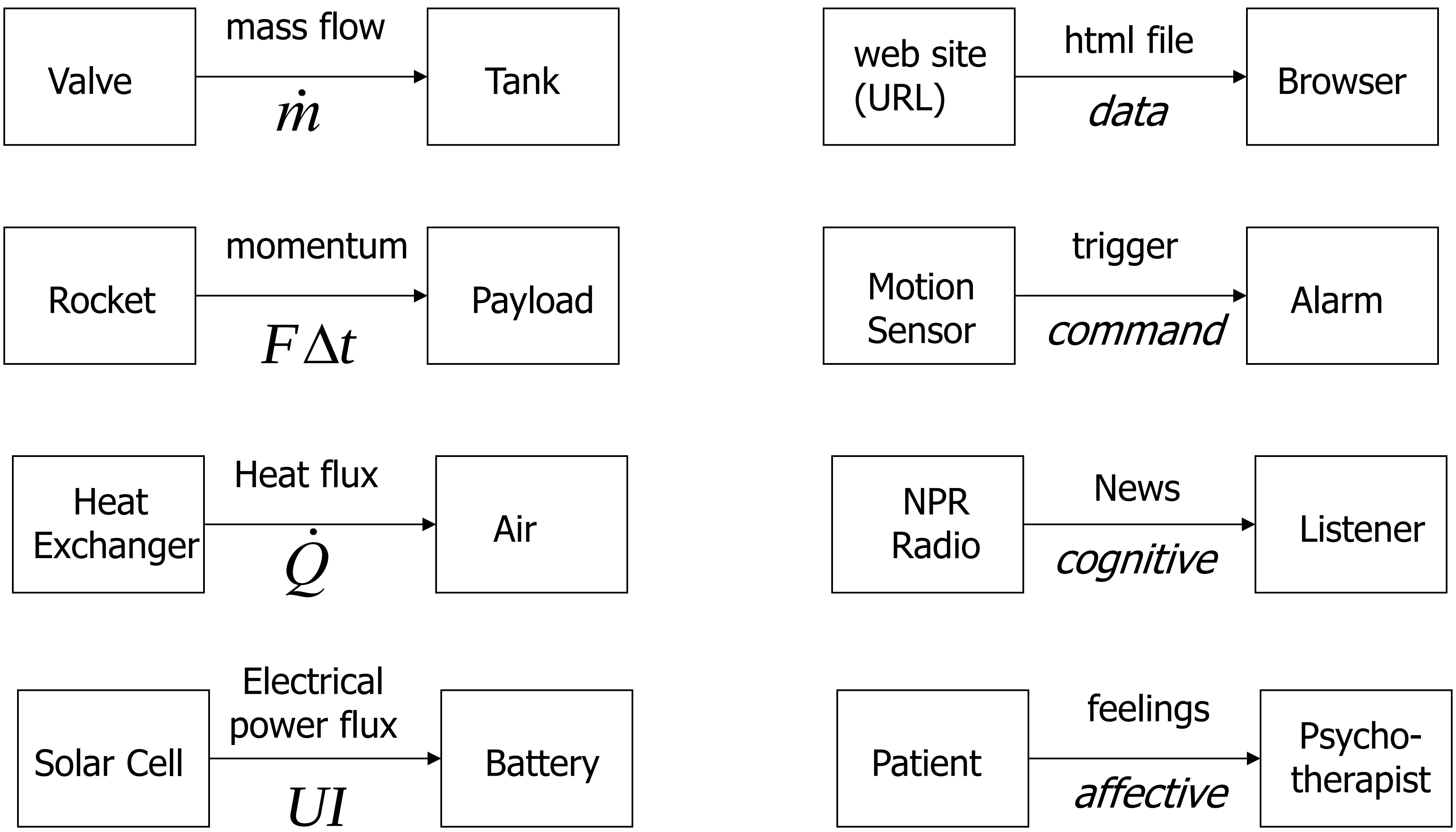
Experiences with Interfaces

- Turn to your partner exercise (5 min)
- **What was an instance in your past experience were carefully defining and managing an interface was critical?**
  - Discuss with your partner.
  - Share.

Outline

- Why is interface management important?
  - System failures due to interfaces
  - Working with partners and suppliers
- Interface Management
  - Types of Interfaces
  - Design Structure Matrix (DSM)
  - Interface Control Documents (ICD) – NASA Approach
- System Integration
  - Sequencing of Integration
  - Role of Standards

Examples of Interfaces



4 Canonical Types of Interfaces

- Physical Connection (always symmetric)
  - If A connects to B, B must also connect to A.
- Energy Flow
- Mass Flow
- Information Flow

Physical Connection

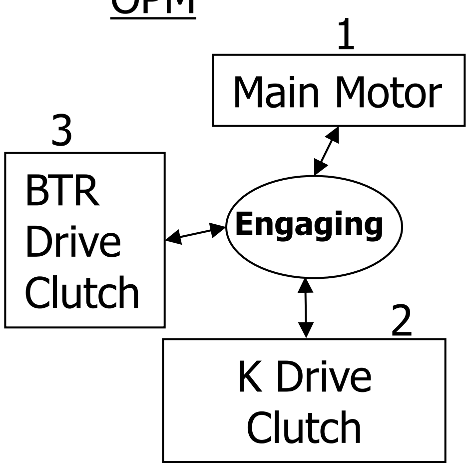
Two parts are in direct physical connection if they

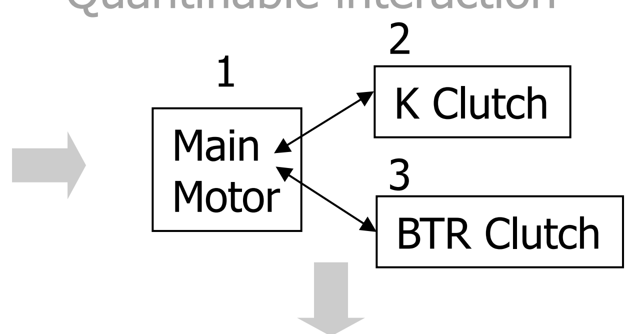
- touch each other
  - examples: rollers, brake pad & disk, finger & touchscreen
- have a reversible connection between them
  - examples: electrical connectors, USB port/cable, latch mechanism, bolts & nuts
- are permanently connected to each other
  - examples: rivets, spot-welded, fusing, compiling (?)

