



Interfaces Module

Space Systems Engineering, version 1.0

Module Purpose: Interfaces

- ◆ Define interfaces, how they are described and why it is important that they be managed.
- ◆ Describe the most common interface tool, the N-squared diagram.
- ◆ Show how N-squared diagrams can be used to:
 - Capture the existence and nature of an interface
 - Highlight input and output assumptions and requirements
 - Demonstrate where there are feedback loops between subsystems
 - Identify candidate functional allocations to subsystems

External Versus Internal Interfaces

- ◆ System level external interface requirements are established with the other system level requirements.
- ◆ So, external interface requirements, like other system functional and performance requirements, are distributed to subsystems via allocation, derivation or flow down.
- ◆ Internal interface requirements are different since they are created as part of system decomposition. Different system solutions, i.e., different requirements distributions will create different subsystems and different subsystem interfaces.
 - This is a powerful opportunity for the development team to optimize the system design.

Example External and Internal Interfaces

- ◆ The interface between a spacecraft and its launch vehicle is an *example of an external interface* for the spacecraft development. The launch vehicle already exists so the spacecraft is developed to accommodate the existing (electrical, data, thermal and mechanical - including vibration and acoustic) interface.
- ◆ For an Earth remote sensing space-based radar system, the interface between the radar sensor subsystem and the spacecraft command and data handling subsystem is an *internal interface example*. The development team has the flexibility to determine the mechanical, electrical, thermal and data interfaces between these two subsystems.
 - So system functional and resource allocations to subsystems have implications for interface requirements. As an example, consider data compression. This function could be allocated to either the radar subsystem, the command and data handling subsystem or even the communications subsystem. Where the function is done effects the data interface (rate and format) of the subsystems down stream.
- ◆ So, optimal system decomposition considers both the implications for subsystem requirements and the implications for subsystem interfaces.

Need for Managing Internal Interfaces

- ◆ System decomposition (via requirements and functional analysis) creates a set of subsystems AND the interfaces between them.
- ◆ 'Ownership' of each subsystem is usually obvious as it is assigned to an individual or group as it is defined.
- ◆ But the subsystem interfaces need special attention since each subsystem owner may assume:
 - the other party is responsible for the interface, or
 - each party makes its own assumptions about the interface requirements.
- ◆ In both cases, the project runs the risk of future interface incompatibility, since the subsystems mature their concepts (and their interfaces) independently.
- ◆ The solution is to explicitly identify the owner of every interface.

Example Interface Requirements Between the Science Instrument and Spacecraft for GLAST

Electrical

3.2.4.1 Bus Voltage The bus voltage supplied to the Science Instrument shall be 28 V +/- 6 V as seen at the input terminals of the Science Instrument.

Data

3.2.6.1.1 Interface Data Rates The Science Instrument to Spacecraft data rate shall be no greater than 70 Mbps.

Mechanical

3.2.2.2 Mass Constraint The maximum launch mass of the Science Instrument shall be constrained to 3000 kg. This is exclusive of the instrument interface structure but inclusive of any Science Instrument hardware mounted on the Spacecraft bus, such as thermal radiators and electronics boxes.

Thermal

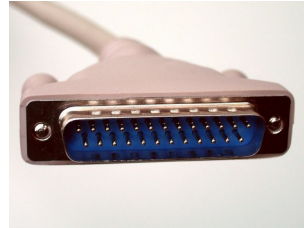
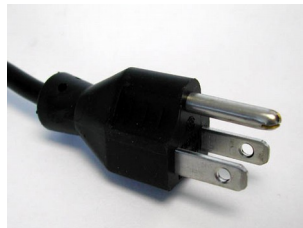
3.2.3.4.1 Interface Temperature Ranges The interface temperature of the Science Instrument electronics boxes shall be controlled to -10 C to +40 C (TBR) operating, and -55 C to +60 C (TBR) survival.



