SYS 5460: Context , Interfaces, Scope

Contents

- Concept of Operations
- System Context
- Interfaces
- Examples

Department of Computer & Engineering Sciences

College of Engineering

Florida Institute of Technology

How the customer explained it

How the project leader understood it

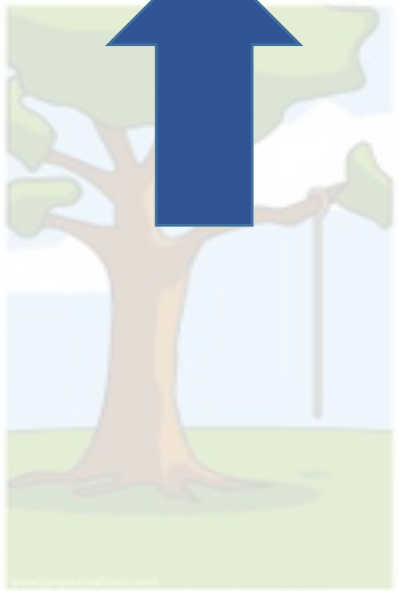
How the analyst designed it

How the programmer wrote it

What the beta testers received

How the business consultant described it

How the project was documented



What operations installed

How the customer was billed

How it was supported

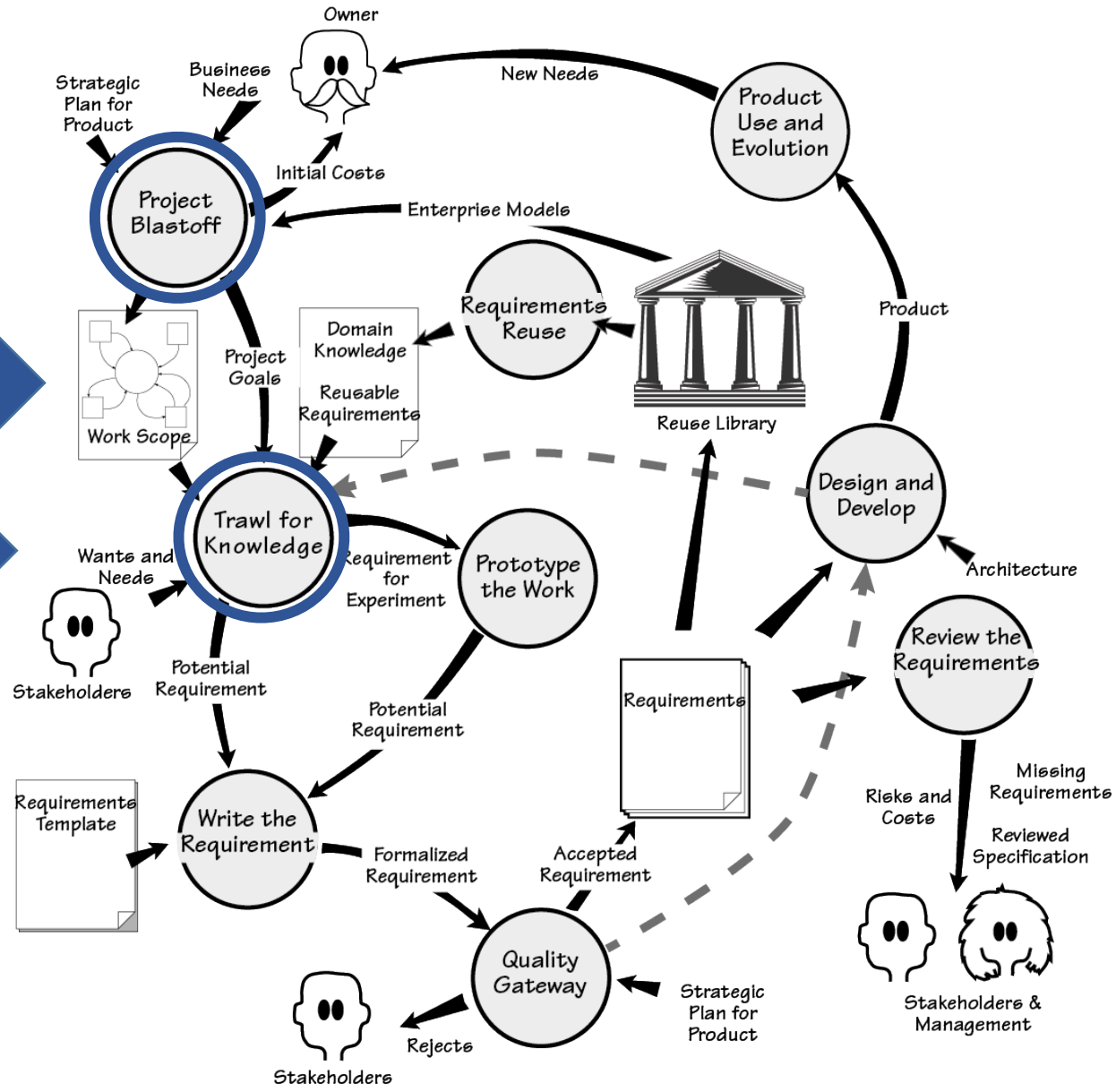
What marketing advertised

What the customer really needed

Volere Requirements Process (Robertson)

Model system functionality

Concept of Operations



Standards For Concept of Operations

Previous standards

ANSI/AIAA-G-043 Outline

1. Scope
2. Referenced Documents
3. User-Oriented Operational Description
4. Operational Needs
5. System Overview
6. Operational Environment
7. Support Environment
8. Operational Scenarios

IEEE 1362 Outline

1. Scope
2. Referenced Documents
3. The Current System or Situation
4. Justification for and Nature of Changes
5. Concepts for the Proposed System
6. Operational Scenarios
7. Summary of Impacts
8. Analysis of the Proposed System

Current standards ISO/IEC 12207 Software lifecycle process or ISO/IEC 15288 Systems and Software lifecycle processes