Quantifiable interaction

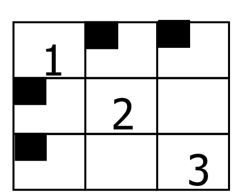
- Force [N], Torque [Nm]

<u>OPM</u>





<u>DSM</u>



<u>Important Note</u>:
physical connection implies symmetric entries in the DSM

(action=reaction)

Examples of Physical Connection

Irreversible structural links

welding

Main.f → Main.o

Sub1.f → Sub1.o

Sub2.f → Sub2.o

Main.o, Sub1.o, Sub2.o → Main.exe

compiling

Reversible structural links

Strut 1

bolting

Strut 2

RJ-45 jack

connecting

plug

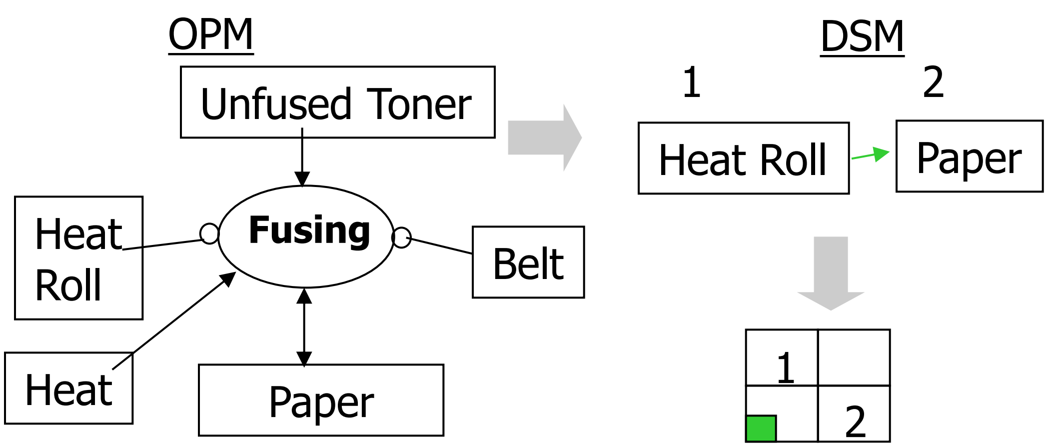
Energy Flow

- Energy Flow is present if there is a net exchange of work between two components
  - *Power = dW/dt [J/s=W]*
- Can take on different forms
  - Electrical Power (most common in products)
    - DC Power (12V, 5V, 24V,…), Power = Current * Voltage
    - AC Power (120 V 60Hz, 220V 50Hz, …)
  - Thermal Power
    - Heat flux: dQ/dt
    - Conduction, Convection, Radiation
  - RF Power
    - Microwaves (2.4 GHz, 5.8 GHz,…)
  - Mechanical Power
    - Linear: Power = Force * velocity
    - Rotary: Power = Torque * angular rate
- Energy Flow typically usually implies a physical connection (but not always !)
  - Wires, conducting surface

Energy Flow (cont.)

Energy Flow is typically directed

- from source to sink



Heat Energy is transferred from system 1 to system 2

Important Note: typically we first map the desired interactions, later as we know more also the undesired ones (e.g. waste heat flux)

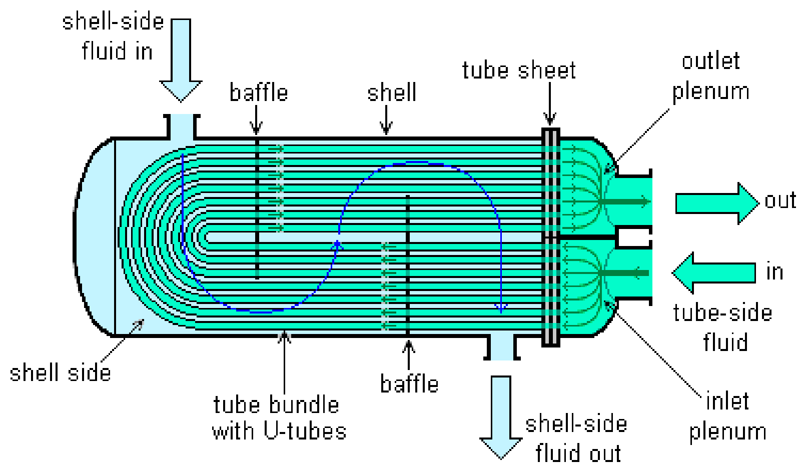
Mass Flow

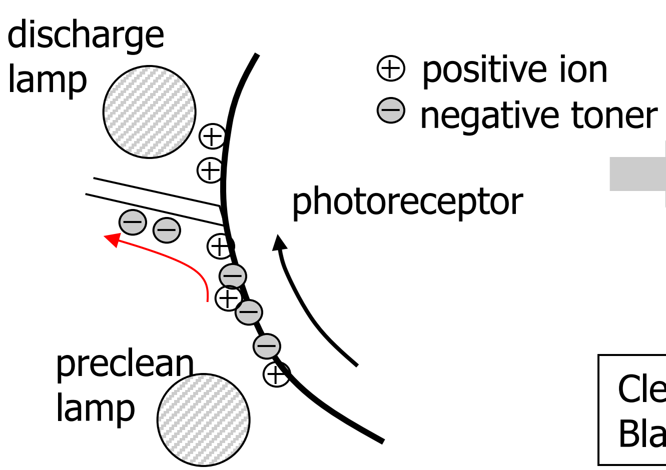
- Mass Flow implies that matter is being exchanged between two elements (or subsystems)
  - *mass flow = dm/dt [kg/sec]*
  - Fluids
    - cooling liquid (refrigerant), fuel, water, …
  - Gases
    - air, exhaust gas, …
  - Solids
    - toner, paper (media in general),…
  - Typically implies an underlying physical connection
- Mass flow is typically directed
  - from source to sink
  - can form a continuous loop

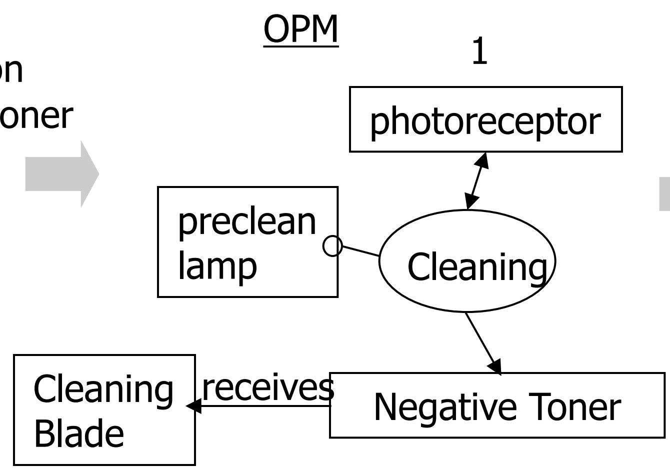
Mass Flow (cont.)

