Intro to Systems Engineering

Illah Nourbakhsh & Reid Simmons

16-467

Credit: Wettergreen & Nourbakhsh slides

Motivation: The Problem

I don't understand why it doesn't work.

Motivation: The Problem

I don't understand why it doesn't work.

This is the first time I'm trying it all.

Motivation: The Problem

I don't understand why it doesn't work.

It worked fine last night.

illah nourbakhsh | CMU Robotics Institute | SysEng

Motivation: The Problem

I don't understand why it doesn't work.

My part is fine. It's a problem with another part of the system.

Systems Engineering

1 Examine systems from many perspectives

2 Exploit systems engineering skills

plan, management, analysis, design, implementation, evaluation, validation

Perspectives on Systems

Which ingredients does each perspective focus on?

How does each perspective define failure and risk?

Perspectives on Systems

ME: What physical embodiment comprises the system? What material qualities result?

Perspectives on Systems

ECE: How do embedded electronics and firmware enable the system to function?

Perspectives on Systems

CS: What algorithms are embodied by the programming to effect the system behavior?

Perspectives on Systems

Robotics: How do mechanism, electronics and programming interface to create a whole embedded, physical artifact?

Perspectives on Systems

Design: What is the product embodied by the system? What use cases define its place as an artifact that people interact with?

Perspectives on Systems

Human Factors: What is the end-goal of using the system, and how well can people use it to that purpose under varying conditions?

Perspectives on Systems

HCI: How usable is the system by humans who must interface with it?

Perspectives on Systems

Education: What mental models are assumed and developed as humans interact with the system?

Perspectives on Systems

Sociology: How does the system change the dynamics of human culture and relations?

Perspectives on Systems

Systems Engineering: How is the system rigorously and formally invented, verified and released?

Management / finanacial perspectives

Terms for hierarchy

[Supersystem]

- Systems
  - Subsystems
    - Components
      - Subcomponents
        - Parts

What is the subsystem hierarchy of a USAirways 80 flight to San Francisco?

[Supersystem]

- Systems
  - Subsystems
    - Components
      - Subcomponents
        - Parts

Terms for hierarchy

[Supersystem]

Systems: *perform a service with the aid of human operators and standard infrastructures*

Subsystems

Components

Subcomponents

Parts

Terms for hierarchy

[Supersystem]

Systems

Subsystems: *major portion of system that is a closely related subset of overall system function*

Components

Subcomponents

Parts

Terms for hierarchy

[Supersystem]

- Systems
  - Subsystems
    - Components
      - Subcomponents: *perform elementary functions and are composed of several parts*
        - Parts

Terms for hierarchy

[Supersystem]

- Systems
  - Subsystems
    - Components
      - Subcomponents
        - Parts: *no significant function except in combination with other parts; usually off-the-shelf.*

Now for a challenge…

Take the cellphone. Answer the question, how much does it really cost to build a cellphone, by creating a hierarchy of elements to represent a cellphone and ascribing cost to everything.

Review

- Needs Analysis is the first step in Systems Engineering
  - Elicit needs
  - Refine requirements
- Analysis continues until there is enough information to begin exploring concepts

Needs Analysis

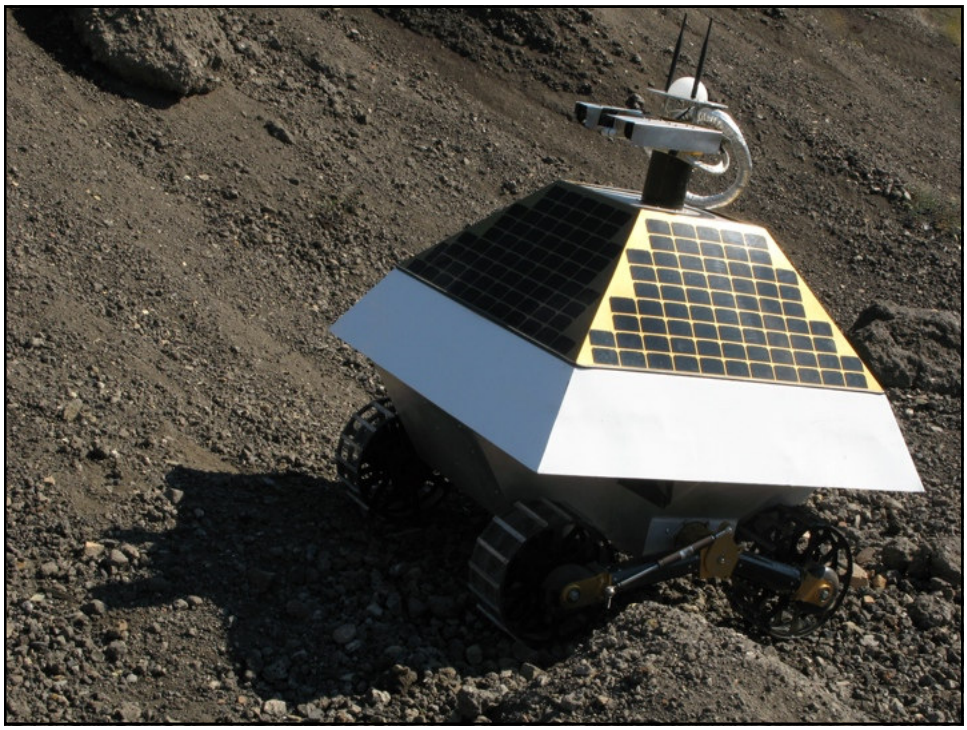
- Goal is to capture as much information as possible about user needs, wants, and domains
  - Develop and document a strategy
  - Prepare to collect and organize information
  - First focus on collection then on organization
  - Use as many sources as possible
  - Verify and cross-check everything

