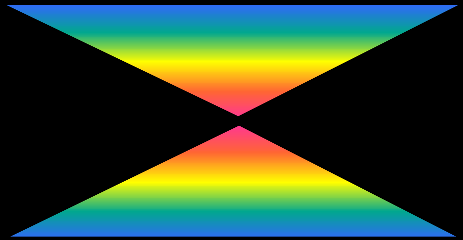
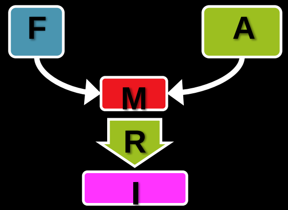
Summary of the Course

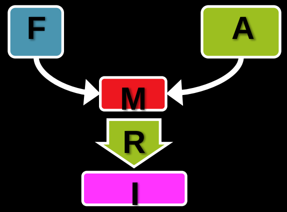
What, Why, When

Design Methods

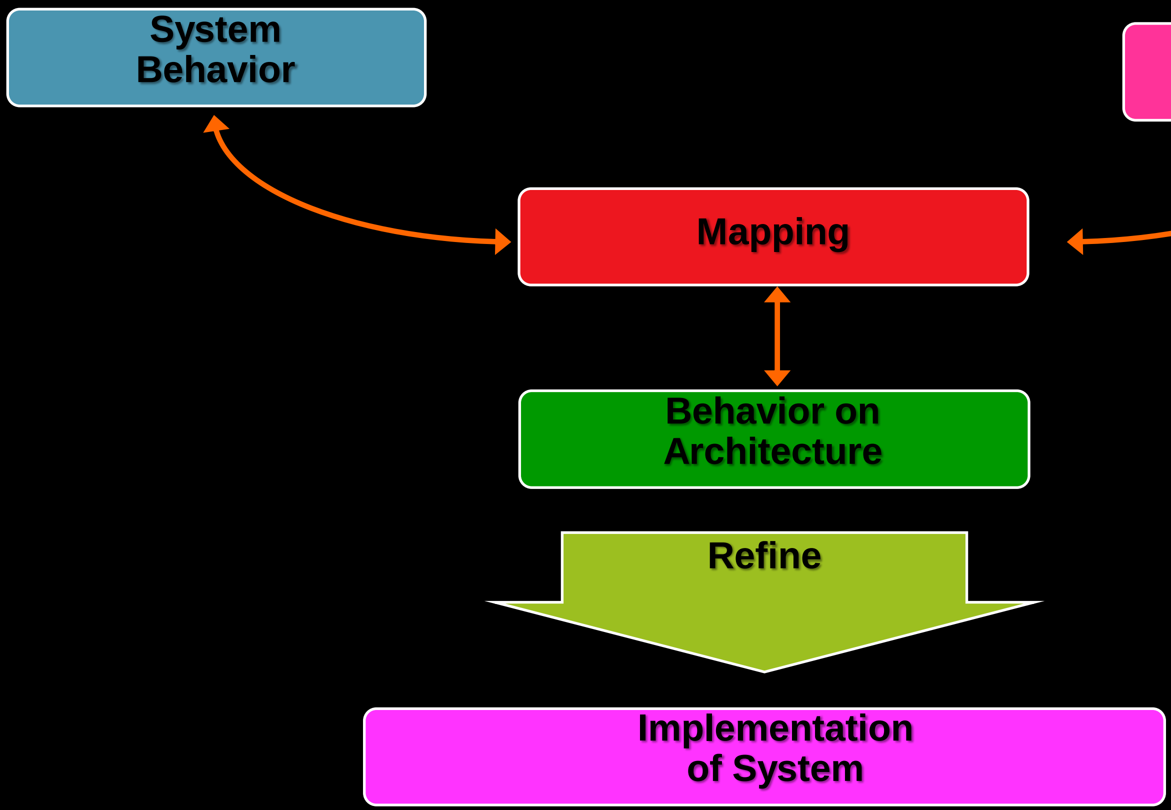
- Platform-Based design and Successive Refinement principle
- Communication-based design thru successive refinement as paradigm for re-use and correct by construction method







The Y-chart view of the Course



System Behavior

System Behavior

- Models of Computation as paradigm for system level behavior capture
  - FSM
  - Synchronous Languages
  - Data-flow
  - Petri-net
  - Discrete Event
  - Tagged Signal Model
  - Metropolis Meta-Model