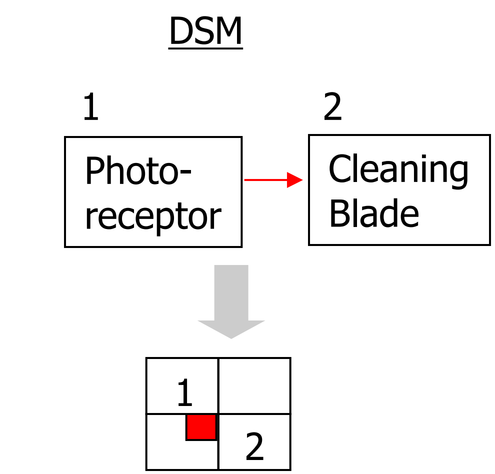
Examples

U-tube heat exchanger









Mass is transferred from system 1 to system 2

Information Flow

Many modern electro-mechanical systems have replaced functions previously implemented with mechanical elements in software

Required for Interactions with the user/operator

- GUI, I/O

Required for interactions with other devices

- Analog (ADC, DAC), Digital (DIO), Wireless (e.g. IEEE 802.11)

Required for internal device controls

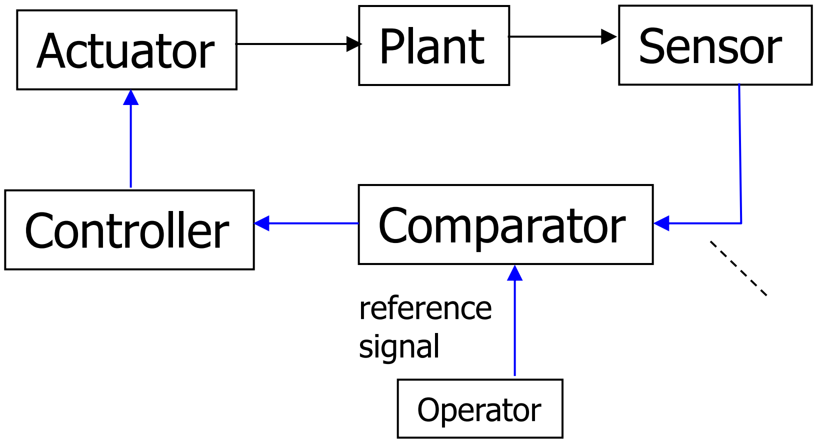
- Sensors
- Actuators
- Controllers
- Filters, Amplifiers, …

Information flow is always directed

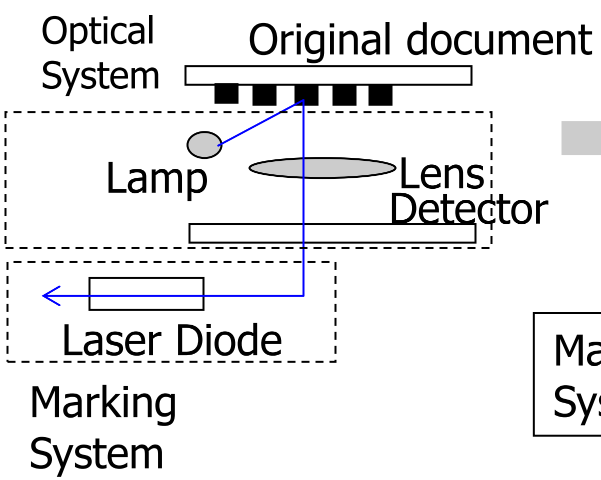
- Telemetry (sensor data) … how is my system doing?
- Command data …this is what I want my system to do

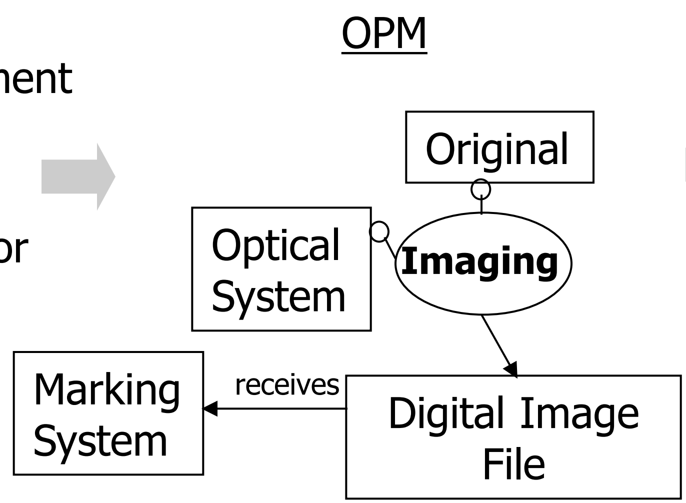
Information Flow (cont.)