**GLAST —
Gamma-ray Large Area
Space Telescope
Launched 6/11/08**

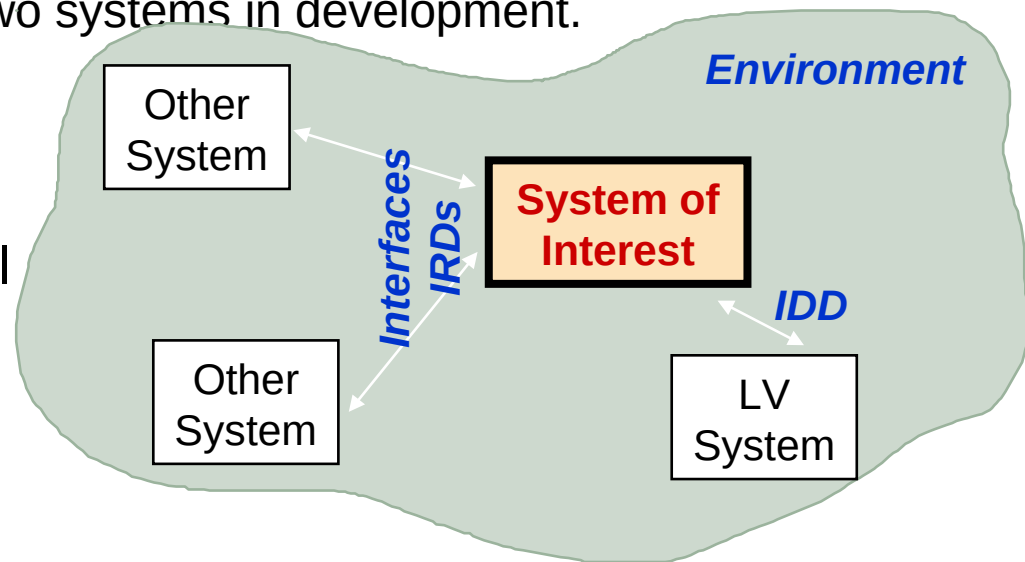
Interface Design Heuristics

- ◆ In system decomposition, choose subsystems so that they are as independent as possible; that is, they have simple interfaces. As a result these subsystems will be easier to develop and test independently and when they are integrated there will be fewer problems with compatibility.
- ◆ Group like functions within a subsystem. This reduces the exchange of common data and focuses development expertise.
- ◆ Consider the use of standard interfaces, e.g., mechanical, electrical and data (e.g., data exchange standards from the Consultative Committee for Space Data Systems).



Key Interface Documents

- ◆ Interface Definition Document (IDD) - defines interfaces to an existing system such as a launch vehicle. It says what interface someone else must meet to use the launch vehicle. Can be anything, such as mass, type of connector, EMI...
 - Owned by manager of the system with which you want to interface
 - Probably not going to change
- ◆ Interface Requirement Document (IRD) - defines interfaces for two developing systems. Includes both physical and functional interfaces and ensures hardware & software compatibility.
 - Jointly managed (NEEDS ONE OWNER) and signed by the managers of the two systems in development.
- ◆ Interface Control Document (ICD) - Identifies the design solution for the physical interface (drawings).