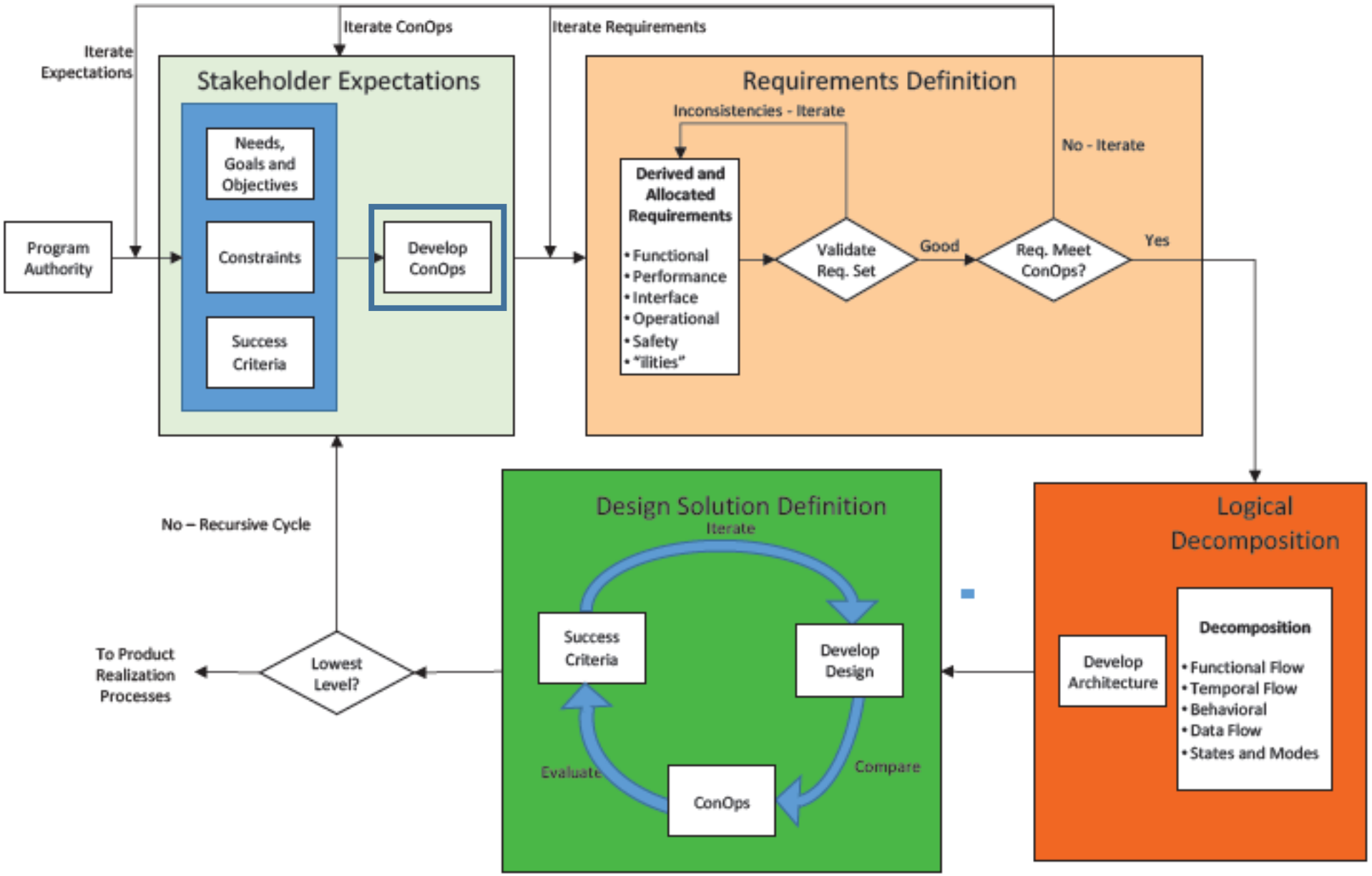
concept of operations

verbal and/or graphic statement, in broad outline, of an organization's assumptions or intent in regard to an operation or series of operations

Note 1 to entry: The concept of operations frequently is embodied in long-range strategic plans and annual operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be carried out simultaneously or in succession. The concept is designed to give an overall picture of the organization operations. See also operational concept.

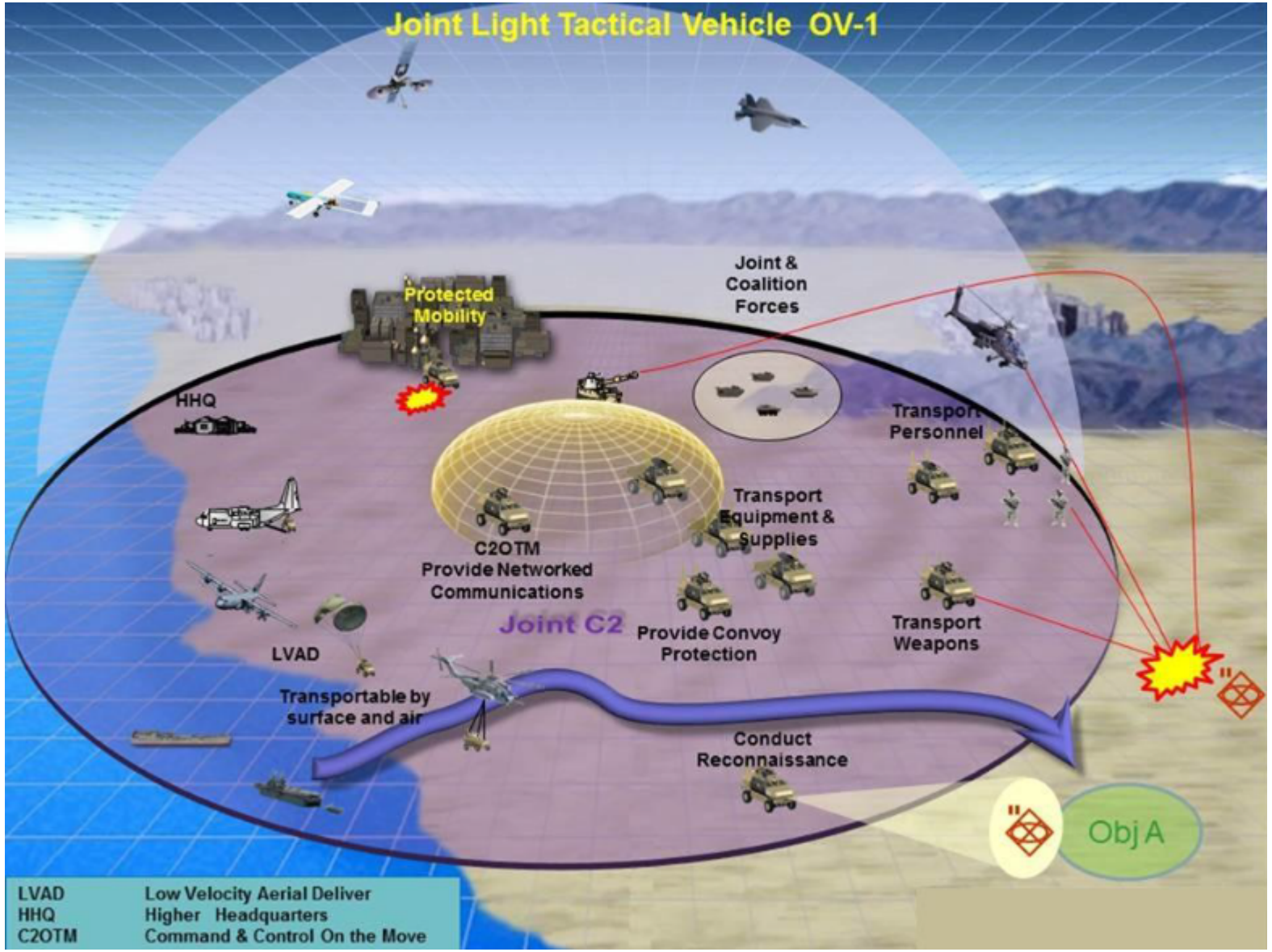
Note 2 to entry: It provides the basis for bounding the operating space, system capabilities, interfaces and operating environment.

NASA SE Handbook



DoDAF Viewpoints and Models

- *OV-1: High-Level Operational Concept Graphic*
- *OV-2: Operational Resource Flow Description*
- *OV-3: Operational Resource Flow Matrix*
- *OV-4: Organizational Relationships Chart*
- *OV-5a: Operational Activity Decomposition Tree*
- *OV-5b: Operational Activity Model*
- *OV-6a, 6b, 6c: Introduction*
- *OV-6a: Operational Rules Model*
- *OV-6b: State Transition Description*
- *OV-6c: Event-Trace Description*



Concept of Operations Document (IEEE 1362-1998)

Title page
Revision chart
Preface
Table of contents
List of figures
List of tables
1. Scope
    1.1 Identification
    1.2 Document overview
    1.3 System overview
2. Referenced documents
3. Current system or situation
    3.1 Background, objectives, and scope
    3.2 Operational policies and constraints
    3.3 Description of the current system or situation
    3.4 Modes of operation for the current system or situation
    3.5 User classes and other involved personnel
    3.6 Support environment
4. Justification for and nature of changes
    4.1 Justication of changes
    4.2 Description of desired changes
    4.3 Priorities among changes
    4.4 Changes considered but not included
5. Concepts for the proposed system
    5.1 Background, objectives, and scope
    5.2 Operational policies and constraints
    5.3 Description of the proposed system
    5.4 Modes of operation
    5.5 User classes and other involved personnel
    5.6 Support environment
6. Operational scenarios
7. Summary of impacts
    7.1 Operational impacts
    7.2 Organizational impacts
    7.3 Impacts during development
8. Analysis of the proposed system
    8.1 Summary of improvements
    8.2 Disadvantages and limitations
    8.3 Alternatives and trade-offs considered
9. Notes
Appendices
Glossary

Volere Requirements Template

ConOps

Project Drivers

1. The Purpose of the Project
2. The Stakeholders

Project Constraints

3. Mandated Constraints
4. Naming Conventions and Terminology
5. Relevant Facts and Assumptions

Functional Requirements

6. The Scope of the Work
7. The Business Data Model and Data Dictionary
8. The Scope of the Product
9. Functional Requirements

Non-functional Requirements

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural Requirements
17. Legal Requirements

