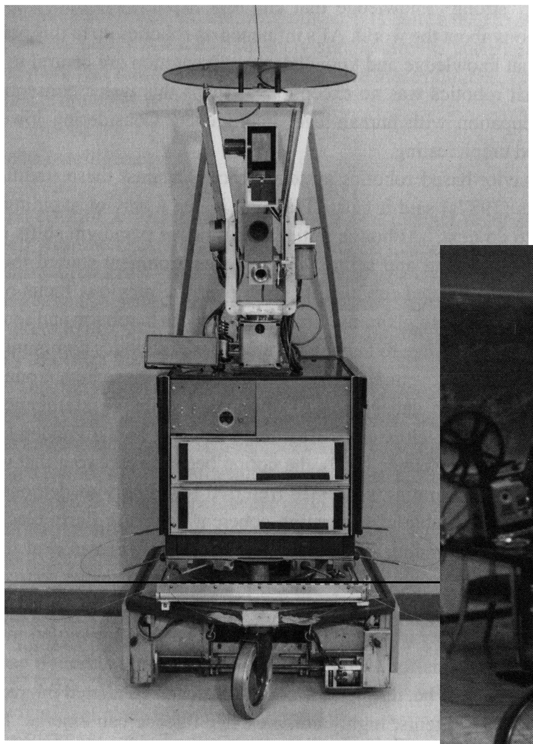
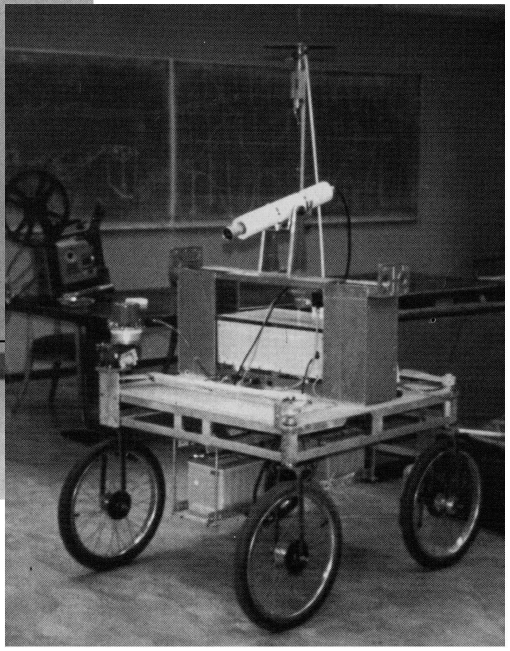
Deliberation

How do robots think?

Deliberative intelligence

- Generating goals and intents
- Selecting or planning how to best meet those goals and intentions
- Implementing the specific actions and resources to carry out the plan
- Monitoring the execution of the plan and replanning if it is not working