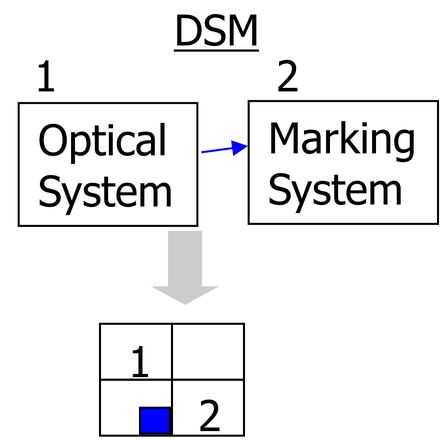
Control Loop



Xerography: Imaging





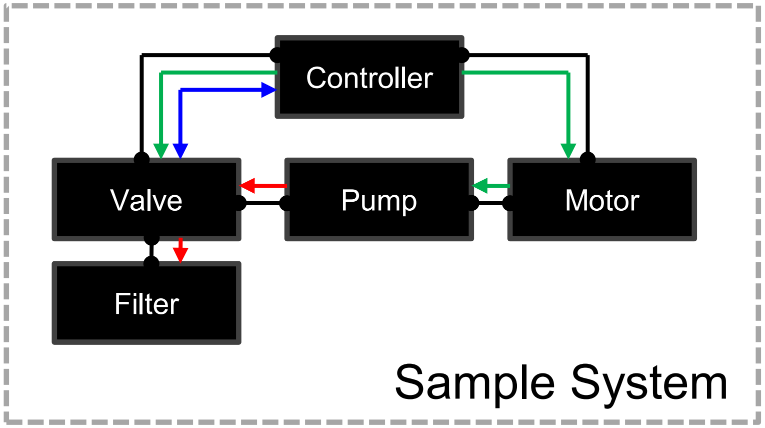


Information is transferred
from system 1 to system 2

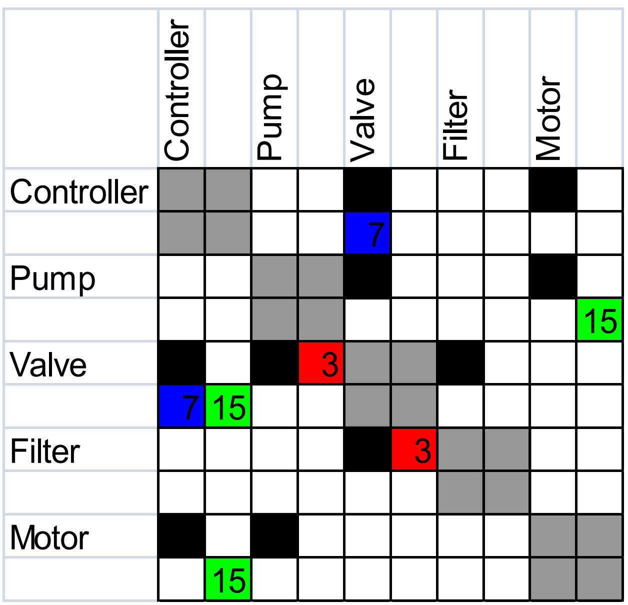
Outline

- Why is interface management important?
  - System failures due to interfaces
  - Working with partners and suppliers
- Interface Management
  - Types of Interfaces
  - Design Structure Matrix (DSM)
  - Interface Control Documents (ICD) – NASA Approach
- System Integration
  - Sequencing of Integration
  - Role of Standards

DSM Captures Form Connectivity Architecture



Architecture Definition:
The embodiment of concept, and the allocation of physical/informational function (process) to elements of form (objects) and definition of structural interfaces among the objects



DSM

Number	Type	Flag
0	No Connection	0
1	Mechanical	
2	Flow	3
3	Information	7
4	Energy	15

Key

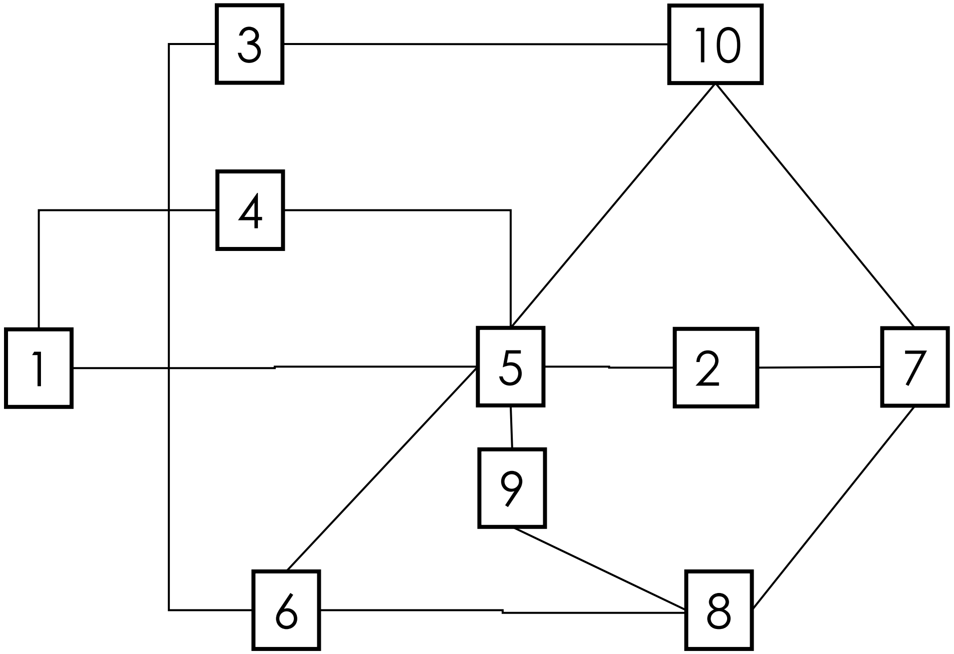
- DSM captures connectivity of components => architecture
- DSM provides analysis capability not present in a traditional schematic

Background

- Design Structure Matrix (DSM)
  - Synonyms (a.k.a.)
  - Design Dependency Matrix
  - N^2-Matrix, N^2-Diagram
  - Adjacency Matrix (Graph Theory)
- Based on Fundamental Work by Don Stewart and Steve Eppinger
  - Donald V. Steward, Aug. 1981, IEEE Transactions on Engineering Management
  - Steve Eppinger et al., 1994, Research in Engineering Design
- Matrix Representation of Product Architecture
  - Most literature simply uses “binary” DSMs
  - Need more process related details to connect to physics