Project Issues

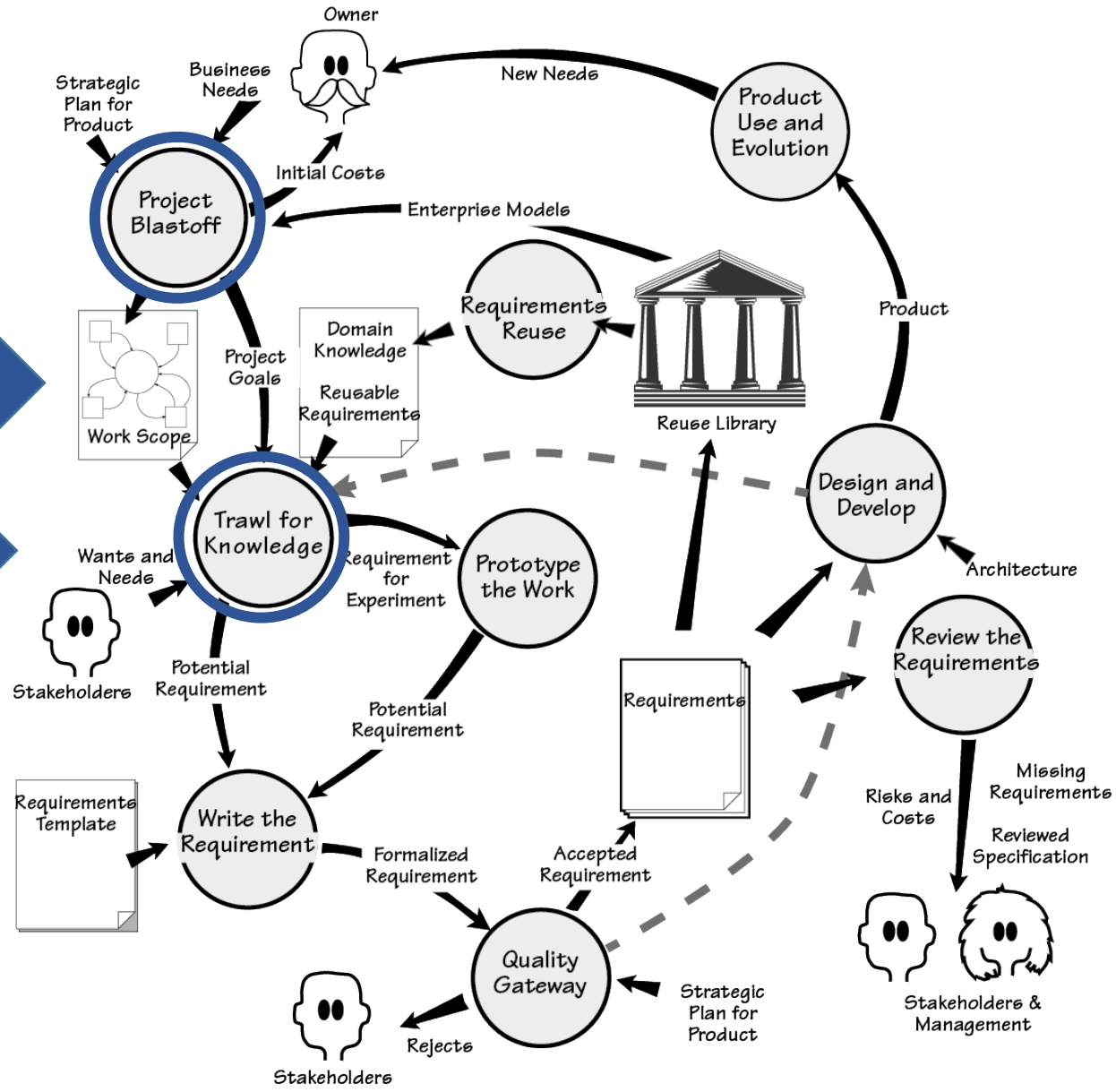
18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Chapters 1-3 Robertson

Volere Requirements Process (Robertson)

Model system functionality

Concept of Operations

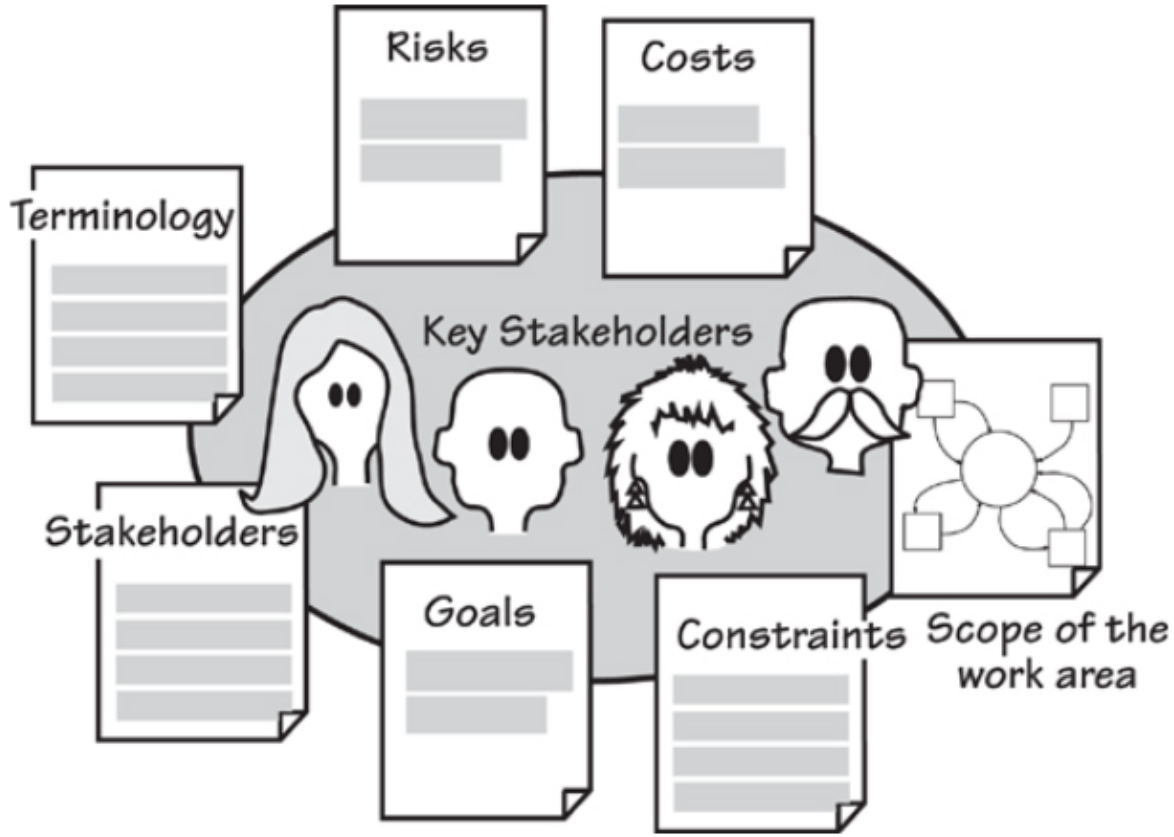


Project Blastoff (kickoff)

- The key purpose of project blastoff is to build the foundation for the requirements discovery, and to ensure that all the needed components for a successful project are in place.

- Involvement:
  - Principal Stakeholder (sponsor)
  - Key users
  - Lead requirement analyst
  - Technical business experts

- Discussions:
  - Purpose of the project
  - Scope of the work
  - Stakeholders
  - Constraints
  - Included/excluded functionalities
  - Special terminology
  - Facts & Assumptions
  - Cost estimates
  - Risks