The Most Common Interface Tool is the N-Squared Diagram

Definition

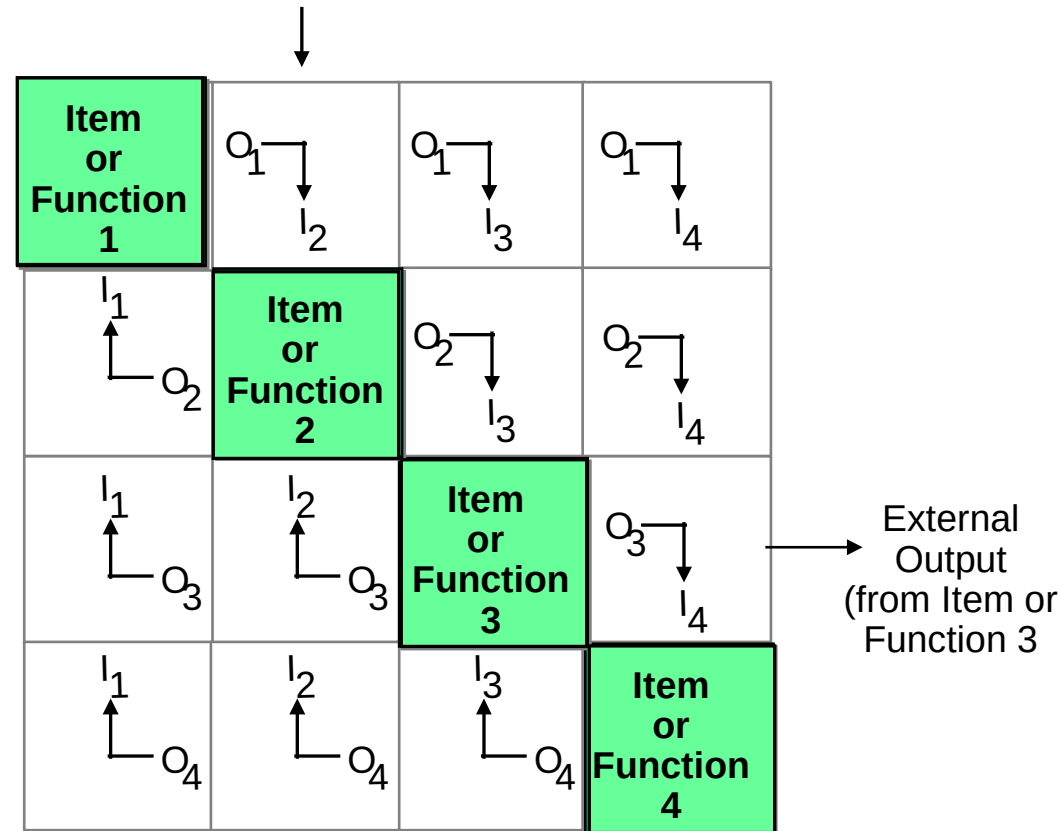
The N-squared (N^2) diagram is used to develop (sub)system interfaces. The system components or functions are placed on the diagonal; the remainder of the squares in the $N \times N$ matrix represent the interface inputs and outputs. Where a blank appears, there is no interface between the respective components or functions. The N^2 diagram can be taken down into successively lower levels to the component functional levels. In addition to defining the interfaces, the N^2 diagram also pinpoints areas where conflicts could arise in interfaces, and highlights input and output dependency assumptions and requirements.

Generic N-squared Diagram as an Interface Artifact

N² diagram rules:

- Items or functions are on the diagonal
- Items or functions have input and outputs
- Item or function outputs are contained in rows; inputs are contained in columns

External Input (to Item or Function 2)

