System Engineering

Prof. Jer-Nan Juang
Adjunct Professor
National Cheng Kung University
Know Your Teacher

• **Title**
  – Director General of National Center for High-performance Computing (NCHC), Taiwan
  – Distinguished Research Associate of NASA Langley Research Center, USA
  – Adjunct Professor of National Cheng-Kung University, Taiwan
  – Adjunct Professor of Texas A&M University, USA

• **Experience**
  – NASA Langley Research Center, 1982-2005
  – Jet Propulsion Laboratory, 1979-1981
  – Computer Science Corporation, 1975-1977
  – George Washington University, Adjunct Professor, 1989-2003

• **Books**

• **Class**
  – System Identification
  – Autonomous Control
  – Space System Engineering

• **Honors**
  – AIAA Mechanics and Control Flight Award, 1993
  – AAS Dirk Brouwer Award, 1991
  – Distinguished Alumnus, National Cheng Kung University, Taiwan, 1995
Outline

• ISO (International Organization of Standard) Quality Control Process
• Definition of System Engineering
• System Engineering - Standards
• Basics of Structure
• Basics of Behavior
Process Example: A Table of Delicious Dishes
作業流程簡例

顧客

業務部門鑑別分析需求與期望

審查

接單

Y

業務（合約審查）

設計（訂規格）

外包

採購

倉庫

生管（生產計畫）

生產及檢驗

出貨檢驗

顧客

售後服務

品質文件與紀錄管制

人力資源及教育訓練

設備保養及維護

工作環境管理

儀器校正與管理

顧客滿意度調查

內部稽核

流程監控與量測

不合格品管制

矯正措施

預防措施

持續改善
ISO 9001:2000 進階
Plan-Do-Check-Act (PDCA)

★ P-D-C-A 理論可應用於所有流程

• Plan: 依據顧客需求與組織政策建立所需之各項目標與流程。

• Do: 執行這些流程。

• Check: 針對政策、目標及產品需求監控與量測諸流程與產品，並報告成果。

• Act: 採取措施以持續改善流程績效。
品質管理系統流程模式
Milestones
Reaching Business Excellence

世界級卓越事業經營典範
(World Class Business excellence model)

ISO 9004
2000年版

ISO 9001
2000年版

ISO 9002
1994年版

QC/QA model
QM model
TQM model

* ISO 14001驗證
* OHSAS 18001勞安衛驗證
* ISO 17025 實驗室品保
* Fin.Risk Management 內控制度
Homework: System Process

• Describe a system process to manufacture the product for one of the three contest problems.

• Describe a quality-control process for writing a technical report.
Definition of System Engineering
Systems Engineering

• What it is:
  It transforms customers’ needs into a description of a system architecture and design that specifies the components to be designed, implemented, and integrated.

• What it covers:
  – Design of the total package of components including their interrelationship
  – *Do not cover* design and implementation of these components

• What it requires:
  – An analysis of the customers’ needs
  – An optimal process to find a “good” solution for the system out of a multitude of possible solutions
  – A broad comprehension of different design disciplines
  – A team of engineers
  – Willingness to compromise
Holistic Thinking versus Local Thinking...

I’m sure glad the hole isn’t in our end 0 0 0
System Engineering - Standards

PDES, Inc. is an international industry/government consortium accelerating the development and implementation of ISO 10303, commonly known as **STEP** (STandard for the Exchange of Product model data).

H. P. Frisch
PDES, Inc. Members

http://pdesinc.aticorp.org/

Copyright PDES, Inc. 2002
The STEP Systems Engineering Project is coordinated through the PDES Inc., and EUROSTEP consortiums. Standards organizations collaborating with the Project are INCOSE (International Council for Systems Engineering) and OMG (Object Management Group).
STEP for Electronics Activities
PWA/PWB Durability Analysis – Boeing

- A metric to identify a failure mechanism and predict time to failure
- Provides assessment of COTS integrity
- Reduces cost of products by concurrent engineering
- Validated on numerous projects

AP210

Thermal Analysis  Vibration Analysis  Failure Assessment

H. P. Frisch
AP233 Module Sets
ISO standard - target dates

- Done - Core information management
- 2004 - Requirements, Risk & Decomposition
- 2005 - Behavior Modules & Allocation
- 2006 - Engineering Test & Analysis, V&V
- 2007 - Wrap up
From a **Unified**
To a **System**
**Modeling Language**
System UML Modeling (Notional)