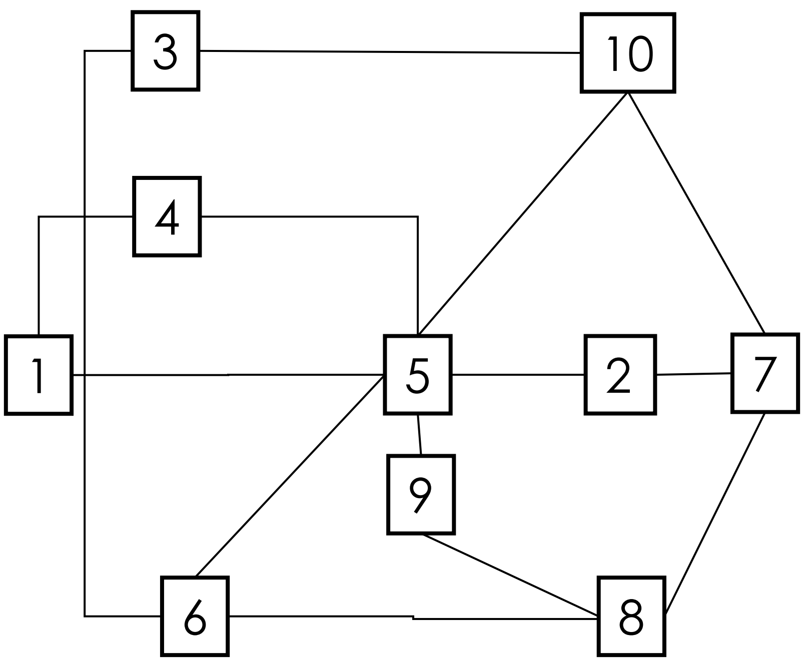
Refrigerator Liaison Diagram (Level 1)

1. Door
2. Condenser
3. Power Supply
4. Hinge
5. Cabinet
6. Thermostat
7. Refrigerant
8. Evaporator
9. Freezer Door
10. Compressor



Key Idea → Map to Matrix

- Matrix Representation



Liaison Diagram
Kenmore Refrigerator 2.5 cu ft

1			X	X					
	2			X		X			
		3			X				X
X			4	X					
X	X		X	5	X			X	X
		X		X	6		X		
	X					7	X		X
					X	X	8	X	
				X			X	9	
		X		X		X			10

What types of interfaces exist?

Classes of Object-Process-Operand Links

Object-based DSM

- "chunks" of product on main diagonal
- interconnections:
  - physical
  - energy
  - mass flow
  - information

Process Link Class	Operand	Process	Instrument Form
Physical Connection	Forces, Torques [N, Nm]	Force or Torque Transmitting	bolts, washers, rivets, spot welds…
Energy Flow	Work [J]	Electricity or Heat Transmitting	copper wires, microwaves, …
Mass Flow	Mass [kg]	Fluid, Gas or Solid Matter Transmitting	fuel lines, air ducts, exhaust pipes …
Information Flow	Bits [-]	Data or Command Transmitting	micro-switches, wireless RF, humans

Note: In many cases, in order for an energy, mass or information flow to exist, there also needs to be a physical connection, but not always

Refrigerator Level -1 DSM

Physical Connections
Mass Flows
Energy Flows
Information Flows

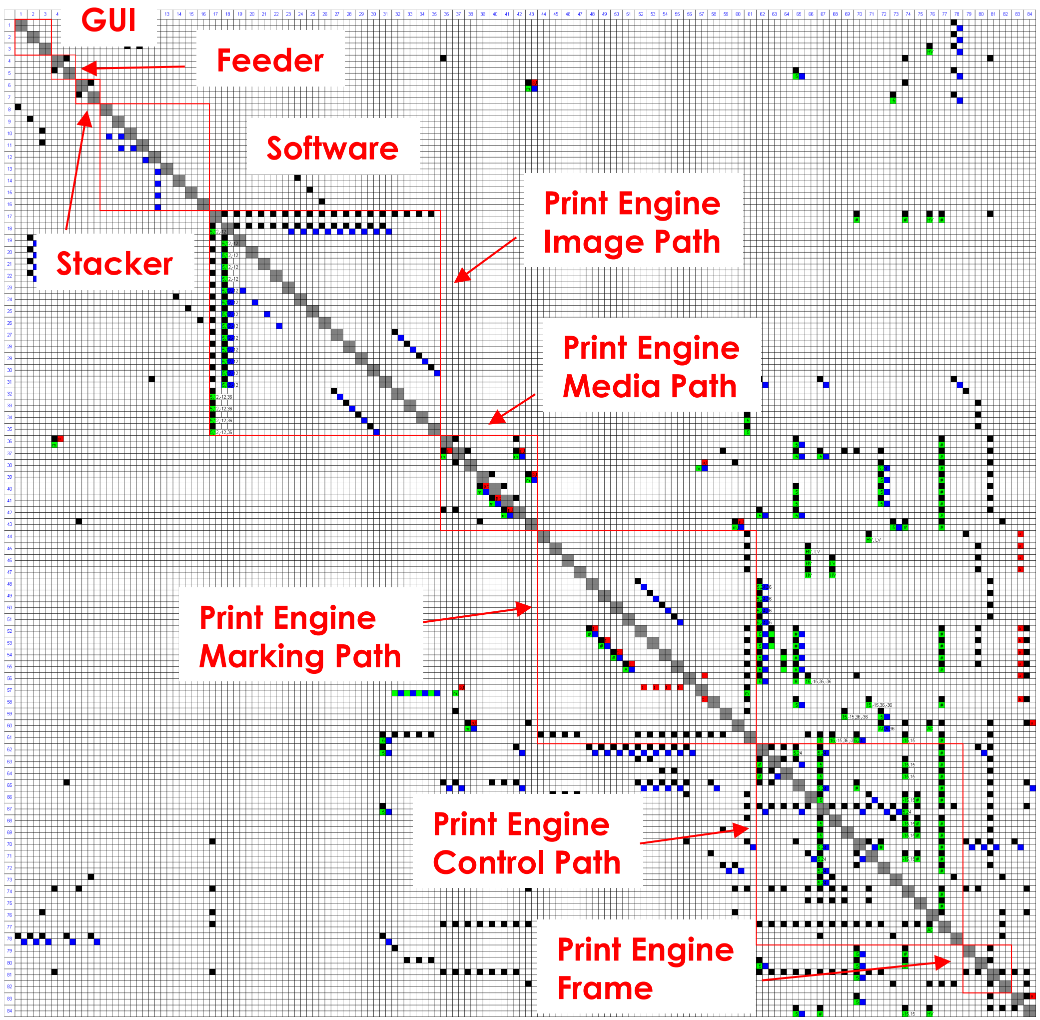
		1 Door		2 Condenser		3 Power Supply		4 Hinge		5 Cabinet		6 Thermostat		7 Refrigerant		8 Evaporator		9 Freezer Door		10 Compressor	
1	Door							1		1											
2	Condenser									1				1							2
																				3	
3	Power Supply											1								1	
													4								
4	Hinge	1								1											
5	Cabinet	1		1				1				1						1		1	
6	Thermostat					1				1						1					
						3															
7	Refrigerant			1	2											1	2			1	2
				3												3				3	
8	Evaporator				2							1		1				1			
				3																	
9	Freezer Door									1						1				1	
10	Compressor					1				1				1		1	2				
						3										3					