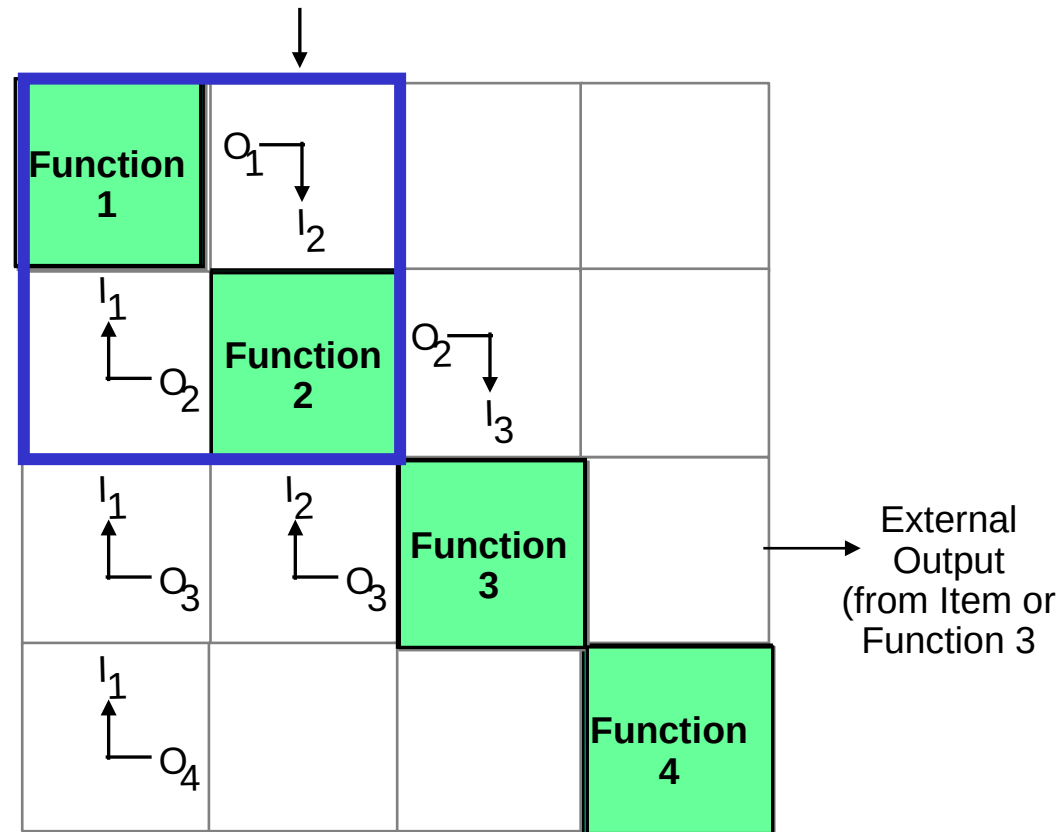


I = Input
O = Output

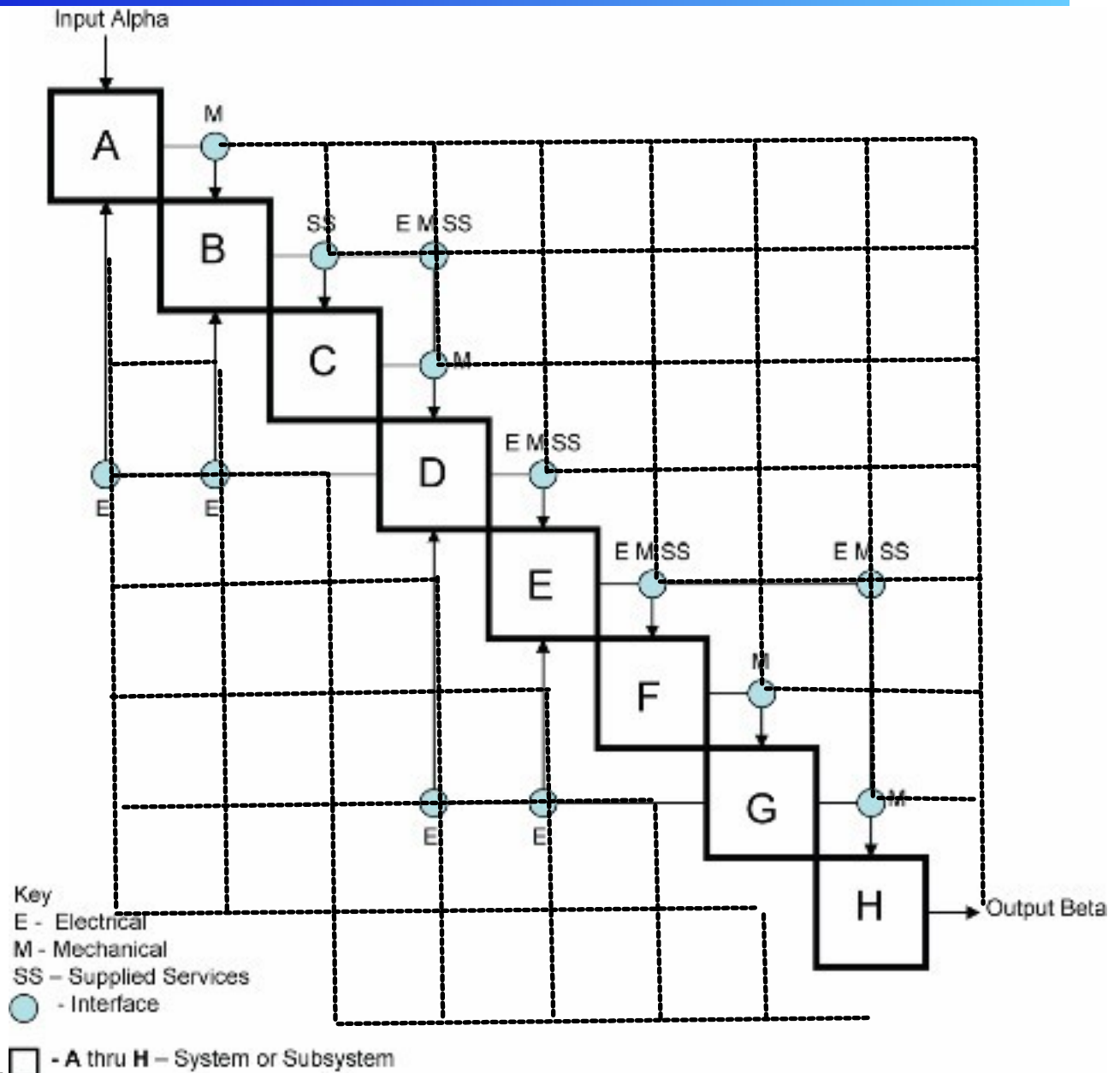
Identify Subsystem Feedback Loops and Candidate Subsystem Boundaries