Shakey

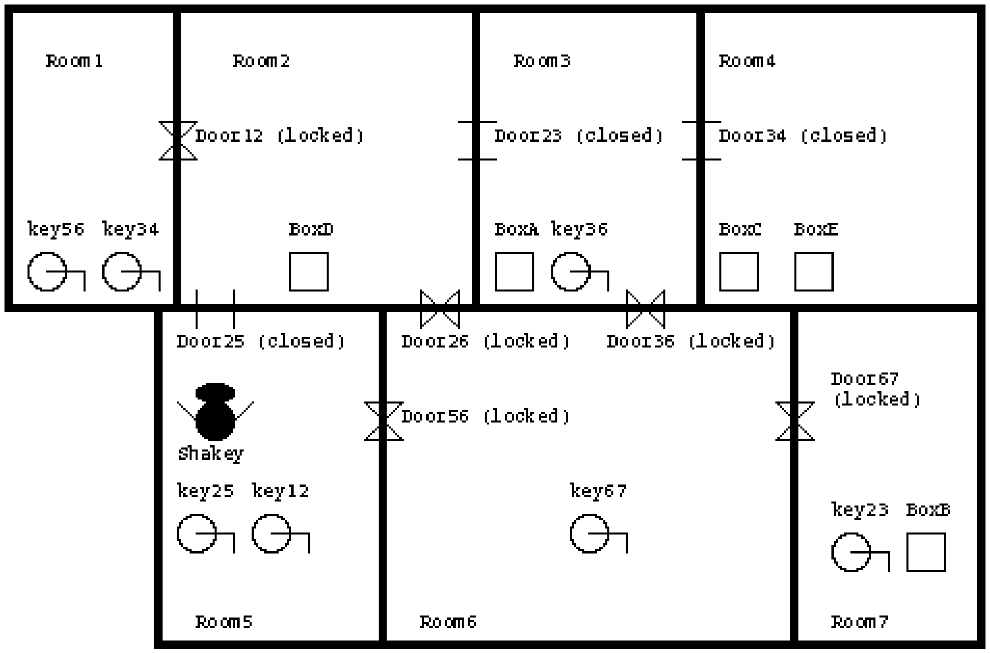

Stanford Cart

Objectives

- Solve a simple deliberation problem using Strips, given an initial world model, operators, difference table, and difference evaluator. Show the state of the world model after each step.
- Describe the Mission Planner, Navigator, and Pilot organization of the Nested Hierarchical Controller.
- Define the physical symbol grounding problem and anchoring; give an example

Planning-Based Approach to Robot Control

- Job of planner: generate a goal to achieve, and then construct a plan to achieve it from the current state.
- Must define representations:
  - Representation of states: data structure describing current situation
  - Representation of goals: what is to be achieved
  - Representation of actions: programs that generate successor state descriptions, defining preconditions and effects
  - Representation of plans: solution is a sequence of actions



First-order logic and theorem proving enable planning of strategies from start state to goal

STRIPS Representation – States and Goals

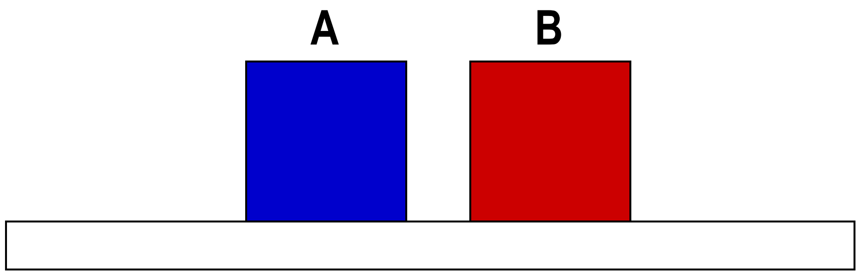
- States: Conjunctions of function-free ground literals (i.e., predicates applied to constant symbols, possibly negated)
  - Example: At(Home) Have(Milk) Have(Bananas)
  - Common assumption: If state description does not mention a given positive literal, then the literal is assumed to be false
- Goals: Conjunctions of literals, which can include variables
  - Example: At(x) Sells(x,Milk)
  - Assumption: variables are existentially quantified

STRIPS Representation -- Actions

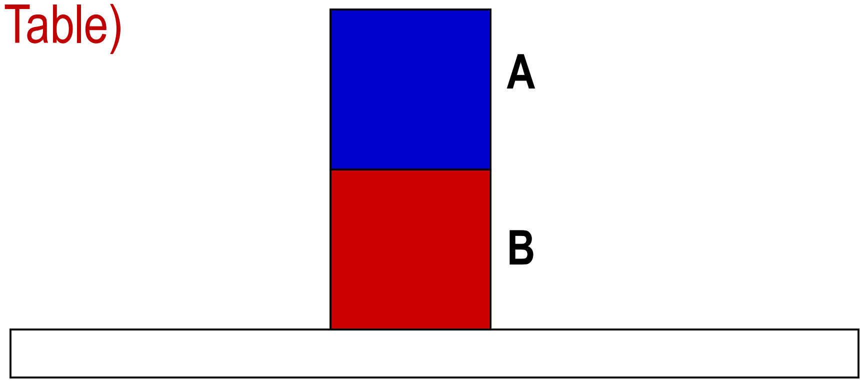
- STRIPS operators consist of:
  - Action description: name for what an agent does
  - Precondition: conjunction of positive literals (atoms) saying what must be true before operator can be applied
  - Effect: conjunction of literals (positive or negative) that describes how the situation changes when operator is applied
    - Organize effect as:
      - ADD LIST
      - DELETE LIST