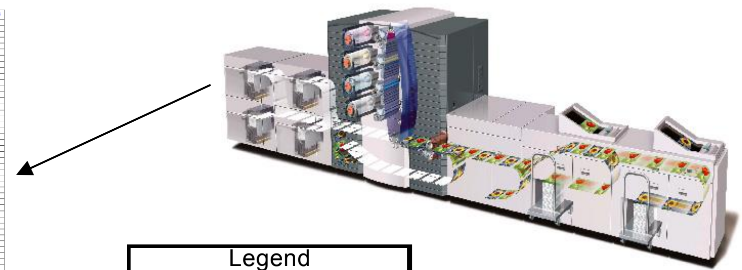
General Process for Generating a DSM

- **Top Down (design and architecting)**
  - Generate System OPM
  - Hide attributes and states
  - Collapse all processes into "tagged" structural links
  - Generate DSM

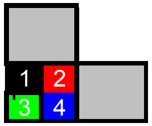
- **Bottom-Up (reverse engineering)**
  - Select system/product to be modeled
  - Perform product dissection
  - Carefully document the following:
    - Parts List/Bill of Materials
    - Liaison Diagram (shows physical connections)
  - Infer other connections based on reverse engineering/knowledge of functions:
    - mass flow, energy flow, info flow
  - Manipulate DSM
    - clustering

Xerox iGen3 Baseline Design Structure Matrix (DSM)





Legend	
1	Physical connection
2	Mass flow
3	Energy flow
4	Information flow



Key

p	Paper
t	Toner
a	air (purified / ready for use)
o	Ozone
d	Dirt
HV	High Voltage
LV	Low Voltage
5, …	DC Voltage
m	Mechanical energy (translation, rotation, etc…)
h	Heat energy (Fuser only)

Base iGen DSM

Total number of DSM Elements	84
Total number of physical connections	572
Total number of mass flow connections	45
Total number of energy flow connections	167
Total number of information flow connections	165
Number of Base DSM cells	27972
Number of non-empty cells	1033
Sparsity (Nonzero Fraction NZF)	0.037

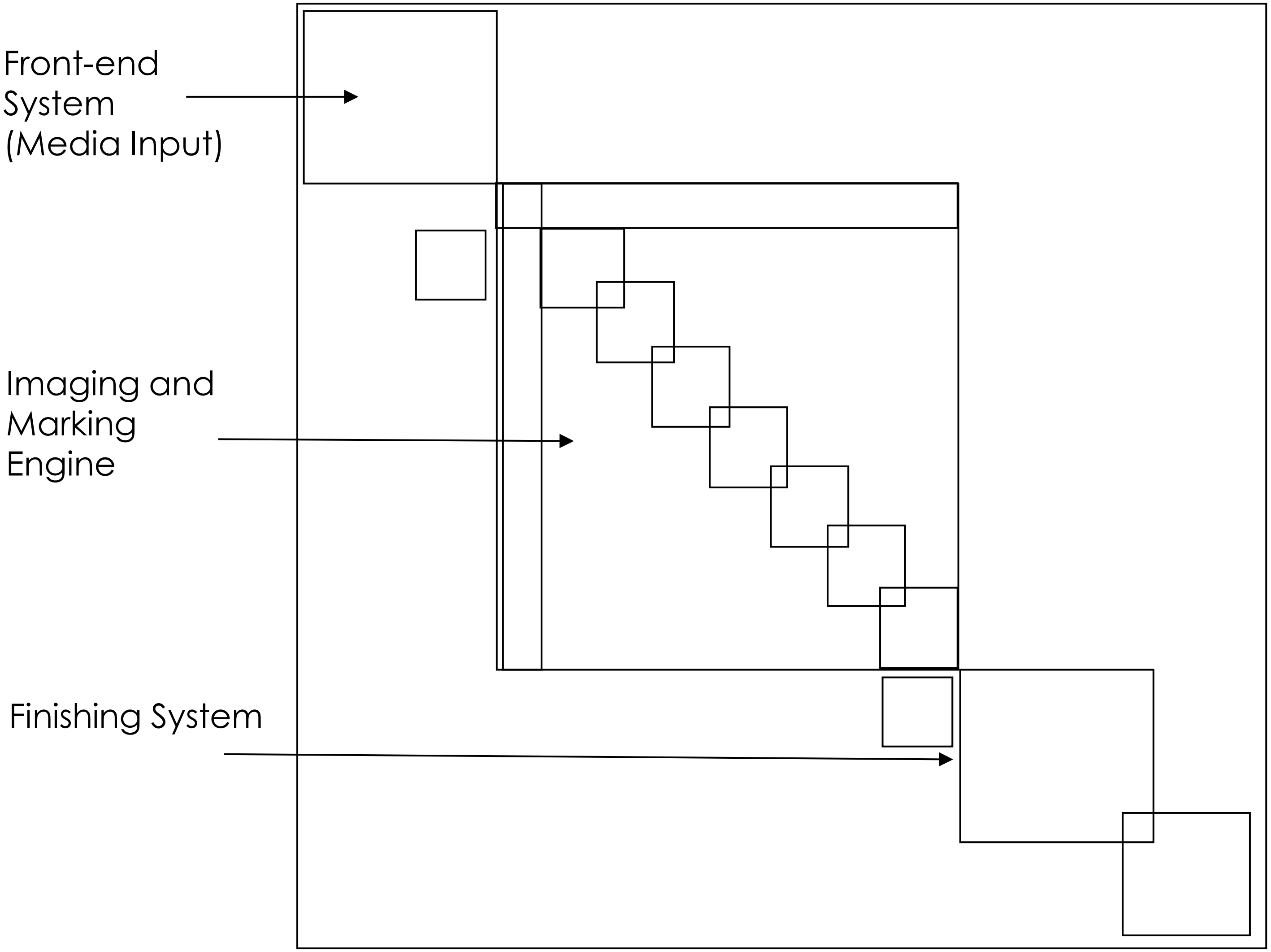
Suh. E.S., Furst M.R., Mihalyov K.J, de Weck O., "Technology Infusion for Complex Systems: A Framework and Case Study", Systems Engineering, 13 (2), 186-203, Summer 2010