- ◆ If there is bi-directional information flow between functions - this is a feedback loop.
- ◆ In this example there are feedback loops between functions 1 & 2 and 2 & 3.
- ◆ Consider combining functions where there is a lot of 'coupling' (including feedback loops) within a subsystem. This may simplify the subsystem designs and their interfaces.

External Input (to Item or Function 2)



Example N-Squared Diagram

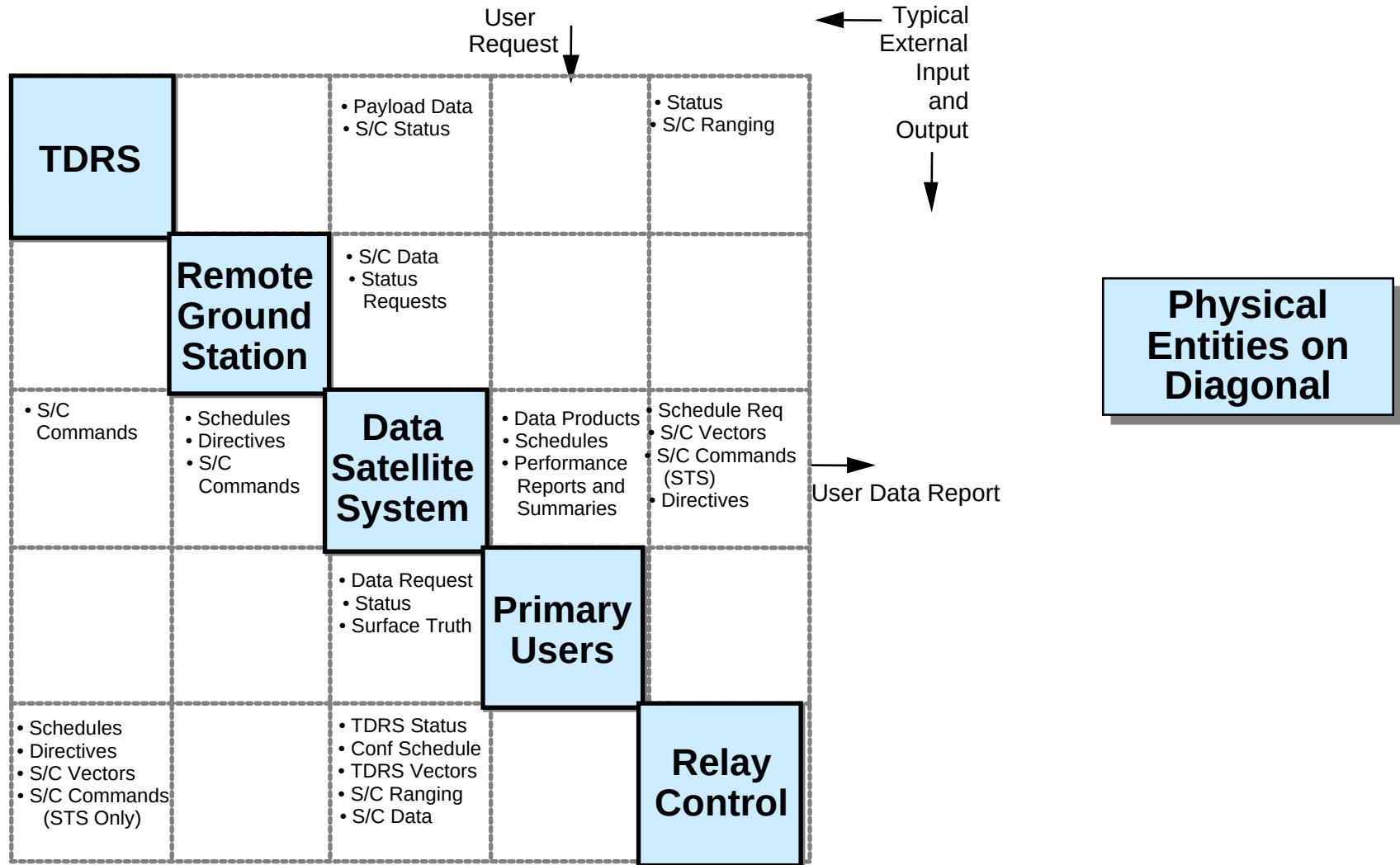


Spacecraft N-squared Diagram Captures the Existence and Type of Subsystem Interfaces

Structure	M	M	M	M	M	M	M	M		M	M	
Fuel pods	M			M	SS					M,E	SS	E
Thrusters										M,E		
Solar arrays					E	SS						
Heat converters					M	SS			M	SS		M,E
Voltage converters								M,E	SS	M,E	E	E
Antenna A								M,E				
Antenna B								M,E			E	
Experiment 1											E	
Experiment 2												E
Experiment 3												M
Gyros												

Key	
E	Electrical/functional
M	Mechanical/physical
SS	Supplied services

TDRS N² Diagram With Information Interfaces

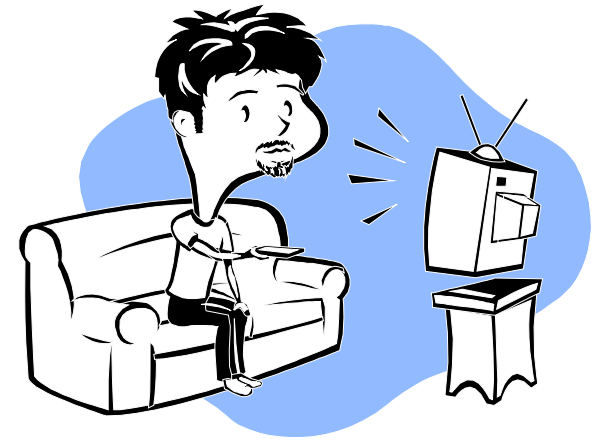


TDRS = Tracking and Data Relay Satellite S/C = Spacecraft

In-Seat Exercise

- ✓ Allocating interface requirements
- ✓ Identify the interfaces in the N² diagram

Human	Push Button ↙	
↖ Keyboard Display	Remote Control Unit	RF Signal ↘
↖ Audio & Video		Television



Pause and Learn Opportunity

Review the sample Interface Requirements Document from the GLAST mission. (IRD_GLAST.pdf)

Understand the need for such interface documents.