Needs Refinement

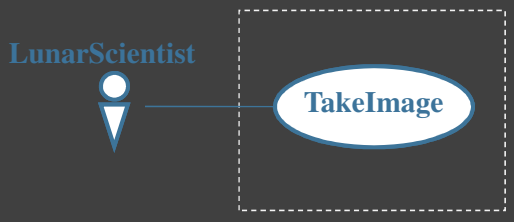
- Identify needs
- Group related requirements
- Distinguish relationships among needs
- Build appropriate levels of detail
- Rephrase using consistent terminology
- Sort out requirements from constraints
- Prioritize requirements
- Decompose requirements
- Identify/index requirements
- Gather enough information to begin design

Use Cases

- A use case shows the general cases of interaction between the system and external objects
- Use case diagrams capture a broad view of the primary functionality of a system in a manner that is easily understood by a non-technical audience

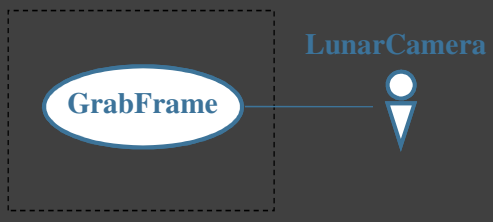


Use Cases - Lunar Exploration



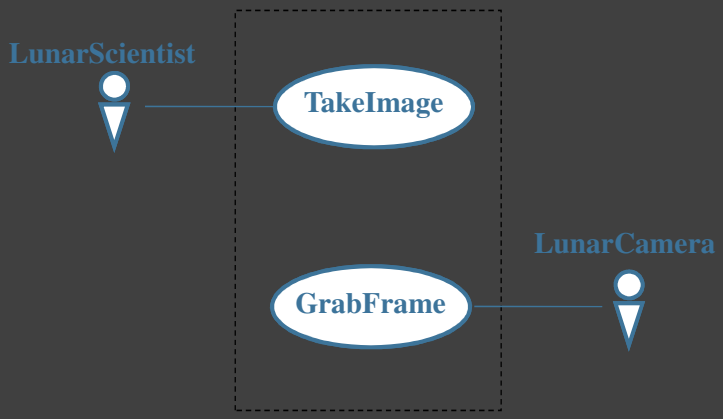
Associate Actors with externally visible system actions

Use Cases - Lunar Exploration



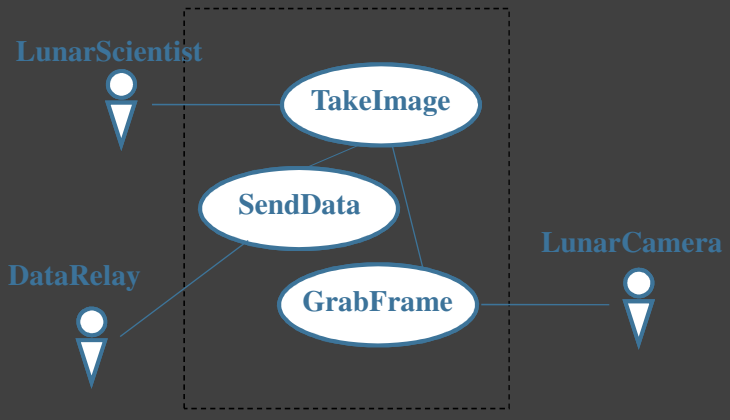
Actors are people but also devices external to the system