LECTURE 4 Continued

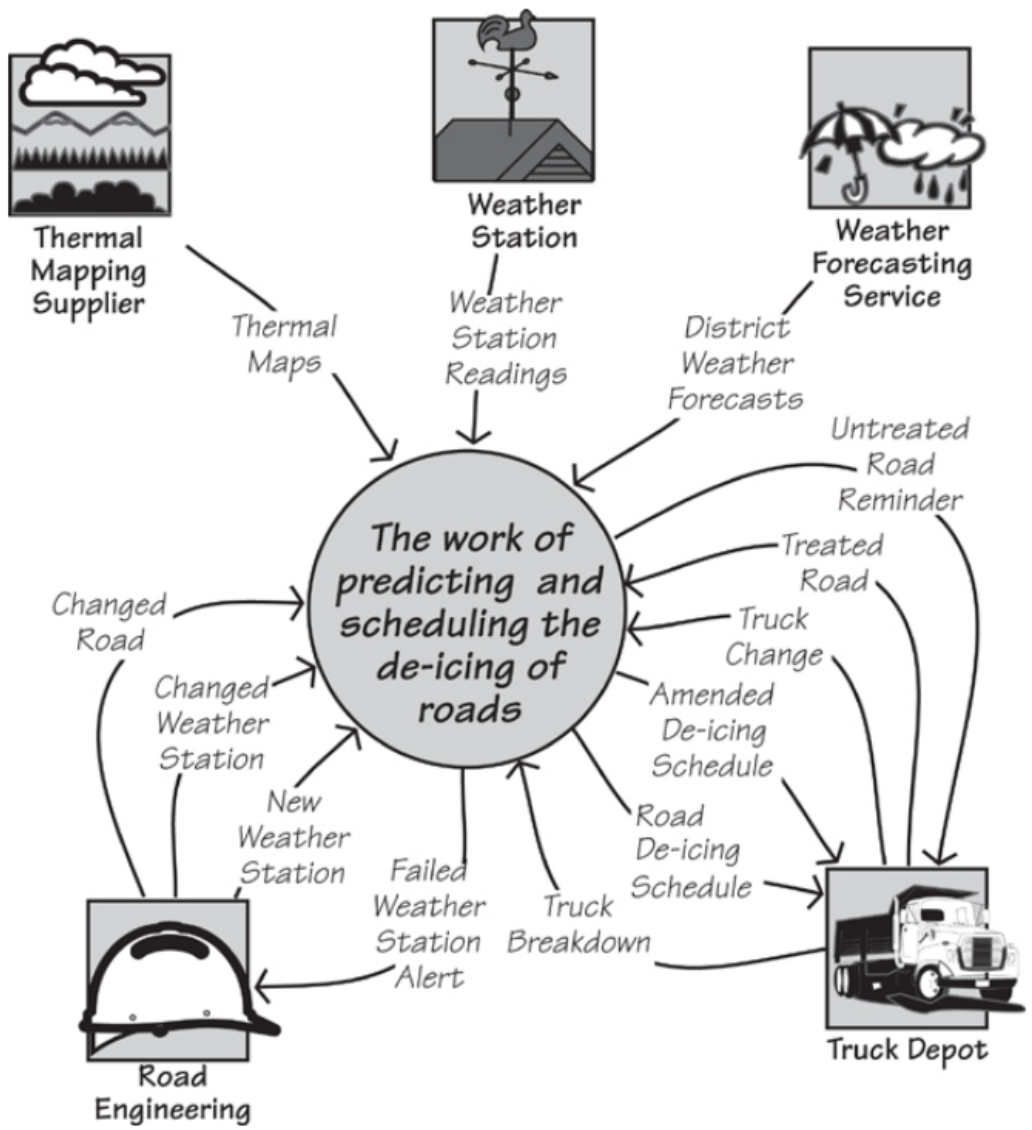
Importance of Product Scope

- A system react to external events
- Boundaries are not always well defined (soft systems, vs hard systems)
- Without boundaries scope creep is likely to occur
- Take time up front early in the project to define the scope of the product
- Understand the product context and interfaces with enough detail to be able to make accurate scoping decisions
- For some projects or program this may influence parts of the work and requirements that are “flown down” to external entities

Scoping the system allows separating the **Work** from its **Environment**

To achieve the optimal value for the owner, study enough of the owner’s work to identify what is valuable (Robertson)

Context Diagram (Robertson)



The work context shows where the responsibilities of the work and the responsibilities of the adjacent systems start and end.

the responsibilities are defined by the flows of data on the context diagram. Always ask …Why is this flow there?

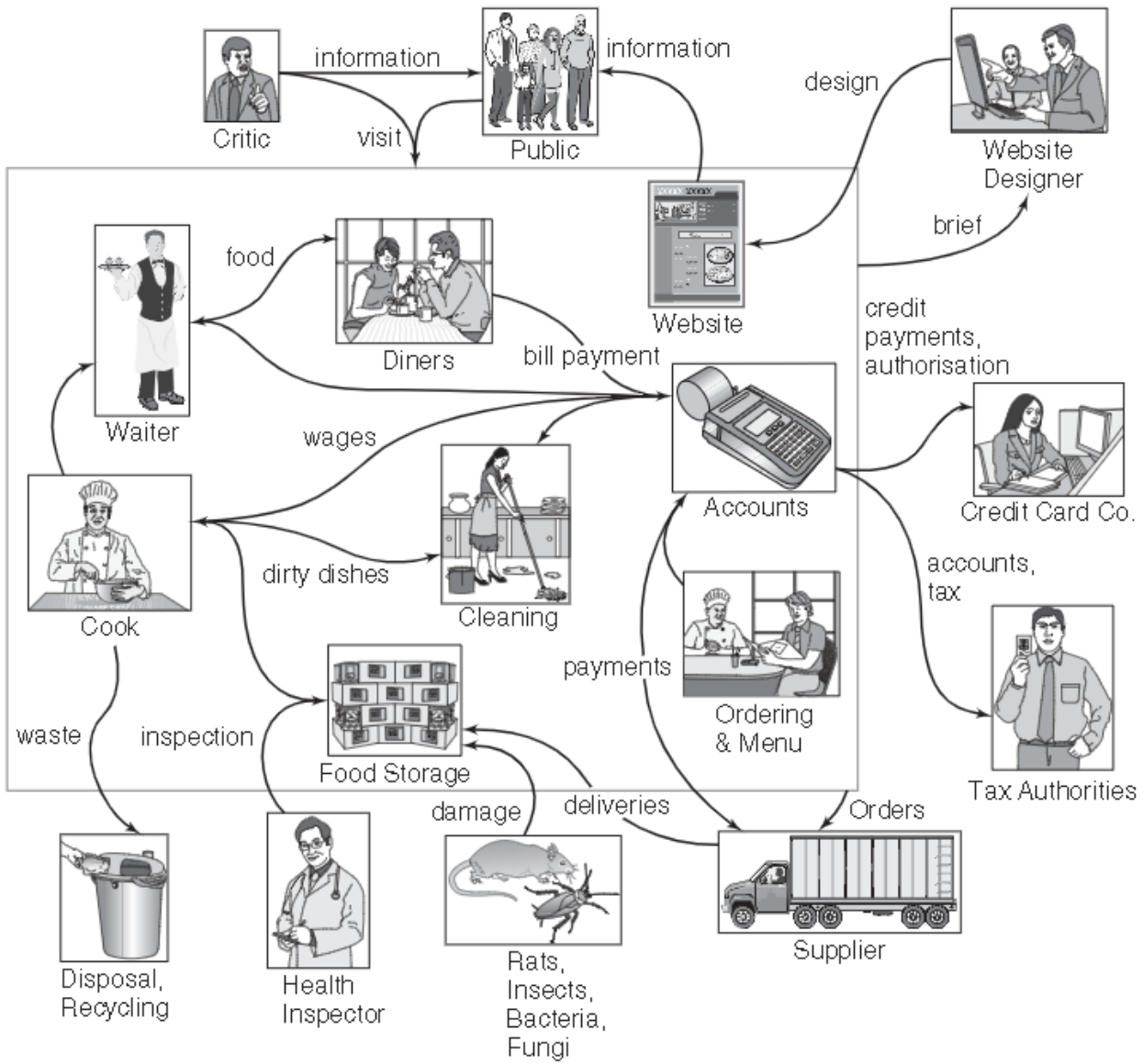
The context diagram intentionally ignores anything that happens outside the boundary that does not directly involve an interface with they system in question

Move from an soft system problem to a hard system problem

Soft System

- A soft system involves social, political and emotional issues as well as technology.
- Soft systems address messy , ill-structured problem situations
- A hard system may be in place , but it may fail for soft reasons:
  - IT solution to provide support, may fail because too complicated for some users
- Once implemented a product or service become part of the system
- Narrow system boundaries-> hard system
- Wide boundaries->soft system
- Draw a rich picture of the system showing:
  - Stakeholder not directly involved in the operations
  - Issues and concerns
  - Processes of stakeholders
  - Negative aspects
- The system boundary is outside the product boundary

