EECE 276
Embedded Systems

Techniques:
Structured and OO Analysis, UML
Structured Analysis

- Modern Structured Analysis (Yourdon):
  » “Recipe” for capturing and structuring system requirements and designs – *Functional* aspects
  1. Environment Model (*analysis aspect*)
  2. Behavioral Model
  3. Implementation Model

Environment model:
- Context diagram: System interfaces
- Event List: What are the significant events?
- Natural Language text
SA: Context Diagram Example

Aircraft Inertial Measurement Unit
Main system interfaces
Data flows
Object-Oriented Analysis and Design

- Key idea: System is built from a set of collaborating “objects”
- OOA: Process to identify key players (“actors”) and ingredients (“classes”) in the system, their functions and interactions with the environment, and their static structure
- Visual language for OOAD:
  - Unified Modeling Language (UML):
    - Standardized by OMG
    - Not formally defined
- OOA analysis tool: Use-Case Diagram
  - System boundary
  - Actors
  - Functions
OOA: Use-case Diagram

System: IMU

Actors: Temp Sensors, Accelerometers, Clock, Visual Display Unit, Gyroscopes, Resolvers

Use cases: Compute and display true position, Control and compensate, Detect and display errors
**OOA: Class diagram**

**Class:** Prototype for objects. Has attributes and methods.

**Association:** Represents a relation between classes
Writing good requirements

- Pyramid structure (hierarchical) – with systematic numbering (1., 2., 3., 1.1, 1.2, 1.3, 1.1.1, 1.1.2, …)
- Qualifiers of “good” requirements:
  - Correct
  - Unambiguous
  - Complete
  - Consistent
  - Ranked (for importance)
  - Verifiable
  - Modifiable
  - Traceable
Requirements validation

- **Verification**: Is the design correct w.r.t requirements?
- **Validation**: Are the requirements correct w.r.t to the user’s expectations?
- **Validity**: functions support the user’s needs?
- **Consistency**: any conflicts?
- **Completeness**: all user’s needs are included?
- **Realism**: can we (afford to) do it?
- **Verifiability**: can they be checked?
Requirements validation

- Reviews
- Manual analysis
- Prototyping/executable model
- Test-case generation for requirements
- (Automatic) Consistency analysis