Use Cases - Lunar Exploration



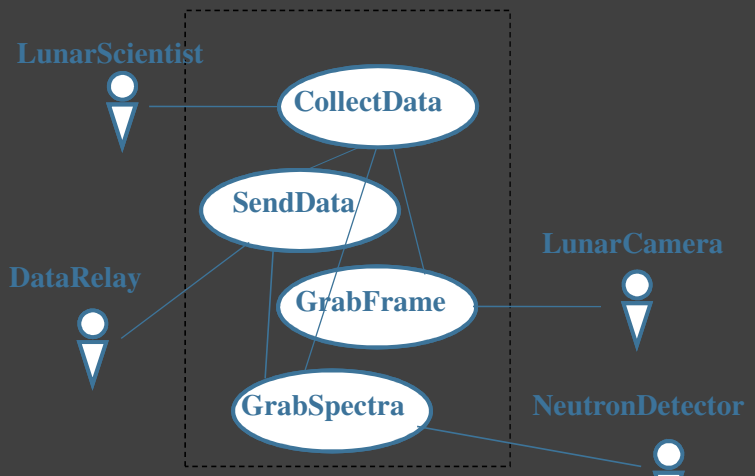
Uses cases are all externally visible

Use Cases - Lunar Exploration



Now scientists can get a picture from the moon…

Use Cases - Lunar Exploration



And data from another instrument…

Documenting Scenarios

- An effective way to capture system requirements is to:
  - Develop a Use Case model
  - Write scenarios for what should happen
  - Include nominal and exceptional scenarios

Refine Requirements

- **What must it do?** Narrow the design space!
- Plus:
- compatibility, reliability, environmental

Hardware / Software allocation

- A lesson on an important tradeoff…
- Build a balancing robot that has a human aspect ratio or narrower

Hardware / Software allocation

- Credit TRI: Ben Brown, Illah Nourbakhsh

Concept Validation Strategy

- 1 Design the right critical experiments
- 2 Run the experiments!
- 3 Iterate as much as needed

Whole System Integration & Evaluation

- Where dreams and reality diverge…

Stages

1. Test Planning & Preparation
2. System Integration
3. Developmental System Testing
4. Operational Test and Evaluation
•

Stages

1. Test Planning & Preparation
   - *First review unforeseen changes: requirements, technologies*
   - Test and Evaluation Master Plan (TEMP)
   - This step mirrors System Development:
     - System Concept <-> Test Concept
     - Functional Design <-> Test Plan
     - Detailed Design <-> Test Procedures
2. System Integration
3. Developmental System Testing
4. Operational Test and Evaluation
•

Stages

1. Test Planning & Preparation
2. System Integration
   - Proceeds linearly, adding a couple of components at a time
   - Add subsystems intelligently to avoid complex input generators
   - Therefore, start with components that have external inputs
   - Hard part: matching subsystem outputs to expected values
   - *how do we determine realistic expectations?*
3. Developmental System Testing
4. Operational Test and Evaluation
•

Stages

1. Test Planning & Preparation
2. System Integration
3. Developmental System Testing: the First Time
   - » Entire-system testing for performance vis a vis specifications
   - » Also use this as a last-ditch chance to check against needs
   - » Specify test conditions as completely as possible
   - » Generate and use *many test scenarios*
4. Operational Test and Evaluation

•

Stages

1. Test Planning & Preparation
2. System Integration
3. Developmental System Testing
   - » Dealing with Discrepancies
   - » 1: Is it really broken consistently? Try again.
   - » 2: Is it really a problem? Think.
   - » 3: How do we fix it.
   - » 4: How comprehensively must we test the fix?
   - » *The law of unintended consequences can bite!*
4. Operational Test and Evaluation

•

Stages

1. Test Planning & Preparation
2. System Integration
3. Developmental System Testing
4. Operational Test and Evaluation
   - » Validation of system design to operational requirements

Stages

1. Test Planning & Preparation
2. System Integration
3. Developmental System Testing
4. Operational Test and Evaluation
   - » Test Plan (part of TEMP)
   - » - directions for conducting the operational test
   - » - critical operation issues to be examined; metrics
   - » - *how much testing is enough?* – answer this question

Thesis

Become a systems engineering team and you will have a better outcome!