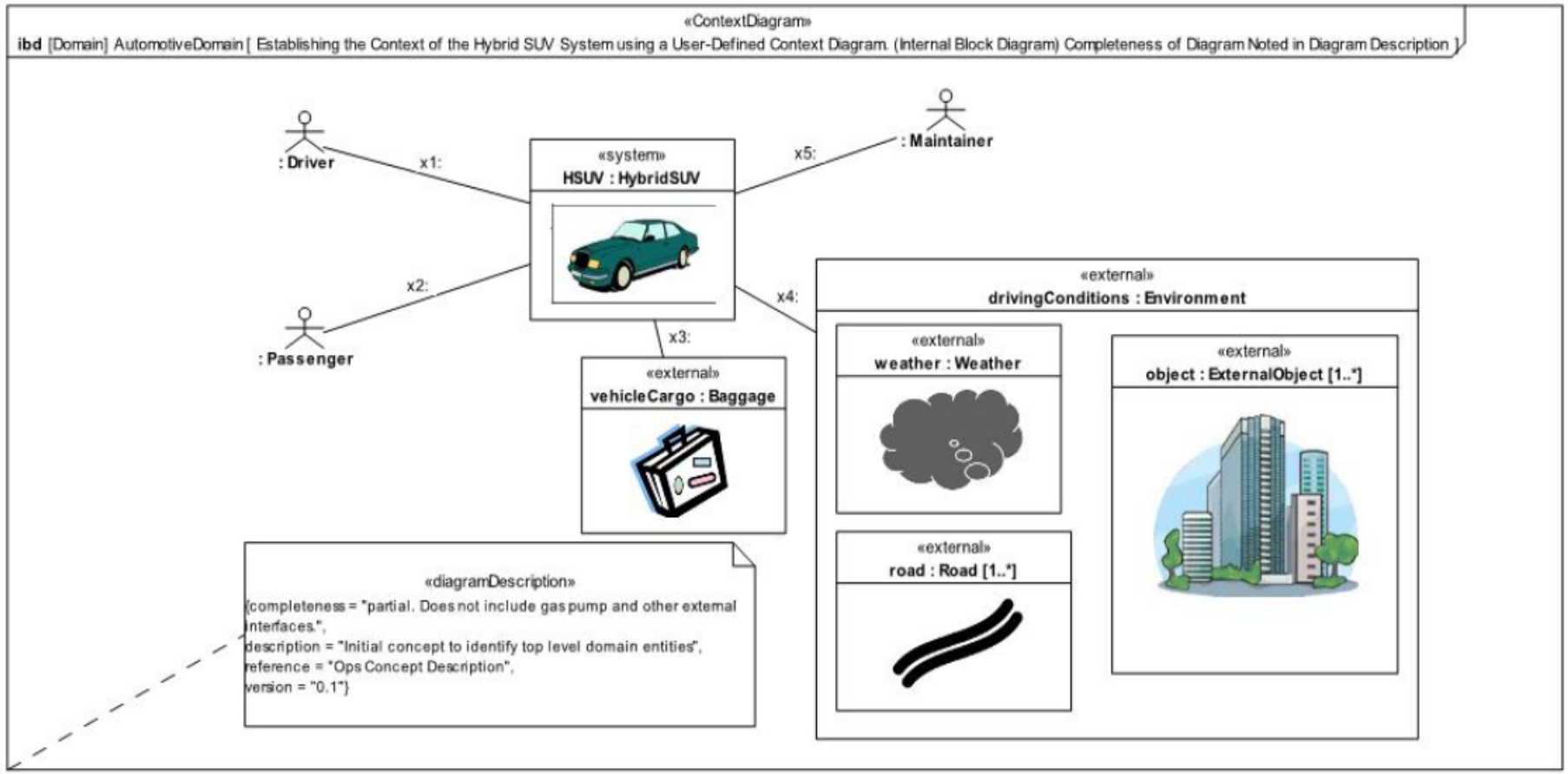
Example: Rich Picture of a Restaurant



What will be the system , if your product is the restaurant accounts?

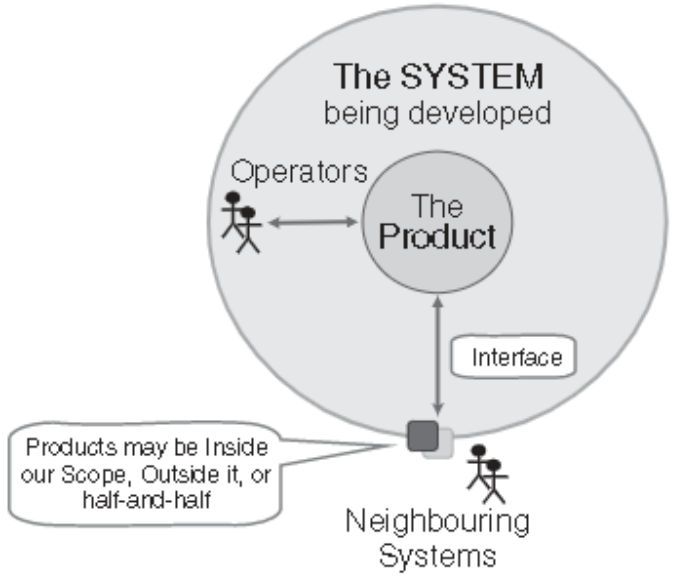
Context Diagram of an SUV (SysML 1.6 Reference Document)

- An actor is an entity that interact with system
- Actors are outside the boundaries of the system
- Actors could be:
  - Hardware devices
  - Legacy systems
  - Other subsystems
  - Human users



System and its Context

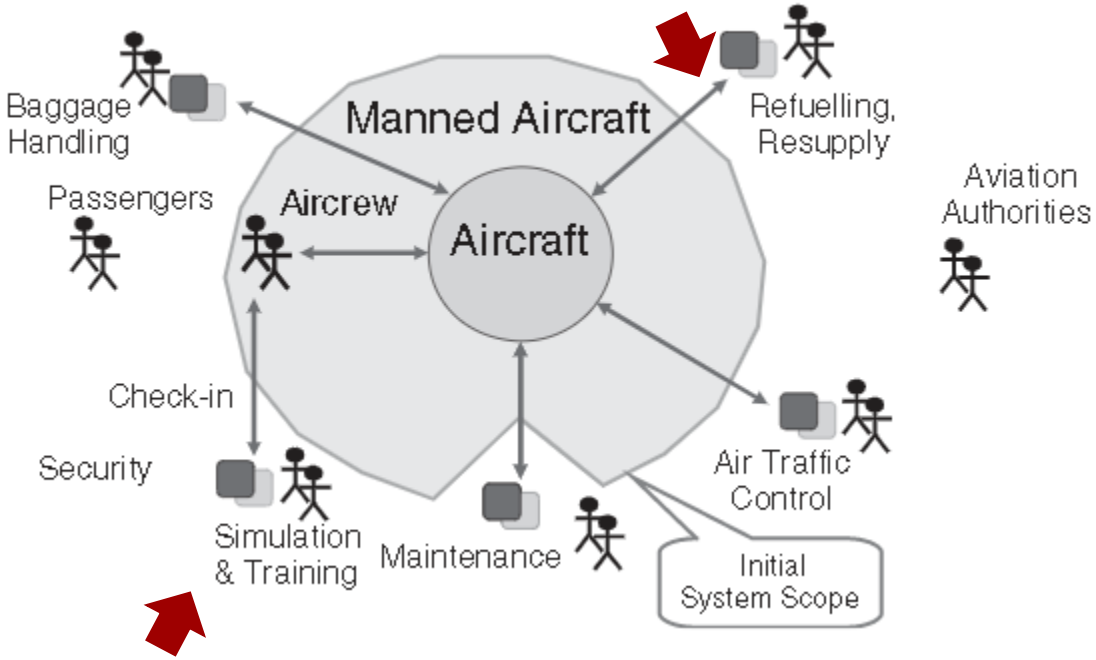
- System consists on several components working together
  - Products
  - People
  - Procedures

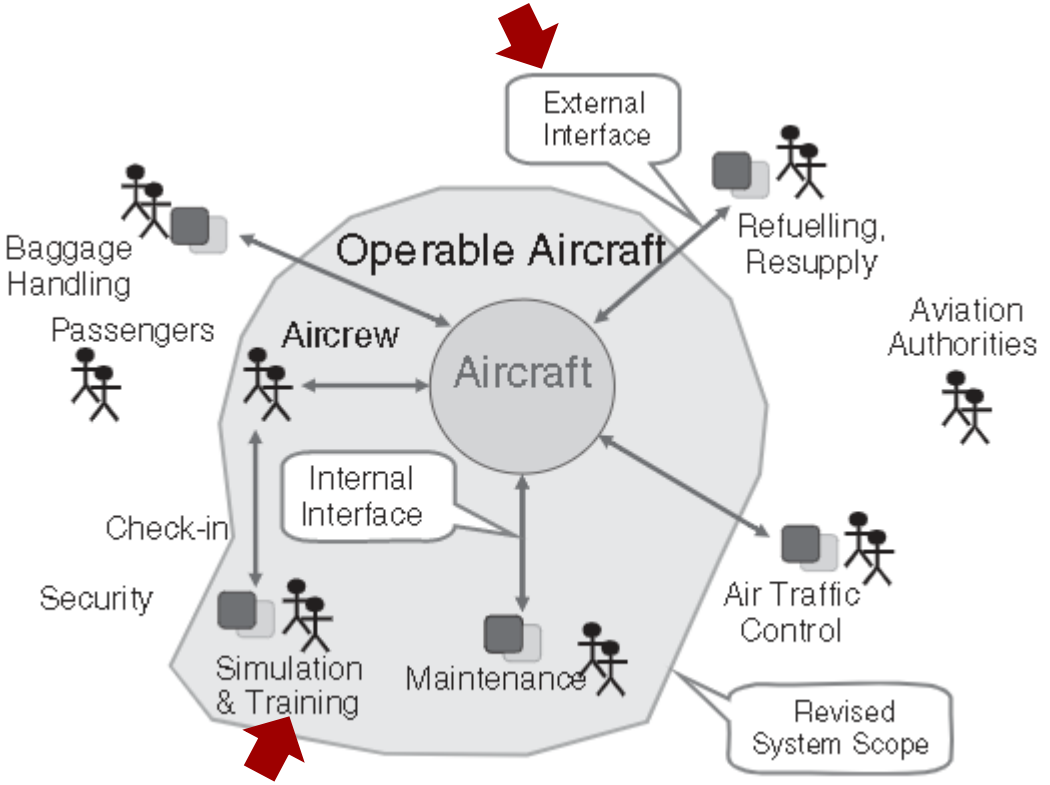


A system is a construct or collection of different elements that together produce results not obtainable by the elements alone.