Simple Example of STRIPS Blocks-World Problem

- Goal State: ON(A,B)
- Start state: ON(A, Table); ON(B, Table); EMPTYTOP(A); EMPTYTOP(B)
- Operator:
  - MOVE(x,y):
    - Preconditions: ON(x,Table); EMPTYTOP(y)
    - Effect:
      - Add-List: ON(x,y)
      - Delete-List: EMPTYTOP(y); ON(x,Table)



Start State



Goal State

Shakey's STRIPS World

- Types of actions Shakey can make (at least in simulation):

Move from place to place:

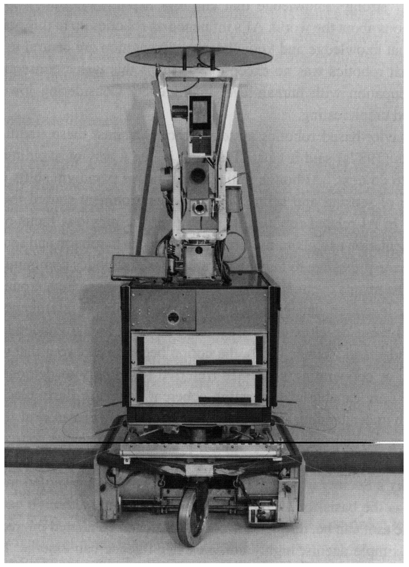
Go(x, y):

- Precondition:
  - At(Shakey,x)
  - In(x,r) $\wedge$ In (y,r)
  - On(Shakey,Floor)
- Effect:
  - Add-List: At(Shakey,y)
  - Delete-List: At(Shakey,x)

Push movable objects:

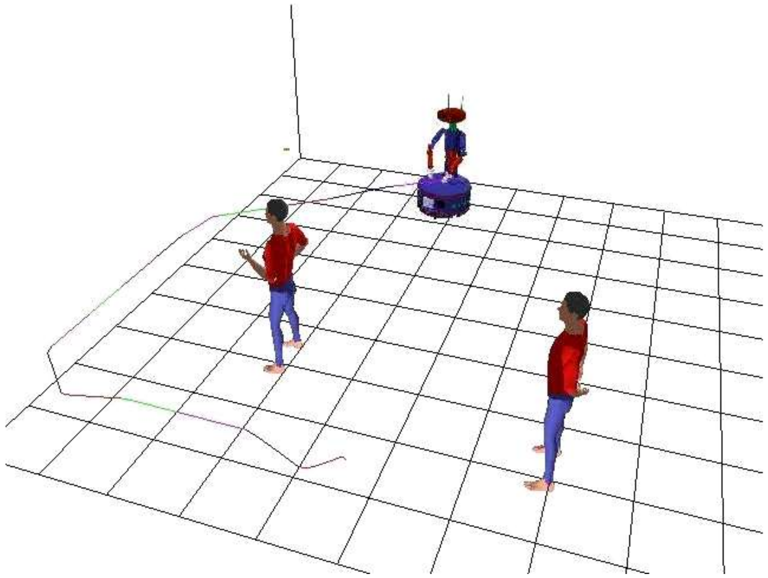
Push(b, x, y):

- Precondition:
  - Pushable(b)
  - At(b,x)
  - At(Shakey,x)
  - In(x,r) $\wedge$In (y,r)
  - On(Shakey,Floor)
- Effect:
  - Add-List: At(b,y)
    At(Shakey,y)
  - Delete-List: At(b,x)
    At(Shakey,x)



STRIPS Representation -- Plans

- A *Plan* is a data structure consisting of:
  - A set of plan steps.
    - Each step is one of operators for the problem.
  - A set of step ordering constraints. $S_i \prec S_j$
    - Ordering constraints specify that an action has to occur sometime before another action
  - A set of variable binding constraints.
    - Variable constraints are of form v=x, where v is variable at some step, and x is either a constant or a variable
  - A set of causal links.
    - Record the purpose of the steps in the plan.
    - E.g., purpose of S_i is to achieve the precondition c of S_j.