Detailed Reverse Engineering DSM Procedure

1. Define which product or system to model
2. Assemble product documentation
3. Create product breakdown structure
4. Start a blank DSM spreadsheet for example in Excel
5. Label the rows and columns of the DSM with both an ID number 1 …N and a component/subsystem name
6. Start by mapping all the physical connections in the system
7. Double-check the physical connections
8. Map out mass flows along physical connections
9. Double-check mass flows from start to origin
10. Map out the energy flows along physical connections
11. Double-check energy flows from start to finish
12. Map out information flows following physical connections
13. Double-check information flows in the system
14. Map interactions (flows) in the system that do not follow physical connections
15. Reorder the DSM to reveal "modules"
16. Double-check accuracy of DSM
17. Sign off and publish

Suh E.S., de Weck O.L., Furst M., Mihalyov K., "Estimating the Impact of Technology Infusion", Xerox Technical Report, Accession Number X200700206, Xerox Corporation, January 2008

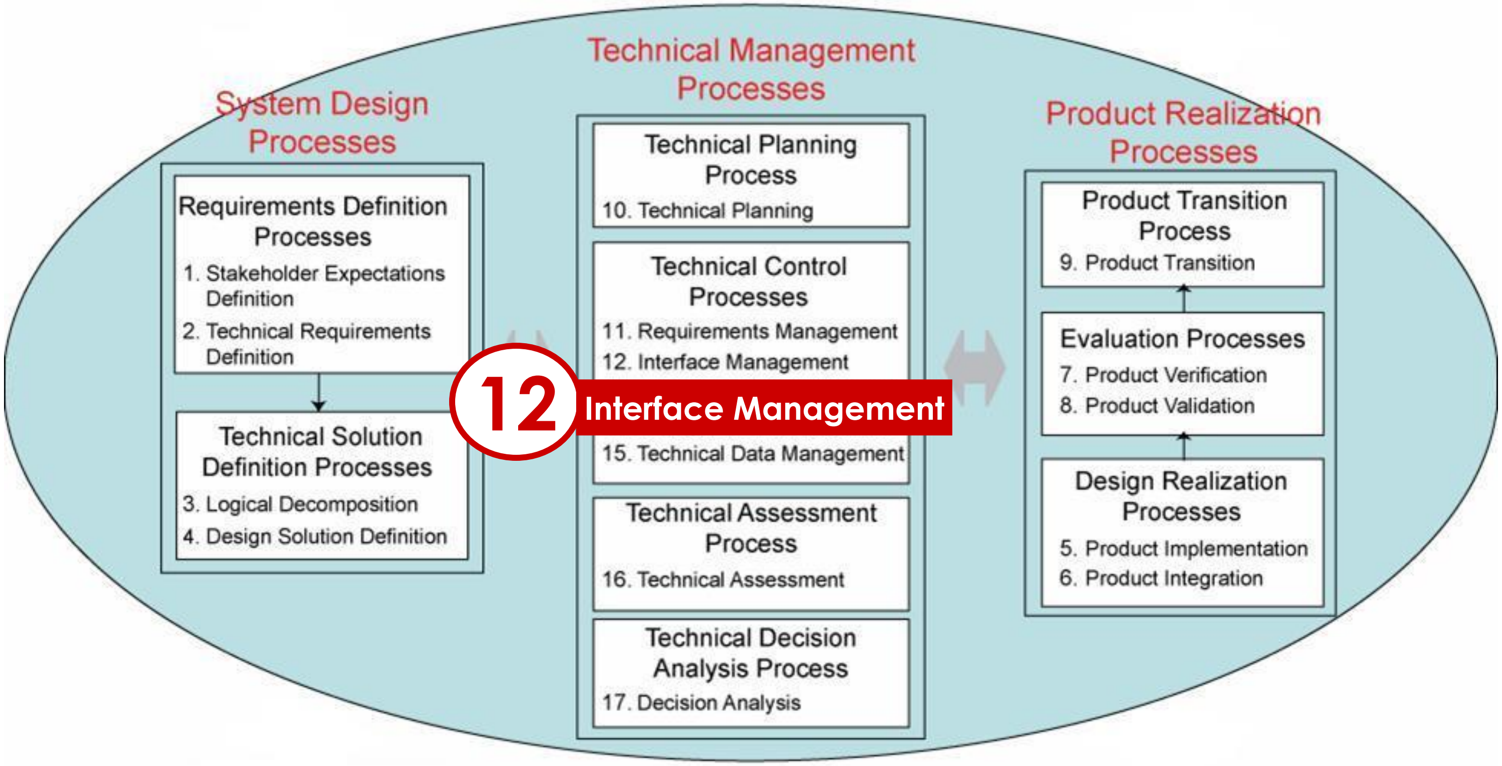
High Level Product Architecture (Xerox)



Outline

- Why is interface management important?
  - System failures due to interfaces
  - Working with partners and suppliers
- Interface Management
  - Types of Interfaces
  - Design Structure Matrix (DSM)
  - Interface Control Documents (ICD) – NASA Approach
- System Integration
  - Sequencing of Integration
  - Role of Standards

Interface Management Process



NASA Systems Engineering Handbook

Interface Management Process Purpose

The Interface Management Process is used to:

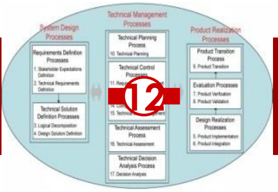
- Establish and use formal interface management to assist in controlling system product development efforts especially when the efforts are divided between government programs, contractors, and/or geographically diverse technical teams within the same program or project

- Maintain interface definition and compliance among the end products and enabling products that compose the system as well as with other systems with which the end products and enabling products must interoperate.