Tools

- Ptolemy II
- LabView
- Simulink
- Metro II

Architecture

System Architecture

- Micro-processor based architectures
- Architectural Services
- Protocols and interconnects

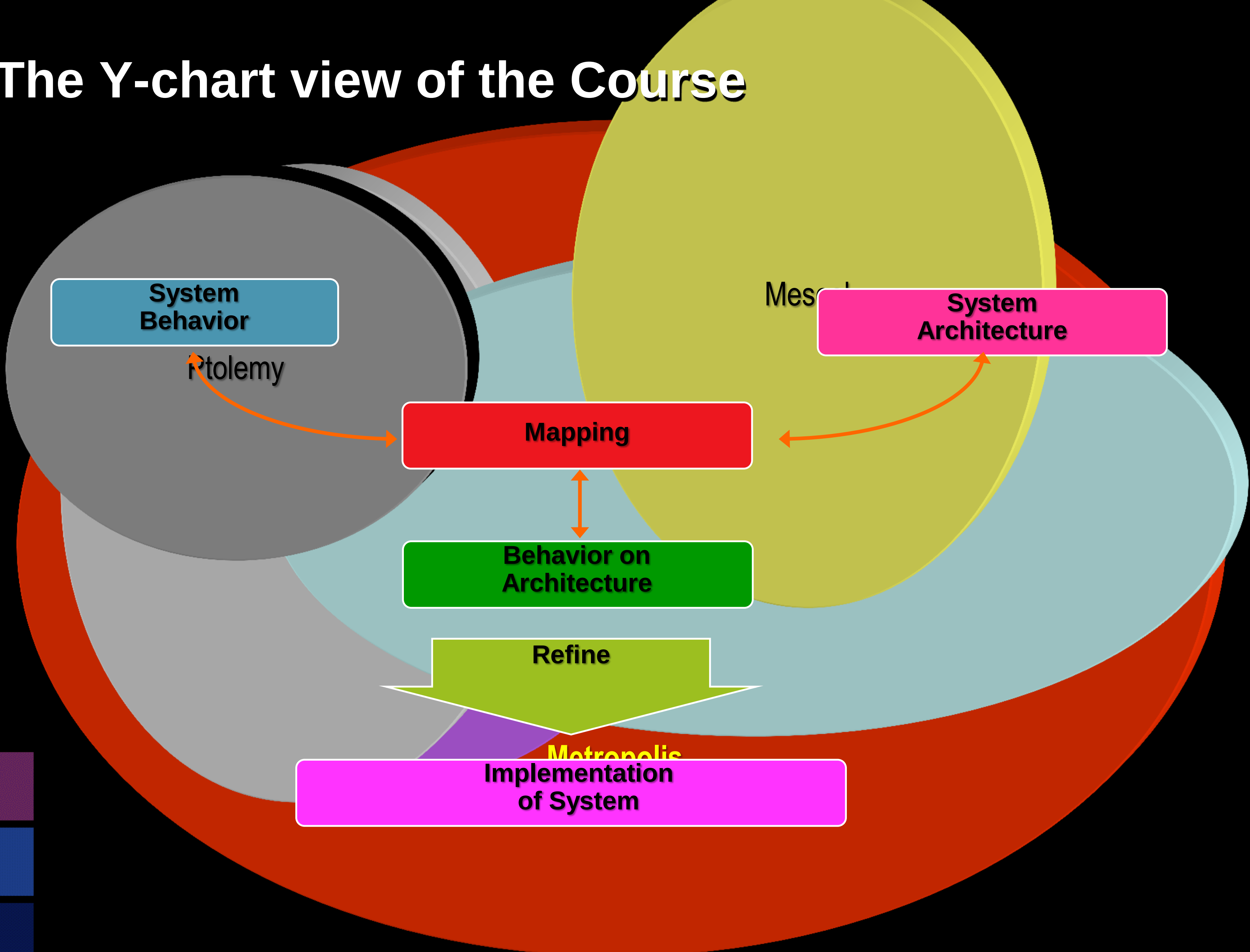
Mapping

- Scheduling Algorithms and RTOSes

Distributed Systems

- Auto Design Flow:
  - Issues related interconnect networks (CAN, FlexRay)
  - Real time OS and Scheduling Issues
  - Autosar
- Energy Efficient Buildings

