ENES 489P Hands-On Systems Engineering Projects

Introduction to Systems Engineering

Mark Austin
E-mail: austin@isr.umd.edu

Institute for Systems Research, University of Maryland, College Park
Class Web Page

See:

http://www.isr.umd.edu/~austin/ense489p.html

Class Syllabus

Outlined on the class web page ...

Assessment

Project presentation and report will count for 60% of the final grade.
Lecture 1: Getting Started

Topics:

2. Our definition of Systems Engineering.
5. End-to-end Lifecycle Development.
6. Models of Systems Engineering Development (e.g., Waterfall, Spiral).
7. Economics of development.
At the end of this lecture you should be able to answer:

1. What is systems engineering?
2. What kinds of problems does the discipline try to solve?
3. Why is systems engineering important?
4. What does a typical systems engineering lifecycle look like?
5. What are the economic consequences of failing to do proper systems engineering?
6. Are there any jobs in Systems Engineering?
SYSTEMS ENGINEER: “BEST JOB IN AMERICA”

Money Magazine

*Best Jobs in America* by Donna Rosato with Beth Braverman and Alexis Jeffries. Oct. 9, 2009
Source: MoneyOnCNMmoney.com

“Money and PayScale.com, a leading online provider of employee-compensation data, surveyed 35,000 people online about what makes a great job, they rated intellectual challenge, a passion for the work, and flexibility just as highly as security.

1. **Systems Engineer**
   - Median salary (experienced): $87,100
   - Top pay: $130,000
   - Job growth (10-year forecast): 45%
   - Sector: Information Technology

**What they do:** They're the "big think" managers on large, complex projects, from major transportation networks to military defense programs. They figure out the technical specifications required and coordinate the efforts of lower-level engineers working on specific aspects of the project.

**Why it’s great:** Demand is soaring for systems engineers, as what was once a niche job in the aerospace and defense industries becomes commonplace among a diverse and expanding universe of employers, from medical device makers to corporations like Xerox and BMW. Pay can easily hit six figures for top performers, and there’s ample opportunity for advancement. But many systems engineers say they most enjoy the creative aspects of the job and seeing projects come to life. "The transit system I work on really makes a tangible difference to people," says Anne O’Neil, chief systems engineer for the New York City Transit Authority.

**Drawbacks:** Long hours are common; project deadlines can be fierce.

**Pre-reqs:** An undergrad engineering degree; some jobs might also require certification as a certified systems engineering professional (CSEP).

Our Definition of Systems Engineering

Systems engineering is a discipline that lies at the cross-roads of engineering and business concerns.

Specific goals are to provide:

1. A balanced and disciplined approach to the **total integration** of the system building blocks with the surrounding environment.

2. A methodology for systems development that focussed on **objectives**, **measurement**, and **accomplishment**.

3. A systematic means to acquire information, and sort out and identify areas for **trade-offs** in cost, performance, quality etc....
Typical concerns on the design side:

1. What is the required functionality?
2. How well should the system perform?
3. What about cost/economics?
4. How will functionality/performance be verified and validated?

Typical concerns on the management side:

1. What processes need to be in place to manage the development?
2. What kind of support for requirements management will be needed?

Learning how to deal with these concerns in a systematic way is a challenging proposition driven, in part, by a constant desire to improve system performance and extend system functionality.
To understand a system, you really need to understand:

1. The ways in which it will be used,
2. The environment in which it will operate, and
3. The knowledge, technologies, and methods that go into making it.

For a wide range of engineering applications this problem is quite tractable.

However as systems become more complex, we need to be strategic in the way we approach design, i.e., points to the importance of:

1. System Decomposition (to simplify design).
2. Abstractions (to simplify decision making in design).
3. Formal Analysis (our understanding of system behavior needs to be right).
**Strategy:** Put original problem aside and focus on understanding the collection of subsystems that make up the original system.