Example of Scope Definition

- Spot the differences in the two system representations
- Discuss on pros and cons of the two scopes





Identifying Interfaces

- Interface: a point where two systems, subjects, organizations, etc., meet and interact
- Interfaces are not necessarily hardware:
  - USB, Firewire
  - Gas tank and pump at gas stations
  - Mailbox
  - Point of contact
- Pay attention to interfaces among people (soft system)
- Create a context diagram

Interfaces and Requirements Elicitation Techniques

Processes

GUI

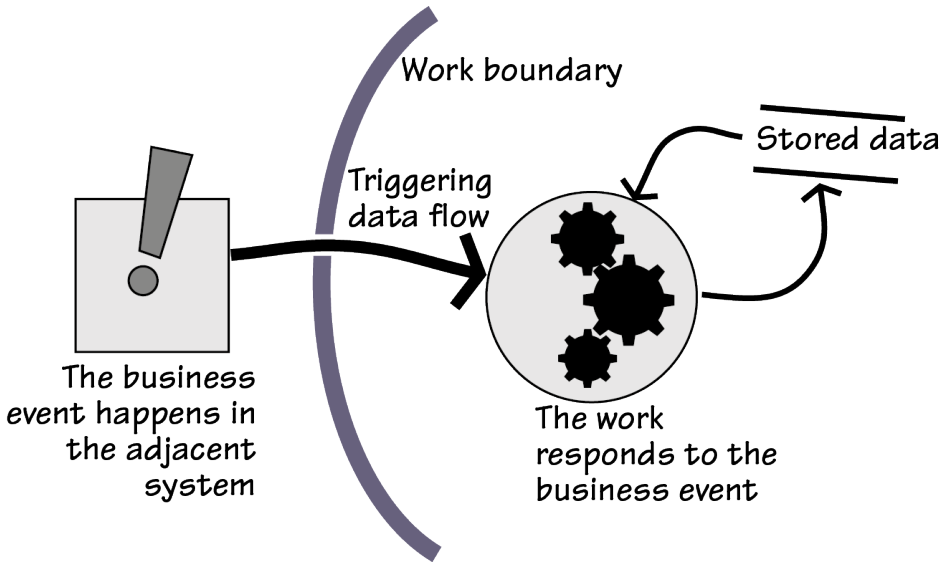
API

Type of interface	Example	How to discover this	Treatment
Between people	Aircrew report a problem to maintenance staff	Study the 'soft' system and its business processes (this chapter) Scenarios (Chapter 5)	Write a procedure; train crew to follow the procedure (i.e., write a training course, and run it)
Between people and products (These can be divided into inputs and outputs)	Aircrew interact with controls on the aircraft, both giving commands and receiving information	Scenarios (Chapter 5) Prototyping (Chapter 13) Similar/existing/rival products (Chapter 13)	Specify the user interface, considering effect on work (ergonomics); write a procedure for each use of the interface; train crew to follow the procedure (i.e., write a training course, and run it)
Between products (These can be in either direction, or both)	Automatic equipment diagnoses a fault on the aircraft	For interfaces to existing external systems: standards, or manufacturers' data sheets For interfaces to existing products/systems: speak to your developers/suppliers For new interfaces: iterate requirements and design (Chapter 14)	Specify the interface (the data and other quantities to be exchanged, and the required behaviour), using standards if possible; impose the interface as requirements on both products

Scope and System Events

- Flows are provided as event that are enabled by the interfaces
- Draw or establish events on the boundary of the system
- An arrow into the circle on a context diagram means an event
  - Unpredictable arrival of a signal to which the **system has to respond by carrying out one or more tasks-> functional decomposition**
- It is a good practice to **start with a list of in/out events the system need to handle**
  - **(or the system needs to trigger, then handle…)**
- The scope consists on deciding the subset of events that will be handled by the system
- Handling such event become a requirement if agreed in the scope
- This has contractual implications for the product acceptance
- Events could be:
  - External: message, signal , control input
  - Time-triggered: time signal, polling cycle, etc.

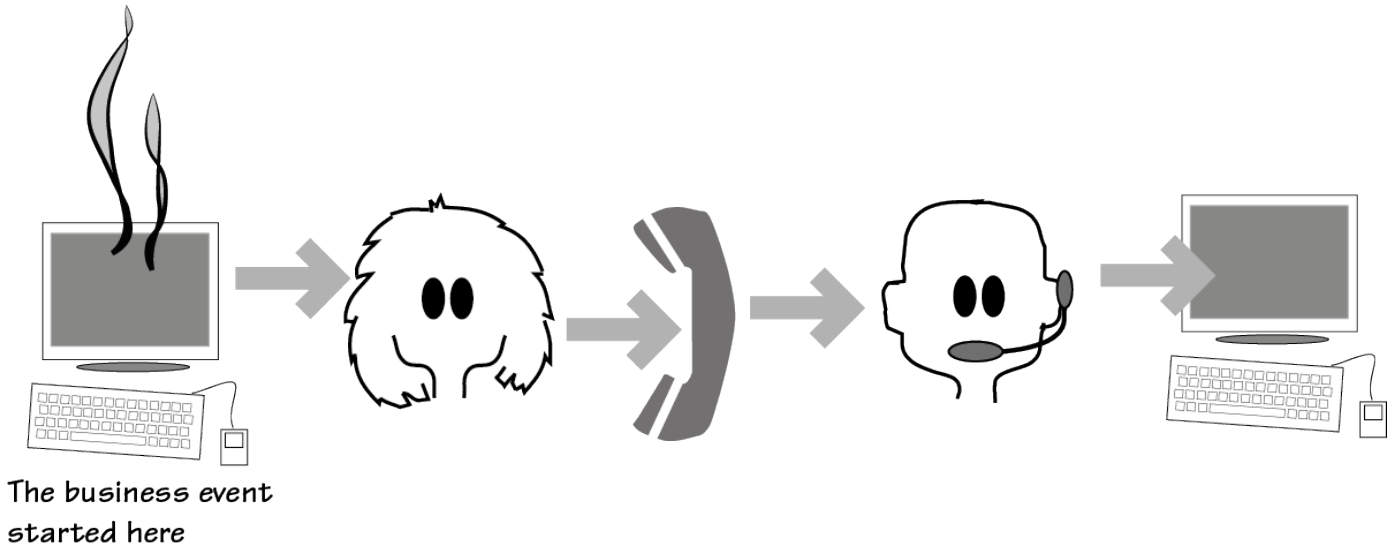
Flows are Triggered by Business Events (Ch4 Robertson)



Understanding of the sources of the business event allow to a better understanding of the system and to come up with innovative designs