Behavior Models

Safety Model

Structure Models

Performance Model
Requirements Summary

• **Structure**
  – e.g., system hierarchy, interconnection

• **Behavior**
  – e.g., function-based behavior, state-based behavior

• **Properties**
  – e.g., parametric models, continuous time variable attributes

• **Requirements**
  – e.g., requirements hierarchy, traceability

• **Verification**
  – e.g., test cases, verification results
SysML Diagram Taxonomy

- **Diagram Description**
  - description: text

- **SysML Diagram**
  - Support table

- **Structure Diagrams**
  - Diagram Name: Class Diagram - M
  - Diagram Name: Structured Class Diagram - M

- **Behavior Diagrams**
  - Diagram Name: Activity Diagram - M
  - Diagram Name: Sequence Diagram - U
  - Diagram Name: State Machine Diagram - M
  - Diagram Name: Use Case Diagram - U

- **Interaction Overview Diagram - U**
- **Timing Diagram - M**

- **Requirement Diagram - N**
- **Parametric Diagram - N**

**Legend:**
- **U** - UML 2 Diagram used w/o Changes
- **M** - Modified UML 2 Diagram with applicable SysML Extensions
- **X** - Diagram not explicitly used by SysML
- **N** - New SysML Diagram
AP233<->SysML Data Exchange

SysML Tools

XMI
XML Meta-model
Interchange

AP-233 NEUTRAL
DATA EXCHANGE
FORMAT

Electrical
CAE

Systems
Engineering

Engineering
Analysis

Algorithm
Design

Planning
Tools

Testing
Tools

SW Dev
Environment

Mechanical
CAD

H. P. Frisch
ISO 9000 + Other Process Standards

- **Problem** - Human sensible ≠ Computer sensible
  - Natural language: ambiguous, redundant, incomplete

- **Need** - Computer sensible view
  - Must be!!! - standards based (STEP-UML-XML)
  - Standards based infrastructure
    - Maintainable - Change auto-reflect into all views
    - Scalable - Enabled by representation & exchange standards
    - Identify - Non-conformance, duplications, inconsistencies
    - Multi-View - Auto-generate graphical views & documentation
    - Hanger - For knowledge & lessons learned
    - Auto-check - We’ve been burnt-here-before
Problem - Very Smart People

• Know how to do their job
  – Don’t like to be told how to do their job

• Communicate crisply
  – Tolerate ambiguity, redundancy, incompleteness

• Document once and once only!
  – Clearly to colleagues - ambiguously to others
    • Documentation mode - a management decision
      – Power-point (bad decision) has minimal archival value
    • Much can/should be auto-generated from database

• **Make it helpful & they will come!**
Basics of Structure

Summary

• Object Classes Having:
  – Name
  – Attributes
  – Functions

• Object Instances Inheriting:
  – Attributes and Having Attribute Values
  – Functions

• Part Tree (Aggregation)

• Interconnection Applied to:
  – Context
  – Assembly

• Classification

• Cardinality

Graphic Notation of Object Modeling Technique (OMT) is used
Structure and Behavior

Behavior Description
(what it does)

mapped onto

Structure Description
(how it is built)

System Description
Description of Structure: The Elemental Views
Example: Modeling a Ordinary Pocket Knife

- Two blades, a can opener/small screwdriver, a bottle opener/large screwdriver, an awl, a cork-screw, a key chain
Class Definition Box

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Attributes</th>
<th>Functions</th>
</tr>
</thead>
</table>

- **Class Name**: primary identifier given by the designer to potential users/customers of the class; be unique to distinguishable from all other classes
- **Class Attributes**: properties shared by each instance of the class
- **Class Functions**: behavior that the instances of the classes can perform
Instance Diagram

- Similar to class diagram with the exception that they describe actual objects or things and not just type definitions
# Initial Class Definition for Pocket Knife

<table>
<thead>
<tr>
<th>Rocket Knife</th>
<th>Characterize the elements that are part of the pocket knife</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Blades</td>
<td>Characterize the performance of the pocket knife in use in its environment</td>
</tr>
<tr>
<td>Number of Tools</td>
<td>An appearance attribute</td>
</tr>
<tr>
<td>Tool Types</td>
<td>Result of the assembly of the parts with interface working together</td>
</tr>
<tr>
<td>Sharpness</td>
<td></td>
</tr>
<tr>
<td>Wear</td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td></td>
</tr>
</tbody>
</table>

| Hold tool open | |
| Hold tool closed | |
### Several Instances of Pocket Knife