Common questions include:

1. What are the subsystems and how are they connected internally?
2. How does the system interact with the surrounding environment?
System Assembly via Integration of Abstractions

System assembly through integration of abstractions

Complex System

Subsystem

Components

Observations

Increasing importance of technology

Increasing range of functionality

Increasing opportunity for reuse of lower level entities

Engineering Concerns

Increasingly heterogeneous

Increasingly homogeneous

Increasing use of abstraction

Increasing need for formal analysis
Modern buildings are:

... advanced, self-contained and tightly controlled environments designed to provide services (e.g., transportation, artificial lighting, etc.).

The design of modern buildings is complicated by:

1. Necessity of performance-based design and real-time management.
2. Many stakeholders (owners, inhabitants), some with competing needs.
3. Large size (e.g., 30,000 occupants; thousands of points of sensing and controls for air quality and fire protection.)
4. Intertwined network structures for the arrangement of spaces, fixed circulatory systems (power, hvac, plumbing), dynamic circulatory systems (flows of energy through rooms; flows of material).
Case Study: SE for Modern Buildings

Framework for interaction of architectural, structural, control, and networked embedded system design activities.

Architecture / Structural View

- External Factors
  - Occupant functionality
  - Performance metrics
  - External environment

Control View

- Occupancy demand.
- Thermal requirements
- Security requirements
- Electrical requirements
- Information requirements

System Architecture

- Design, layout and connectivity of spaces...
- Building envelope / structural design
- Feasibility of implementation
- Spatial constraints
- Control System
- Scheduling of thermal comfort, security, electrical and information services.

Networked Embedded Systems View

- HVAC components
- Security components
- Computer components
- Electrical components
- Building Networks Design
- Selection, positioning and connectivity of networked embedded systems.
- Feasibility of implementation
- Spatio–temporal constraints
## Case Study: SE for Modern Buildings

<table>
<thead>
<tr>
<th>System Level</th>
<th>Subsystem Level</th>
<th>Component Level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Architectural Concerns</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form and functionality.</td>
<td>Floor level spaces, positioning of spaces, connectivity among spaces.</td>
<td>Walls and spaces, portals, doorways, windows ...</td>
</tr>
<tr>
<td>Services, access, comfort.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Structural Concerns</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural assemblies, overall system stability</td>
<td>Frame, floor, and wall systems. Forces, deflections.</td>
<td>Beam and column elements, beam/column joints, material behavior.</td>
</tr>
<tr>
<td><strong>Electro-mechanical Concerns</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access, comfort, safety</td>
<td>HVAC, lighting, fire protection</td>
<td>Heat exchangers, pipes, elevators, escalators, sprinklers</td>
</tr>
</tbody>
</table>
Traditional engineering and systems engineering serve complimentary roles:

- **Traditional Engineering.**
  
  Focus on generation of knowledge needed to create new technologies and new things.

- **Systems Engineering.**
  
  Focus on understanding how existing technologies and things can be integrated together in new ways (...to create new kinds of systems).

So here’s the bottom line:

... systems engineers need traditional engineers, and vice versa.
Focus on:

...liaison among disciplines, supported by formal methods for systems analysis and design.
Systems are developed by teams of engineers – the team members must be able to understand one-another’s work.
Key concerns:

1. How to gather requirements that might extend beyond functionality, performance and cost (e.g., social concerns, political concerns, long-term sustainability)?

2. Partitioning of the design problem into several levels of abstraction and viewpoints suitable for concurrent development by design teams;

3. Synthesis of good design alternatives from modular components;

4. Integration of the design team efforts into a working system; and

5. Evaluation mechanisms that provide a designer with critical feedback on the feasibility of a system architecture, and make suggestions for design concept enhancement.

6. Formal methods for early validation/verification of systems.
SE at the Product Level

Use of available technology to achieve overall best cost–benefit result.

Features
- Functionality
- Cost
- Reliability
- Interfaces

Representation

New Knowledge...
New Technologies...
New Things...

Diagram showing the relationship between features and their representations, highlighting the utilization of available technology for achieving the best cost-benefit result.
SE at the Product Level

Key concerns:

1. How to describe what a product does? Can this be done formally?
2. How to describe pre-conditions for using a product?
3. How to describe a product’s interfaces?
4. How to describe various representations (visual, mathematical).
The principal products of systems engineering development are as follows:

- Requirements specification; system (logical) architecture; system (physical) design; the physical system itself.