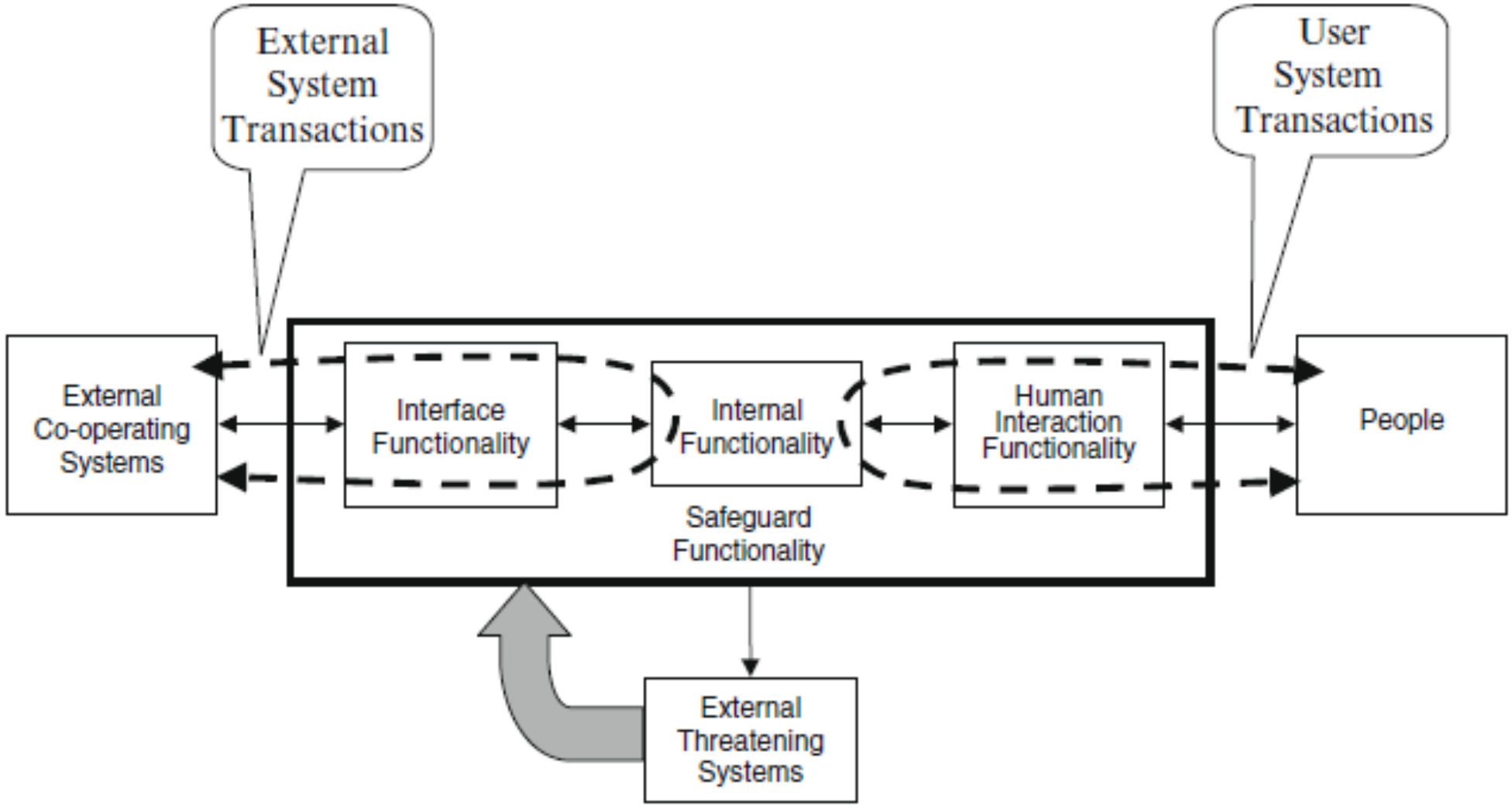
Business events can be time-based triggered (timers, or a monthly bill)

Establish the true actors of the system and flows by identifying business events



The maintenance call is received by the operator and input in the system *

Grouping Events Based on Interfaces (Hull, 2011)



Context Diagram and Events

Event/Task	In/Out
Load Bags	In Scope
Unload Bags	In Scope
Refuel	In Scope
Check-in Passengers	Out of Scope
Radio Air Traffic Control	In Scope
Provide Airport Security	Out of Scope

Event Handling Functions

- Express event handling by:
  - Textual event handling requirements
  - Condition-action tables
- Template for a traditional event handling requirement:
  - When<event is detected> do <required action>
  - Every <time period> do <required action>

When pressure in Main_Input_Pipe falls below Minimum_Input_Pressure, run Main_Pump at Full_Power.

Every Sample_Period, send Cylinder_Temperature to Engine_Controller.

When Payment_Confirmation is received, set Order_Status to To_Be_Despatched.

If Account_Balance exceeds Private_Banking_Threshold for Qualifying_Period then issue Private_Banking_Invitation to Customer.

Event Handling Functions(2)

- Condition-Action Tables
  - Condition action tables allow a more formal definition
  - Conditions are expressed in columns (AND)
  - Implication: conditions are mutually exclusive

When condition$_1$	and condition$_2$	then
Pressure (Main_Input_Pipe) < Minimum_Input_Pressure		*Run Main_Pump at Full_Power*
Pressure (Main_Input_Pipe) > = Minimum_Input_Pressure		*Main_Pump Inactive*

Users and Scenarios

- Using the events and interfaces identified, lets generate user cases and system scenarios

As a <user role> when <event> , I want to be able to <capability> so that I can <goal>

Requirement Elements / Discovery Contexts	Stakeholders	Goals	Context, Interfaces, Scope	Scenarios	Qualities and Constraints	Rationale and Assumptions	Definitions	Measurements	Priorities
Introduction									
From Individuals									
From Groups									
From Things									
Trade-Offs									
Putting it all Together									

- Interfaces=>enable events
- Scope=collection of events
- System capabilities allow stakeholder to achieve goals

How do we achieve capabilities?

Grouping Actions into Scenarios

- The system has to take several actions to achieve a capability
- This lead to the idea of scenario
- A scenario is a sequence of actions

System Output

- The system under design generates events that affect the world outside its boundaries
- Examples include:
  - Sending a message , sounding an alarm, etc.
- For a function or capability to be provided to a human operator:
  - The <operator> shall be enabled to< do the required action>
- For a product function:
  - The <product> shall <do the required actions> <with abc performance>

From Context to Use Case Diagrams

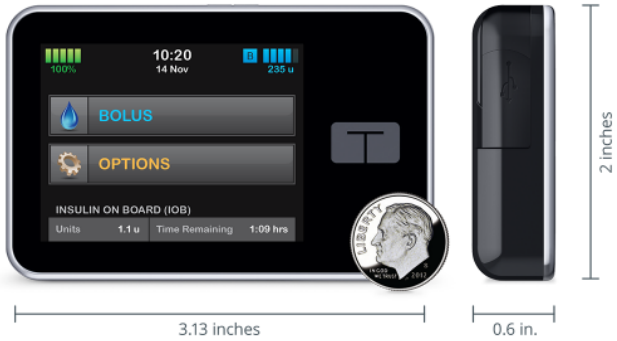
- Context diagram are not exactly use cases
- Context diagrams show interfaces not functions
- Use cases lists all the high-level functions that will allow the actors to achieve their goals (thorugh the use of the system of course)

Sometimes the system is in the interface

dexcom

CONTINUOUS GLUCOSE MONITORING

TANDEM DIABETES CARE



ZERO FINGERSTICKS

When integrated with Dexcom G6 CGM‡

LARGE COLOR TOUCHSCREEN
Easy to read, simple to use interface

WATERTIGHT CONSTRUCTION
Tested to 3 feet for 30 minutes (IPX7)

RECHARGEABLE BATTERY
Convenient charging via micro-USB port

CUSTOM SETTINGS
Create up to six Personal Profiles

BLUETOOTH® WIRELESS TECHNOLOGY
Modern connectivity for a modern world

Product Life Cycle

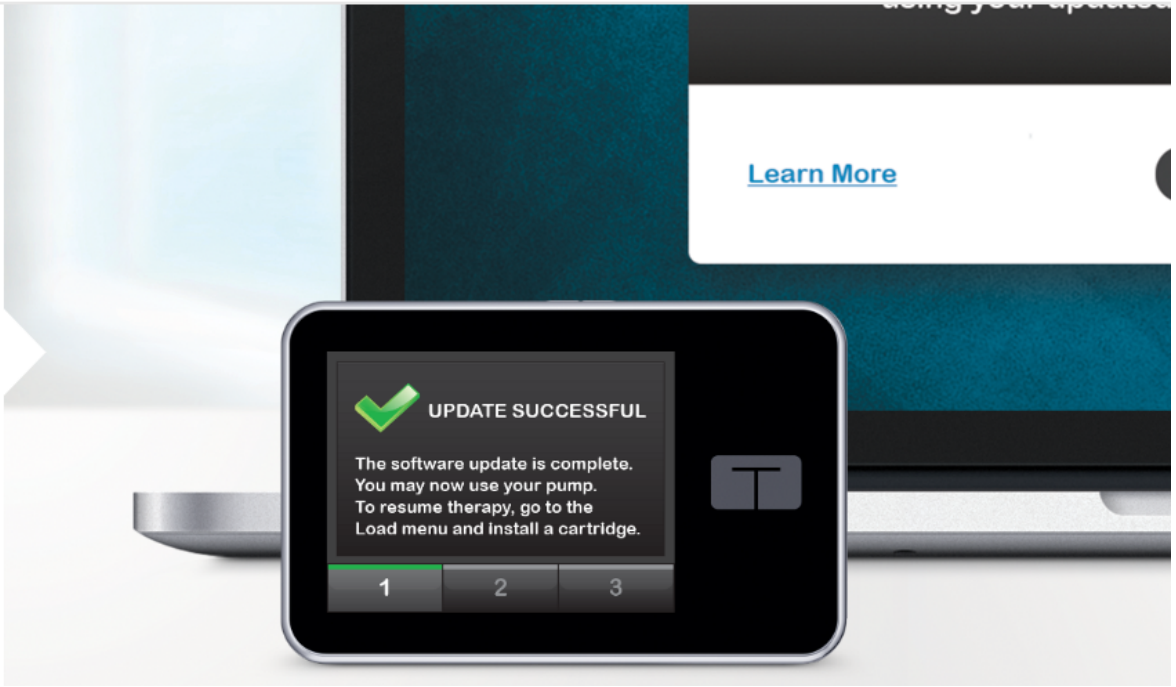
- Requirements exist for the lifecycle of the product