The Y-chart view of the Course



EE249 Fall'12: The Fronteer

1. Introduction	Design complexity, examples of embedded and cyber-physical systems, traditional design flows, Platform-Based Design, design capture and entry
2. Functional modeling, analysis and simulation	Overview of models of computation. Finite State Machines, Process Networks, Data Flow, Petri Nets, Synchronous Reactive, Hybrid Systems. Tagged Signal Model. Simulation of heterogeneous systems. **Compositional methods and Contract-based Design.**
3. Architecture and performance abstraction	Definition of architecture, examples. Real time operating systems, scheduling of computation and communication.
4. Mapping	Definition of mapping and synthesis. Code generation from Simulink and SysML models. Design Space Exploration and Metropolis. **Mapping and Contracts.**
5. Verification	Validation vs. Simulation. Formal methods. **Horizontal and Vertical Contracts. Interface automata and assume-guarantee reasoning.**
6. Applications	Automotive: car architecture, communication standards (OSEK/AUTOSAR), scheduling and timing analysis. Building automation. **Aircraft electric power system.**

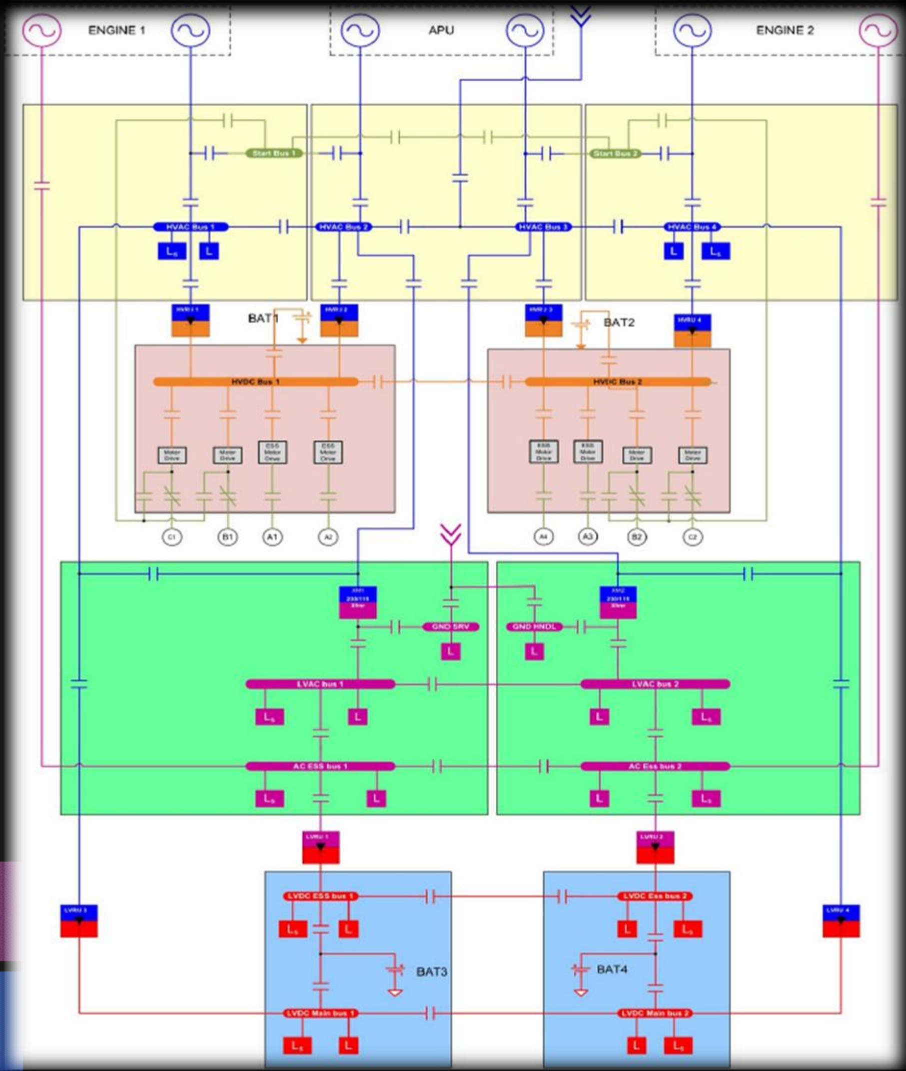
Contract-Based Design: an all-encompassing framework

