Administrative Stuff

- Visio Professional on its way
  - One per team - install on a personal machine
- Books
Contract Summary

- Overall, very good—some excellent
- “What” versus “how” (“how” now…)
- Diagram/section numbering schemes
  - Labels as persistent identifiers, tokens, or “icons” for an item
- Schedules
  - Milestones
  - Risk Analysis (Assessment)
Schedule

- **Due on Tuesday**
  - Design Specification Document (w/ Schedule III)
- **Suggested schedule**
  - Before Thursday
    - DIR: sketch out design w/ diagrams
  - Thursday in place of class (no class)
    - Team: hold a preliminary design review
  - Friday-Sunday
    - Team: work on Design Specification Document
  - Monday
    - Team: meet for final review of document
Design I

Design in general
The Design Specification Document
Architectural Design Breakdown

*Note: this is tailored to COMP 145*

- **High-Level Design**
  - System structuring
  - Control modeling
- **Detailed Design**
  - Modular decomposition
Graphical Models (General)

- Structural model (static)
  - Subsystems and modules
- Run-time process model (dynamic)
  - Run-time process/thread organization
  - E.g., stand-alone server processes, spawned processes, etc.
- Interface model
  - Public interface specification
High-Level Design

System structuring
Control modeling
System Structuring

- Overall static architecture
  - Provide context—show all of the pieces
  - (~User-Level Requirements)
- Decompose system into set of interacting subsystems
  - How are the subsystems distributed?
  - How do they manage/share data?
  - High-level interfaces
Structural Model (Static)

The HiBall Tracking System

A. Main

B. Initialization (Acquisition)

C. Chooser

D. Kalman filter

E. VRPN Server

F. Param Files

Ceiling info

HiBall and Ceiling info

Pose estimates

LED choice

Pose estimates To VRPN Client

HiBall and Ceiling info

Pose estimates
Control Models (Typical)

- **Centralized**
  - One subsystem is the master controller
  - Call-return (sequential) vs. manager (concurrent)

- **Event-driven**
  - Subsystems respond to external events or triggers
  - Broadcast (message) vs. interrupt-driven
Centralized Control: Call-Return

Main program

Routine 1
  - Routine 1.1
  - Routine 1.2

Routine 2

Routine 3
  - Routine 3.1
  - Routine 3.2
Event-Driven Control: Broadcast

Sub-system 1
Sub-system 2
Sub-system 3
Sub-system 4

Event and message handler
Event-Driven Control: Interrupts

Interrupts

Interrupt vector

Handler 1

Process 1

Handler 2

Process 2

Handler 3

Process 3

Handler 4

Process 4
Detailed Design

Modular decomposition
(Object-Oriented)
Modular Decomposition (DD)

- No hard/fast rules about granularity
  - Lowest levels would be modules
  - Top level is the system
  - In between are subsystems
  - All diagrams could apply at all levels

- For us
  - About same level as Contracts
  - Suggest OO decomposition/design
  - Read Coggins/Brooks Chapter 7
Object Models (OO-Centric)

- Encapsulation and inheritance
- Useful object-oriented traits
  - Class <-> module
  - Task decomposition
    - Minimize communication and dependencies among modules
  - Separate architecture from implementation
    - Data and process hiding
Object Model Diagrams

F. Param Files

F.1 Ceiling
- LED Array
- Panel list
- KF param array
- Load Ceiling
- Save Ceiling
- Mark bad LED

F.2 HiBall
- Camera models
- Correction tables
- Bad view list
- Load HiBall
- Mark camera bad

Common class?
Other DD

- Intra-module data-flow diagrams
  - Same as with Contract, but lower level and specific to objects/classes
- Inter-module control diagrams
Design Specification Document

- Introduction
- High-Level Design
  - Structure and control diagrams
- Detailed Design
  - Object model diagrams (equivalent)
- Requirements Traceability
  - Matrix w/ requirements vs. design
- Schedule
  - New and improved milestones

(See example document on-line)