TANDEM® DIABETES CARE

PRODUCTS SUPPORT ABOUT CONTACT GET STARTED

Remote software updates

The t:slim X2 insulin pump is the first FDA-approved insulin pump capable of remote feature updates.† Using a personal computer, patients can keep their pump up to date with the latest technology during its warranty period.



You know interfaces are important when…

Amazon, Apple, and Google unveil smart home collaboration

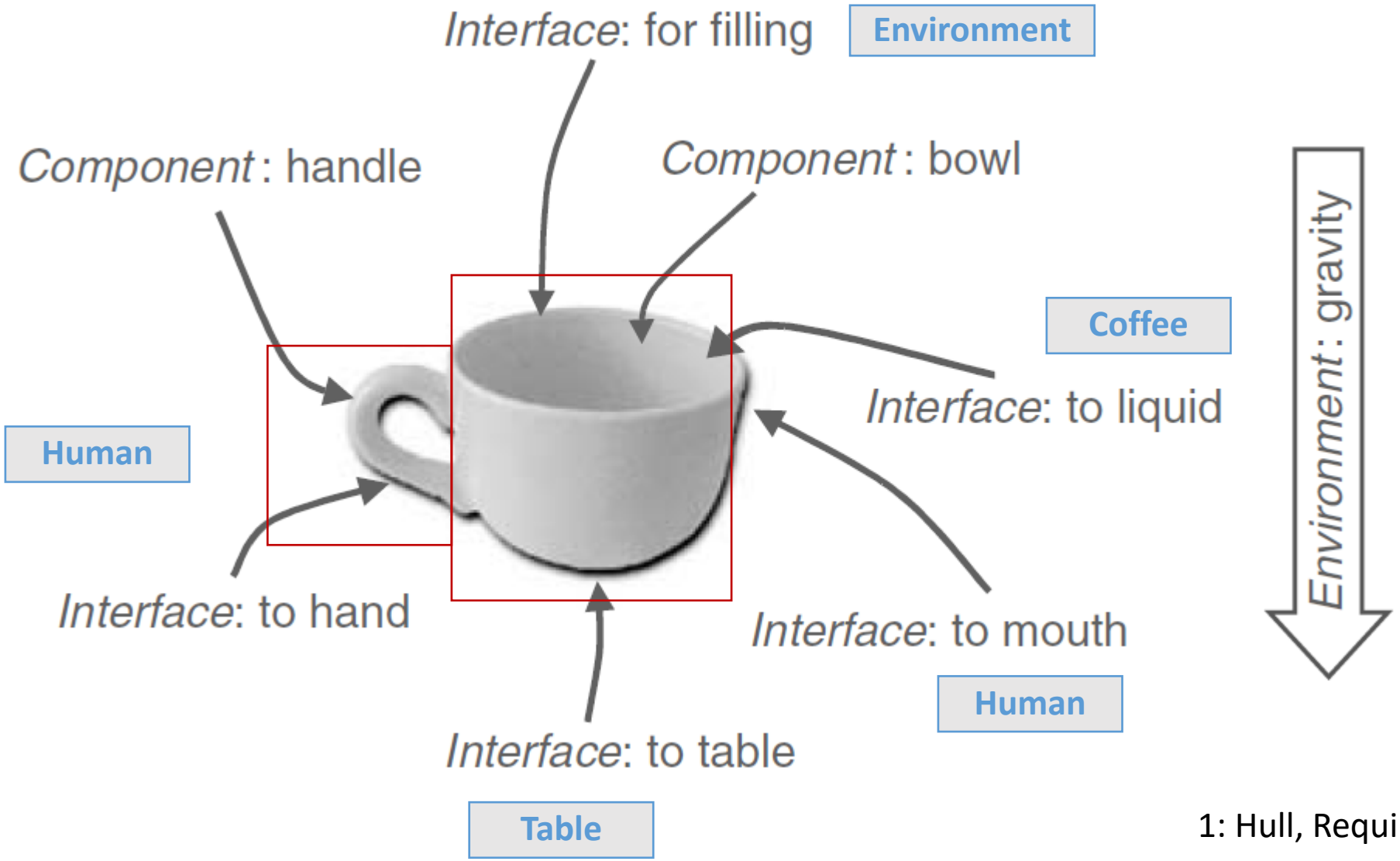
By Pablo Valerio

Posted on December 31, 2019

Interfaces and N-Squared Diagram

- The N-squared (N^2) diagram is used to determine system interfaces between components or functions.
- Rules:
  - The system components or functions are placed on the diagonal
  - Off-diagonal elements represent the interface inputs and outputs
  - Outputs are represented by rows
  - Inputs are represented by columns
  - Blanks mean no interface between the respective i-j components or functions
- Other usage
  - pinpoints areas where conflicts could arise in interfaces, and
  - highlights input and output dependency assumptions
- Not a formal SysML diagram

A Cup Viewed as a System[1]



1: Hull, Requirements Engineering

Interface Documents

- **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 or system elements
- **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.
- **Interface Definition Document (IDD) :** A unilateral document controlled by the end item provider. Provides the details of the interface for a design solution that is already established.

Validate Interfaces and Events

- For standard interfaces, ensure that the exact version of the relevant standard is identified
- For custom interfaces, agree all details of each interface with the owner of the other system
- Each event should be handled by when-requirements or their equivalent condition-action tables
- List of alternative conditions in condition-action tables cover all the possibilities exhaustively
- Check that each scenario referenced in the event handling requirement is defined as a use case
- Check that each interface is defined in the data dictionary

Group Activity
Get in new groups of 4

WHEREWELEARN

Demo Button

Demo Button

Demo Button

Demo Button

Demo Button

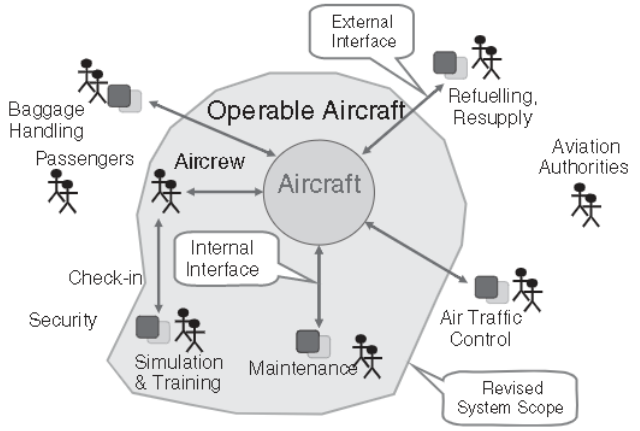
Demo Button

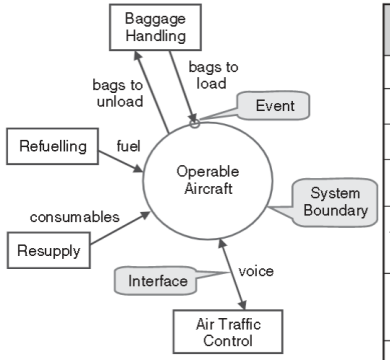
Demo Button

MASTER SYSTEMS DISPLAY

In Class Exercise

- Create a Context Diagram and Events for the selected system.
- Include as many interfaces and events as you can identify
- Present to the class in under 3 min
- May use laptop or paper to present.





Event/Task	In/Out
Load Bags	In Scope
Unload Bags	In Scope
Refuel	In Scope
Check-in Passengers	Out of Scope
Radio Air Traffic Control	In Scope
Provide Airport Security	Out of Scope