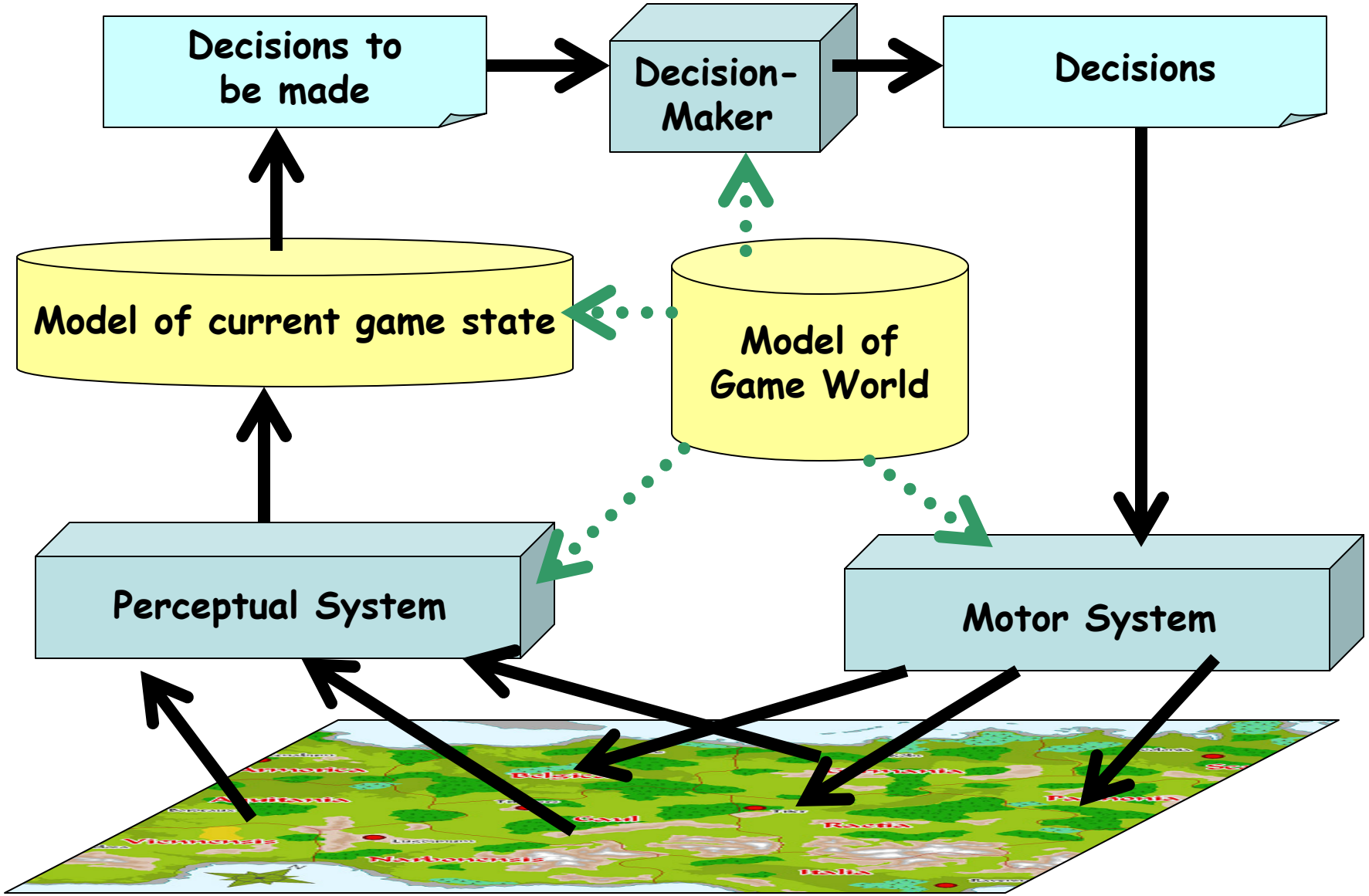
# Exploration

CS395 GAI

Spring, 2005

# Abstract Architecture strategy-game AI



# Sensing the World

- Need hooks into the simulator to gather information about game state
  - First step in building world model
- Design issues
  - How much abstraction to introduce?
    - If you're also the world designer, can align simulation and AI perception quite closely
  - How much to record, over what period?
    - What is needed to support decision-making, learning?

# Modeling the world

- Perception tells you what is happening
- Must be assessed in terms of
  - What your goals and plans are
  - What your opponents/allies goals and plans are
- Assessment process identifies
  - Threats
  - Progress
  - Opportunities
- Assessment process provides *situational awareness*

# Changing the World

- Need hooks into world simulator
- Design tradeoffs
  - Controlling continuous changes
    - Factored out in turn-based designs
    - Require tight, often autonomous, feedback control
  - Need to report consequences
    - Actions don’t always succeed

# Making Decisions

- Some dimensions of decision-making
  - Deliberative versus Reactive
  - Centralized versus Local
  - Hierarchical versus Flat
  - Learned versus Hard-wired
- Mostly orthogonal
- Often used in mixtures
- Trade-offs can be subtle

# Deliberative versus Reactive

- Deliberative ➔ Construct a plan, then execute it
  - Plans often involve multiple steps, including sensing, conditional branching
  - Enables optimization, but can be slow
- Reactive ➔ Just do something, based on sensors
  - Provides rapid, reflex action
  - Can lead to silly behaviors if unanticipated situations arise

# Centralized versus Local

- Centralized → AI structured as computer player
- Local → AI structured as models for what units should do in the simulated world
- Local often easier to implement
  - Combinatorics of explicit coordination can become nasty
  - Gradient methods used to provide simulation of coordination

# Hierarchical versus Flat

- Hierarchical → Use structure of the problem for divide-and-conquer
  - Example: Echelon distinctions in military → different levels of AIs
    - Company, Squad, individual AIs
  - Factors decision-making to make it more manageable
  - Imposes extra overhead of communication between layers

# Learned versus Hard-wired

- Hard-wired
  - Fast runtime execution, guaranteed understanding of local behavior
  - Brittle, can be too predictable for player, non-local interactions hard to debug
- Learned
  - Can adapt to player, provide surprises
  - Slower runtime execution, higher memory load, can lead to unpredictable, degenerate behaviors

# Strategies for making decisions

- Goals can be achieved in many ways
- Situations often allow many actions
- How to choose?
  - Generate a set of alternatives
  - Compute numerical evaluation of each of them
  - Pick the best
    - Or, for variability in play, pick randomly with bias proportional to perceived quality of choices

# Exploration

- Goals of exploration:
  - Find territory to expand into
  - Find your neighbors
  - Find out how soon you need a navy
  - Find exploitable terrain for defense

# How does the FAP do it?

- See Phil Houk's technical report
  - http://www.cs.northwestern.edu/publications/techreports/2004_TR/NWU-CS-04-29.pdf

# How to explore the whole world?

- Need to build a navy
  - Find coastal sites for cities
  - Develop technology to build ships
    - Map-making, …
    - Optimize order of technological advances?
    - Strategies for ferrying units
- Build more explorers
  - Manage more explorers
    - Send off in different directions
    - Distribute across land masses
- Build infrastructure
  - Roads, to get explorers to embarkation points
- Get alliances to share maps