The key to Platform Based Design

- Components
- Composition rules
- Refinement rules
- Abstraction rules

Contracts

Aircraft Electric Power System (EPS) Design



Single Line Diagram from Honeywell Patent

- A complex CPS in modern aircrafts
  - **Actuation and control** are largely implemented **with electrical and electronic components**
  - Large number of **hardware** subsystems
  - More interactions with the **embedded control software**
- EPS design is still a derivative **heuristic process** ("V-diagram")
  - **Lack of formal** specifications
  - Inability to model interactions between **heterogeneous components**
  - **Inefficient** implementations, **delays, cost** overruns

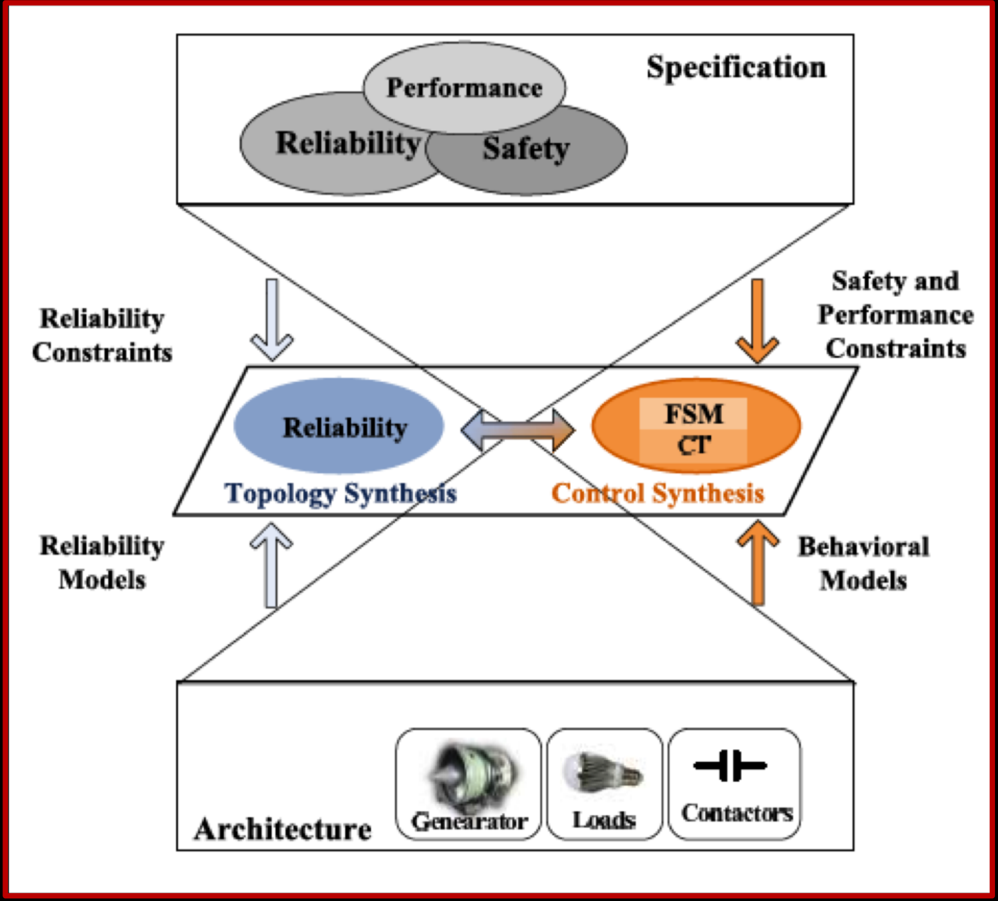
Contract-Based Design Methodology

Top-Down Phase: Formalize requirements and associate them to system entities (vertical contracts):

- Mixed integer-linear arithmetic constraints
- State machine diagrams
- Sequence diagrams
- Linear Temporal Logic
- Signal Temporal Logic
- Probabilistic constraints

Bottom-Up Phase: build library of hierarchical executable models

- **Horizontal contracts** specify legal compositions
- **Vertical contracts** define when models are faithful representations of the physical elements



Contract-Based Design Methodology

Mapping

Optimization problem where we search for candidate configurations that satisfy the conjunction of all contracts

- **Topology Synthesis**: generate optimal topology w.r.t. number of component and cost
  - Guarantee desired reliability
  - …under assumptions on the control protocol
- **Control Synthesis**: generate the controller state machine and clock frequency to drive contactor
  - Guarantee that critical loads are always powered
  - …under assumptions on the EPS topology