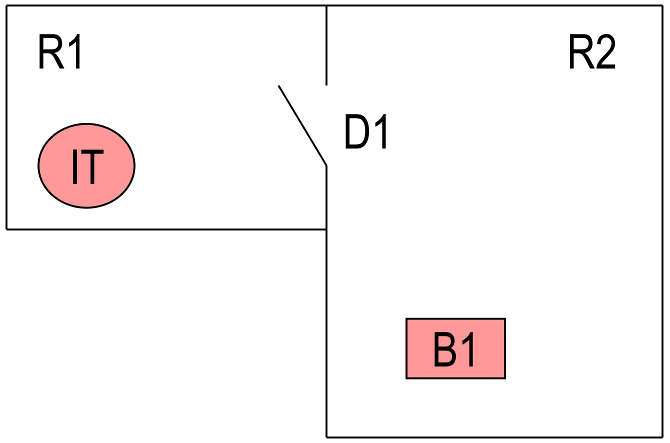


How to Develop a Plan in STRIPS?

- Use “means-ends analysis”
  - If robot is not at goal, then measure the distance (or difference) between the robot’s current state and its goal state
  - Find an operator that the robot can take to reduce this distance (difference), and select it if can bind the variables in preconditions to make preconditions true
  - Make the first false precondition of this operator the new “subgoal”, remembering old goal by pushing on stack
  - Recursively reduce difference by repeating above.
  - When all preconditions for operator match, add the operator to the plan stack and update world model. Continue until we find an operator that can be executed in robot’s current state
  - When done, pop the actions off the stack to create the plan from start to goal.

An Example of Building a Plan Using STRIPS

- The world:
  - Two rooms R1 and R2
  - One door D1 connecting R1 and R2 and is open
  - One robot IT in room R1
  - Another movable object B1 in R2
- The task:
  - Move IT from room R1 to room R2



- The robot's knowledge can be represented by the following predicates:
  - INROOM(x, r) : x is in room r, where x is IT or B1, r is R1 or R2
  - NEXTTO(x, t) : x is next to t, where x is IT or B1, t is D1, B1, or IT
  - STATUS(d, s) : d is D1, s is OPEN or CLOSE
  - CONNECT(d, rx, ry) : d is D1, (rx is R1 and ry is R2), or (rx is R2 and ry is R1)

An Example of Building a Plan Using STRIPS (Cont.)

- Initial State:
  - INROOM(IT, R1)
  - INROOM(B1, R2)
  - CONNECT(D1, R1, R2)
  - CONNECT(D1, R2, R1)
  - STATUS(D1, OPEN)

- Goal State
  - INROOM(IT, R2) ✓
  - INROOM(B1, R2)
  - CONNECT(D1, R1, R2)
  - CONNECT(D1, R2, R1)
  - STATUS(D1, OPEN)

Operator	Preconditions	Add-list	Delete-list

OP1
GOTODOOR(IT,dx)
	INROOM(IT,rk)
	CONNECT(dx,rk,rm)
		NEXTTO(IT,dx)	

OP2
GOTHRUDOOR(IT,dx)
	CONNECT(dx,rk,rm)
	NEXTTO(IT,dx)
	STATUS(dx,OPEN)
	INROOM(IT,rk)
		INROOM(IT,rm)	INROOM(IT,rk)

An Example of Building a Plan Using STRIPS (Cont.)