Point out:

- Use of definitions
- Spacecraft to Instrument requirements
- GLAST N² charts

Module Summary: N² and Interfaces

- ◆ System external interfaces are defined, distributed and managed like other system requirements.
- ◆ Internal interfaces, since they are created as the system is decomposed, can be optimized by the development team.
- ◆ In developing interfaces group like functions, keep them simple and consider the use of standard interfaces.
- ◆ Since all interfaces run the risk of being ignored as development teams focus on their subsystem responsibilities, explicitly identify the owners of all interfaces.
- ◆ N-squared diagrams the most common interface identification and management tool. They are used to:
 - Capture the existence and nature of an interface
 - Highlight input and output assumptions and requirements
 - Demonstrate where there are feedback loops between subsystems
 - Identify candidate functional allocations to subsystems

Backup Slides for Interfaces Module

Managing Technical Interfaces

- Interface Specifications (IS) and Interface Control Documents (ICD):



- Firm agreement between two parties
- Need an IS or ICD for each external partner and often for internal partners
- Each IS or ICD may specify multiple interface requirements

How formal do these need to be?

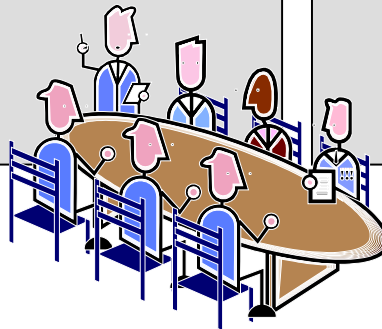
Managing Interfaces – Interface Control Working Group (ICWG)

◆ Purpose and role

- Focus on solution interfaces - both external and internal
- Participating partners on the ICWG
- Under change management authority
- Conflict resolution
- Maintain interface integrity - synchronization of changes with documentation

◆ Managing Interface Agreements

- Documenting the interface is critical
- Agreement between the partners is essential
- May include many interfaces
- Evaluate the impacts of proposed changes
- Closely manage the agreement – it is a contract between the interfacing parties