These products are produced by the following processes:

- Requirements engineering; system architecting; systems design and integration; optimization and trade-off analysis; validation and verification.
Technical process for system architecting.....

- Needs
- Requirements Analysis
  - Function Analysis
  - System Synthesis
- Control Factors
  - Requirements
  - Functions
  - Reuse of components
- Assessment of Risks and Uncertainty
- System Architecture
  - Cost estimate
  - Performance estimate
  - Schedule
The terms system validation and verification refer to two basic concerns, “are we building the right product” and “are we building the product right?” Satisfactory answers to both questions are a prerequisite to customer acceptance.

Validation and verification concerns are a prerequisite to customer acceptance.
Pre-defined plans of development ...

... provide the discipline to keep development activities predictable and on track.

The project participants know what’s expected and when.

Interaction of technical development and engineering management processes

During the past 3-4 decades this approach to system development has served many industry sectors (e.g., aerospace) well.
Engineering/Systems Engineering Activities and Artifacts

<table>
<thead>
<tr>
<th>Engineering Activity</th>
<th>Systems Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity</td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>Requirements Analysis</td>
</tr>
<tr>
<td>Architecture</td>
<td>Function/behavior analysis</td>
</tr>
<tr>
<td>Design</td>
<td>Synthesis</td>
</tr>
</tbody>
</table>
# Systems Engineering Management Activities and Artifacts

<table>
<thead>
<tr>
<th>Management Activity</th>
<th>Systems Management</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity</td>
</tr>
<tr>
<td>Requirements Management</td>
<td>Requirements Manage-ment</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Planning of activities and tasks. Communicate compliance status</td>
</tr>
<tr>
<td>Baseline Management</td>
<td>...</td>
</tr>
</tbody>
</table>
The waterfall model works well when:

... problem and solution method are well understood, requiring no large-loop corrections to development problems.
Limitations of Waterfall Model

- Changing requirements proved to be the biggest cause of cost overruns and schedule slips in the waterfall era.

- Users were found to be unable to define the requirements of a complex system without having had hands-on previous experience with the system – A Catch 22.
Spiral model corresponds to risk oriented iterative enhancement.

Categories of risk include: technical risk, schedule risk, cost risk, programmatic risk.
Flowdown of requirements in the V-Model of system development.

Flowdown of Requirements

Stakeholder Requirements
- System Requirements
  - Subsystem Requirements
    - Component Requirements
  - Subsystem-Level Design
    - Component Design

Allocate requirements to components.

Design Problem Definition

Validate the system
- System-Level Design
  - Subsystem Requirements
  - Subsystem-Level Design
  - Component Requirements
  - Component Design

Verify the system
- System Test
  - Subsystem Test
  - Component Test

Validate the system
- Stakeholder Test

Implementation and Test
Iterate to find feasible solution

Reduce defects via reallocation of resources

Define Effectiveness Measures

Assess Available Information

Create Behavior Model

Create Structure Model

Check design for defects

Improve defects at lower level

Perform Trade-off Analysis

Create Sequential Build – and Test Plan
Funding Commitments in Product Life-Cycle

- **Cumulative Percentage**
- **Product Lifecycle**

- **Preliminary Design**
- **Commence Production**
- **Funds Committed**
- **Funds Expended**
Knowledge Gap in Systems Development

- Preliminary Design
- Commence Production
- Funds Committed
- Knowledge
- Funds Expended
- Ease of change
### Cost of Correcting Design Errors

<table>
<thead>
<tr>
<th>Project Phase</th>
<th>Bug Description</th>
<th>Relative Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Design Team</td>
<td>1</td>
</tr>
<tr>
<td>Write and Test</td>
<td>Designer</td>
<td>10-20</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA Personnel</td>
<td>70-100</td>
</tr>
<tr>
<td>Shipment to Customer</td>
<td>Customer</td>
<td>Very-expensive</td>
</tr>
</tbody>
</table>