The logical difference between the initial state goal state is simply INROOM(IT, R2) = FALSE

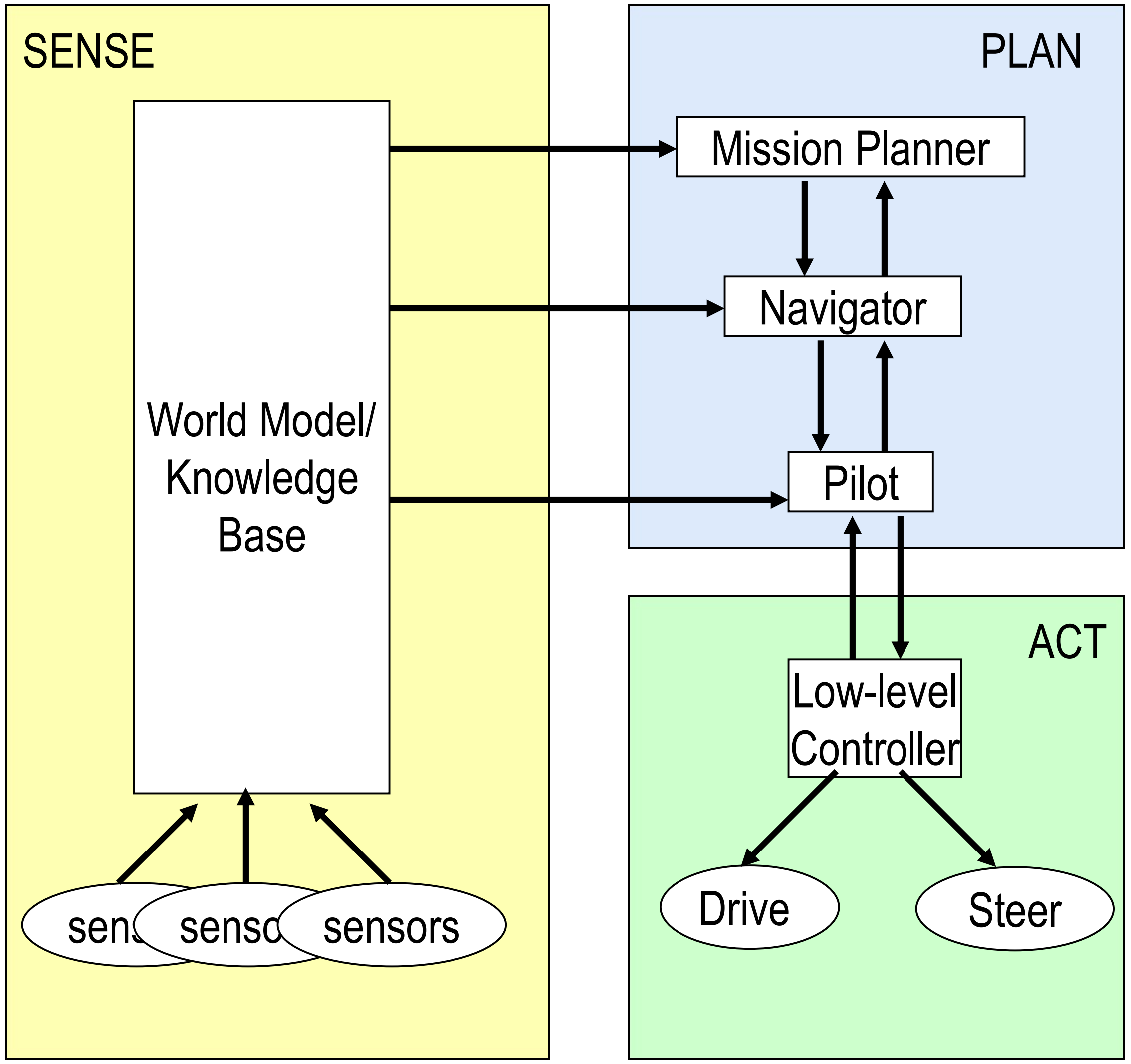
Subgoal:	**Plan Stack:**		**Preconditions:**	
INROOM(IT, R2)	GOTHRUDOOR(IT, D1)	2	CONNECT(D1, R1, R2) ← OK	
			NEXTTO(IT, D1) ← FAILED	✓
			STATUS(D1, OPEN) ← OK	
			INROOM(IT, R1) ← OK	
NEXTTO(IT, D1)	GOTODOOR(IT, D1)	1	INROOM(IT, R1) ← OK	✓
			CONNECT(D1,R1,R2) ← OK	✓