Scope Elements - Definitions

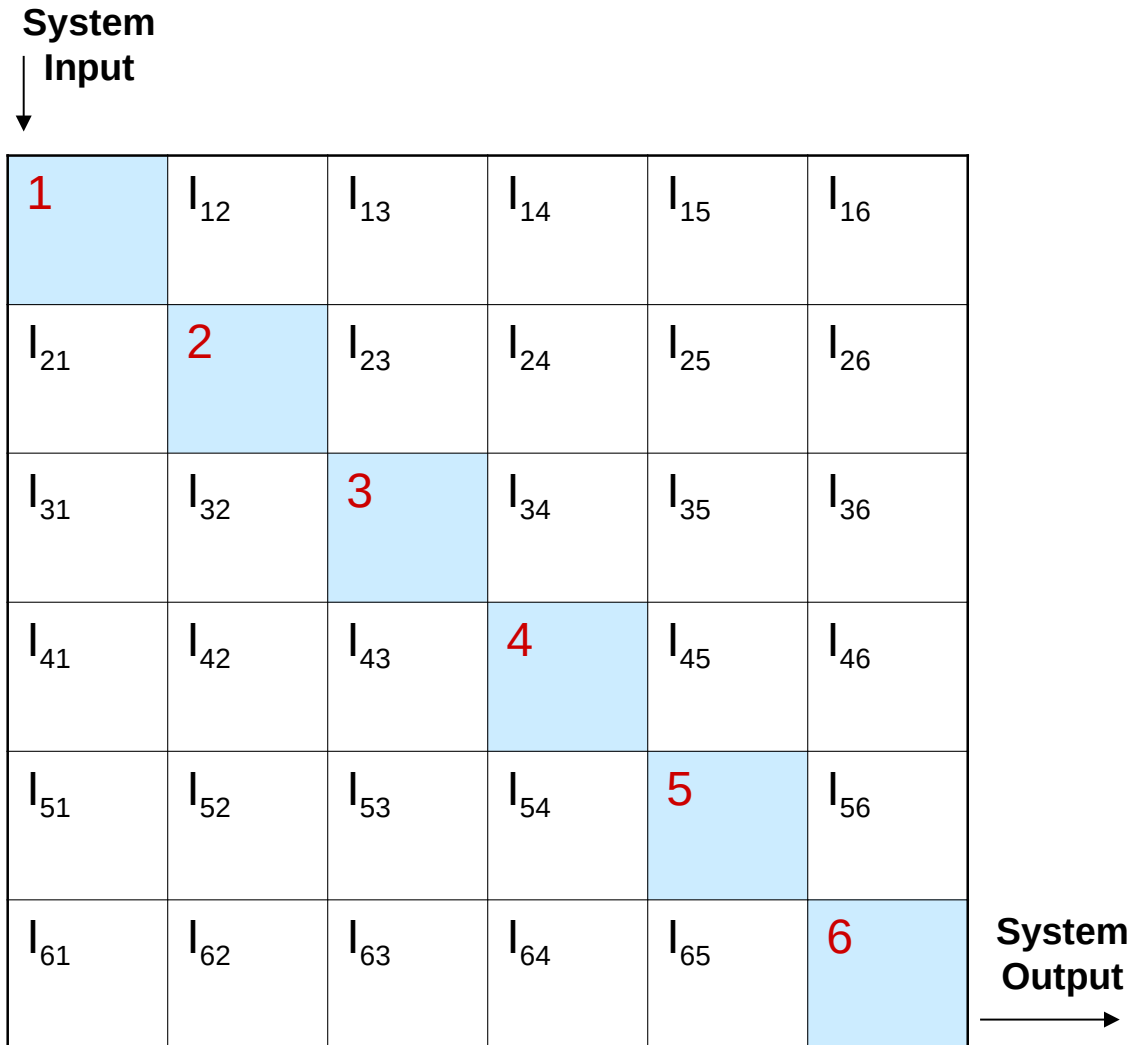
Interfaces

- ◆ **External interfaces form the boundaries between the system-of-interest and the rest of the world.**
- ◆ **Ask the following questions about each boundary to the system-of-interest:**
 - **What does the system do to/for the world?**
 - **What does the world do to/for the system?**
 - **What is the worst thing that can happen across this interface?**
 - **Is the interface likely to change during the development of the system?**
 - **Is this interface likely to change after the system is in use?**
- ◆ **It is useful to create an external interface diagram to the system-of-interest.**
- ◆ **Document every industry standard or Interface Control Document (ICD) that exists for the external interfaces.**
- ◆ **Interface verification check questions:**
 - **Have you identified and documented all product interfaces?**
 - **Have you created a mechanism to monitor interface changes outside your control?**
 - **Have you involved people from the other side of the external interface?**
 - **Have you simplified interfaces as much as possible?**

The N^2 Diagram*

Constructing an N^2 Diagram

1. The system components (1-6 in this case) are placed on the diagonal.
2. The outputs from each component are placed in the horizontal rows.
3. The inputs to each component are placed in the vertical columns.



* Also known as design structure matrix.

Allocating Interface Requirements

- Decomposition into lower-level entities establishes interfaces between those entities.
- External interface requirements must be satisfied by the entity or allocated to lower-level entities.
- Interfaces between lower-level entities must be specified.
- Specify only to the extent required.



Build in and maintain options as long as possible in the design and implementation of complex systems. You will need them.