<table>
<thead>
<tr>
<th>Pocket Knife</th>
<th>Dave’s Knife #1</th>
<th></th>
<th>Pocket Knife</th>
<th>Dave’s Knife #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Blades</td>
<td>2</td>
<td>Number of Blades</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Number of tools</td>
<td>3</td>
<td>Number of tools</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Tool Types</td>
<td>(knife, pick, awl)</td>
<td>Tool Types</td>
<td>(knife, can opener, bottle opener)</td>
<td></td>
</tr>
<tr>
<td>Sharpness</td>
<td>60%</td>
<td>Sharpness</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>Wear</td>
<td>40%</td>
<td>Wear</td>
<td>23%</td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>red</td>
<td>Color</td>
<td>blue</td>
<td></td>
</tr>
</tbody>
</table>

| Pocket Knife  | Jim’s Knife   |  | Pocket Knife  | Carol’s Knife |
|---------------|---------------|---------------------|---------------|
| Number of Blades | 2           | Number of Blades    | 2             |
| Number of tools     | 3             | Number of tools     | 3             |
| Tool Types          | (knife, pick, awl) | Tool Types          | (knife, screwdriver, awl) |
| Sharpness           | 90%           | Sharpness           | 90%           |
| Wear                | 23%           | Wear                | 23%           |
| Color               | red           | Color               | blue          |
Pocket Knife Disassembled

• Two plastic side panels

• A metal case built from plates, springs and rivets, six tools, and a key chain

• It is secured with a rivet and hinges with three rivets
Part Tree for Pocket Knife

Pocket Knife

Metal Knife Assembly

- Metal Knife Case
- Hinge Rivet
- Metal Side Plate

Plastic Side Panel

- Awl
- Can Opener
- Channel Plate
- Spring
- Assembly Rivet

- Large Knife
- Small Knife
- Cork Screw
- Screwdriver
- Bottle Opener
- Key Chain

*At each ♦, there is an assembly*
Part Tree for Six Tool Pocket Knife

Six Tool Pocket Knife
- Hold Tool Open
- Hold Tool Closed

Metal Knife Assembly
- Metal Knife Case
  - Hinge Rivet
- Awl
  - Wear
- Large Knife
  - Wear
  - Sharpness
- Can Opener
  - Wear
  - Torque
- Cork Screw
  - Wear
- Small Knife
  - Wear
  - Sharpness
- Key Chain
  - Screwdriver
  - Bottle Opener
  - Wear
  - Torque
- Spring
  - Stiffness
- Assembly Rivet
- Plastic Side Panel
  - Color
  - Logo
Cardinality and Conditions

- **Class** - Exact one
- **Class** - Many, zero or more
- **Class** - Optional, zero or one
- **Class** - One or more
- **Class** - Numerically specified

1+, 1-2, 4
Part Tree for Six Tool Pocket Knife with Cardinality

Pocket Knife

Metal Knife Assembly

Metal Knife Case

Hinge Rivet

2

Awl

Can Opener

Large Knife

Cork Screw

Small Knife

Screwdriver Bottle Opener

Plastic Side Panel

Key Chain

Metal Side Plate

Channel Plate

Spring

Assembly Rivet

2

3

2
Classification Tree for Pocket Knife

Pocket Knife  
Store tools in handle  

Switchblade Knife  
Gravity Knife  
Lockblade Knife  
Six Tool Pocket Knife

Superclass  
Kind of, OR Tree  
Subclasses
Types of Tools for Class Tool

- Tool
  - Chisel
    - Phillips Screwdriver
  - Awl
    - Pliers
  - Large Knife
  - Can Opener
  - Small Knife
  - Cork Screw
  - Screwdriver Bottle Opener
  - Hook
  - Wood Saw
  - Reamer
She reached into her right pocket and took out the pocket knife. She opened the large blade with her thumbnail against the closing force of the knife and the knife held the blade open. She grasped the knife firmly and picked up the work piece in her other hand and whittled it, by applying force to the knife which transferred the force to the workpiece, to the desired shape. She put down the work piece, closed the knife against the holding force of the knife and the knife held the blade closed, and then put it back into her right pocket.
Assembly Interconnection for Metal Knife Case

- The object diagrams for structure capture the parts, the choices among parts, and how the parts are to assemble, without designing the parts.
Homework: Structure

- Observe your surroundings
  - Create class definitions for six things in your environment.
  - Create assembly or interconnection diagrams for these six things.
  - Choose two of the classes and show how they are related using classification.