The plan is GOTODOOR(IT, D1) and then GOTHRUDOOR(IT, D1)

Symbol Grounding Problem

- The process of assigning symbols to objects or concepts
- Robotics concentrates on **physical symbol grounding**, where data acquired about the physical world from sensors are assigned to symbols
- Physical symbol grounding involves **anchoring**: assigning labels to perceptual inputs and thus anchoring perception to a common frame of reference that can be used by the deliberative functions of the robot
- Anchoring is usually accomplished through one of two general strategies:
  - Top-down object recognition that matches existing symbols in the robot’s knowledge representations
  - Bottom-up object recognition, where the robot senses an unknown object, determines it is of interest, declares it to be an object, and assigns it a label

Nested Hierarchical Controller (NHC)



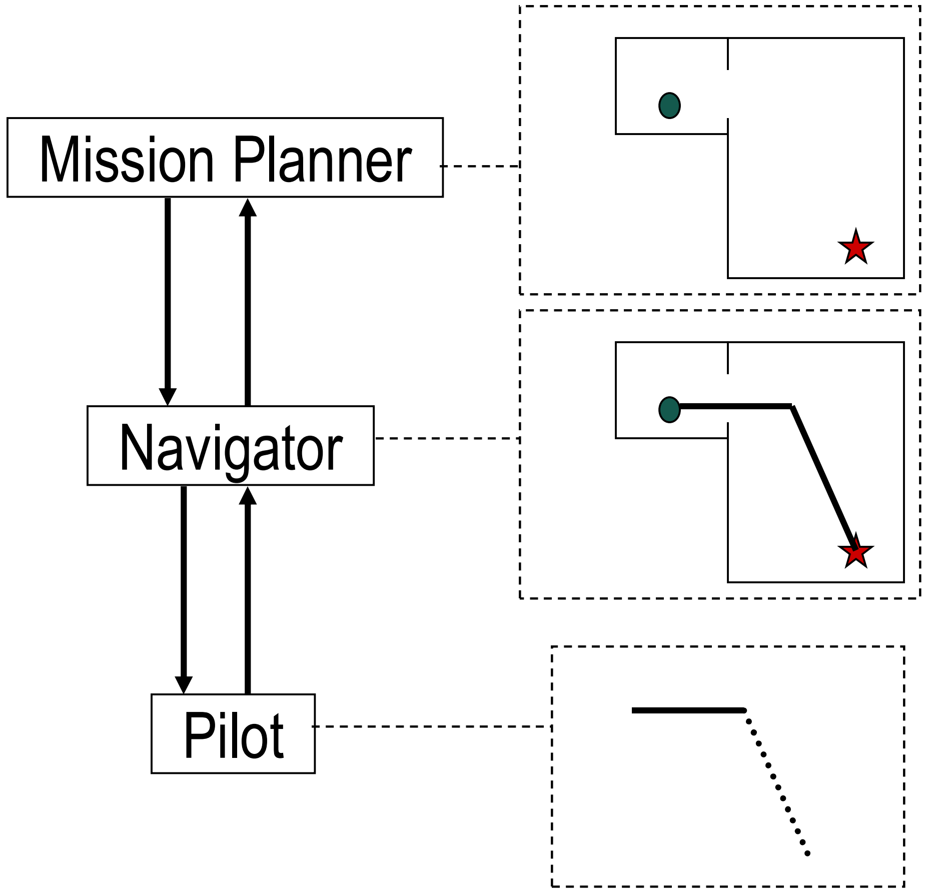
Major contribution of NHC: Decomposition of planning into three subsystems

Planning is Hierarchical

Uses map to locate self and goal

Generates path from current position to goal

Generates actions robot must execute to follow path segment



Advantage/Disadvantage of NHC

- Advantage:
  - Interleaves planning and acting (unlike STRIPS)
    - Plan is changed if world is different from expected

- Disadvantage:
  - Planning decomposition is only appropriate for navigation tasks