Key Interface Documentation

- **Interface Requirements Document** (IRD) - Defines the functional, performance, electrical, environmental, human, and physical requirements and constraints that exist at a common boundary between two or more functions, system elements, configuration items, or systems.
  - Interface requirements include both logical and physical interfaces.
- **Interface Control Document or Interface Control Drawing** (ICD) - Details the physical interface between two system elements, including the number and types of connectors, electrical parameters, mechanical properties, and environmental constraints.
  - The ICD identifies the design solution to the interface requirement.
  - ICDs are useful when separate organizations are developing design solutions to be adhered to at a particular interface.
- **Interface Definition Document** (IDD) - A unilateral document controlled by the end item provider, and provides the details of the interface for a design solution that is already established.
  - This document is sometimes referred to as a “one-sided ICD.”
  - The user must then design the interface of the system to be compatible with the already existing design interface.

Interface Management Process



Activities

- Prepare or update interface management procedures
- Conduct interface management during system design activities for each WBS-like model in the system structure
- Conduct interface management during product integration activities
- Conduct interface control
- Capture work products from interface management activities

Input

From user or program and system design processes

- Interface requirements

From project and **Technical Assessment Process**

- Interface changes

Output

To **Configuration Management Process**

- Interface control documents
- Approved interface requirement changes

To **Technical Data Management Process**

- Interface management work products

Outline

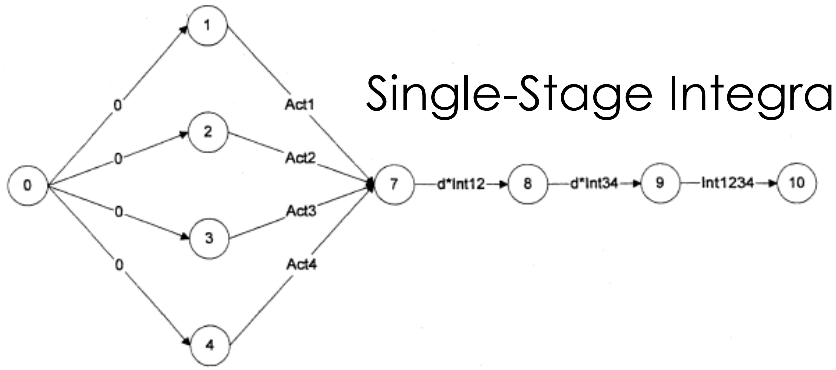
- Why is interface management important?
  - System failures due to interfaces
  - Working with partners and suppliers
- Interface Management
  - Types of Interfaces
  - Design Structure Matrix (DSM)
  - Interface Control Documents (ICD) – NASA Approach
- System Integration
  - Sequencing of Integration
  - Role of Standards

System Integration

- **System integration is the process of deliberate assembly of the parts of the system into a functioning whole**
  - Physical assembly of parts
  - Connecting different conduits, hoses
    - Filling in various kinds of consumables
  - Connecting electronics to power sources, avionics etc… (often with wire harnesses)
  - Uploading of test and operational software
- The sequence in which integration occurs may be important (see paper by Ben-Asher et al.)
- In complex systems many errors are only discovered during system integration and test

Systems Integration Sequencing

Single-Stage Integration



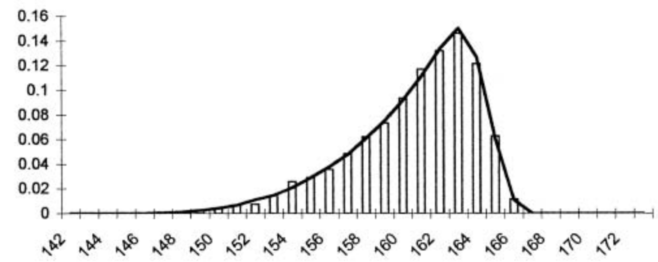
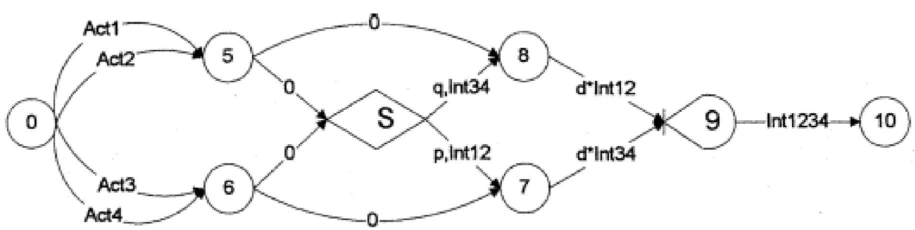


Figure 5. Total time for Case 1 with $d = 0.8$—single integration process.

Incremental Integration



q=1-p

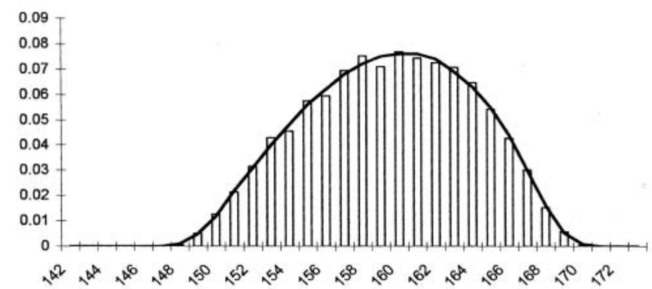


Figure 6. Total time for Case 1 with $d = 0.8$—Incremental integration.

- Integration Sequence can be optimized
  - All-in-one vs. incremental integration
  - Expected time duration distribution for system integration depends on sequence and uncertainty

Figure 1. Case study: Mini-UAV with a plug-in optronics payload (TAMAM-IAI).

1. Engine system
2. Flight control system
3. Optronic payload
4. Payload control system

Ref: Tahan and Ben-Asher [2004]

Concept Question 8: Role of Standards

- Industry Standards are established to simplify interface management. ***Which of the following standards are you familiar with?***

  - IEEE 802.11g
  - MIL-STD-1553
  - RS-232C
  - Bba_J23100
  - Others

- Answer Concept Question 8 (see supplemental files)

Summary Lecture SE6

Why is interface management important?

- System failures due to interfaces
- Working with partners and suppliers

Interface Management

- Types of Interfaces
- Design Structure Matrix (DSM) as a
- Interface Control Documents (ICD) – NASA Approach
  - Many other organizations use ICDs in one way or another

System Integration

- Needs to be carefully planned
- Sequence of integration may be very important, area of active research in Systems Engineering
- Industry Standards for Interfaces are critical