<table>
<thead>
<tr>
<th>Opn</th>
<th>Process</th>
<th>VMC1</th>
<th>VMC1</th>
<th>LN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>A</td>
<td></td>
<td></td>
<td>0.00</td>
<td>Install 0.30-diameter drill bit</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td>0.10</td>
<td>Install 0.20-diameter drill bit</td>
</tr>
<tr>
<td>004</td>
<td>A</td>
<td></td>
<td></td>
<td>0.00</td>
<td>Install 0.15-diameter side-milling tool</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td>0.10</td>
<td>Install 0.08-diameter end-milling tool</td>
</tr>
<tr>
<td>005</td>
<td>D</td>
<td></td>
<td></td>
<td>30.00</td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td>54.77</td>
<td>Total time on EC1</td>
</tr>
<tr>
<td>006</td>
<td>A</td>
<td></td>
<td></td>
<td>30.00</td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td>30.00</td>
<td>Setup</td>
</tr>
<tr>
<td>006</td>
<td>C</td>
<td></td>
<td></td>
<td>30.00</td>
<td>Setup</td>
</tr>
</tbody>
</table>

May All Your Plans Succeed! (or have a high expected utility)

Dana S. Nau

---

© 2006

May All Your Plans Succeed!
(or have a high expected utility)

Dana S. Nau
plan |plan|
noun
1 a detailed proposal for doing or achieving something: the UN peace plan.
   • [with adj. ] a scheme for the regular payment of contributions toward a pension,
     savings account, or insurance policy: a personal pension plan.
2 (usu. plans) an intention or decision about what one is going to do: I have no plans
to retire.
3 a detailed diagram, drawing, or program, in particular
   • a fairly large-scale map of a town or district: a street plan.
   • a drawing or diagram made by projection on a horizontal plane, esp. one
     showing the layout of a building or one floor of a building. Compare with
     ELEVATION (sense 3 ).
   • a diagram showing how something will be arranged: look at the seating plan.

verb (planned, planning) [trans. ]
1 decide on and arrange in advance: they were planning a trip to Egypt | [with infinitive
   ] he plans to fly on Wednesday | [ intrans. ] we plan on getting married in the near future.
   See note at INTEND .
   • [ intrans. ] make preparations for an anticipated event or time: we have to plan
     for the future.
2 design or make a plan of (something to be made or built): they were planning a garden.
### Plan

**nouns**

1. a detailed proposal for doing or achieving something: the UN peace plan.

---

**Processes:**

<table>
<thead>
<tr>
<th>Opn</th>
<th>BC/WW</th>
<th>Setup</th>
<th>Runtime</th>
<th>LN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td>VMC1</td>
<td></td>
<td>2.00</td>
<td></td>
<td>Orient board</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Clamp board</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Establish datum point at bullseye (0.25, 1.00)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Install 0.30-diameter drill bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rough drill at (1.25, -0.50) to depth 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Finish drill at (1.25, -0.50) to depth 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Install 0.20-diameter drill bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rough drill at (0.00, 4.88) to depth 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Finish drill at (0.00, 4.88) to depth 1.00</td>
</tr>
<tr>
<td>001 B</td>
<td>VMC1</td>
<td></td>
<td>0.10</td>
<td></td>
<td>Install 0.15-diameter side-milling tool</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rough side-mill pocket at (-0.25, 1.25) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Finish side-mill pocket at (-0.25, 1.25) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rough side-mill pocket at (-0.25, 3.00) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Finish side-mill pocket at (-0.25, 3.00) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td>001 C</td>
<td>VMC1</td>
<td></td>
<td>0.10</td>
<td></td>
<td>Install 0.08-diameter end-milling tool</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre-clean board (scrub and wash)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dry board in oven at 85 deg. F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>001 T</td>
<td>VMC1</td>
<td></td>
<td>2.20</td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>004 A</td>
<td>VMC1</td>
<td></td>
<td>0.00</td>
<td></td>
<td>Orient board</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Clamp board</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Establish datum point at bullseye (0.25, 1.00)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Install 0.15-diameter side-milling tool</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rough side-mill pocket at (-0.25, 1.25) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Finish side-mill pocket at (-0.25, 1.25) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rough side-mill pocket at (-0.25, 3.00) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Finish side-mill pocket at (-0.25, 3.00) length 0.40, width 0.30, depth 0.50</td>
</tr>
<tr>
<td>004 B</td>
<td>VMC1</td>
<td></td>
<td>0.10</td>
<td></td>
<td>Install 0.08-diameter end-milling tool</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre-clean board (scrub and wash)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dry board in oven at 85 deg. F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>004 C</td>
<td>VMC1</td>
<td></td>
<td>0.10</td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>005 A</td>
<td>EC1</td>
<td>30.00</td>
<td>0.48</td>
<td></td>
<td>Pre-clean board (scrub and wash)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dry board in oven at 85 deg. F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>005 B</td>
<td>EC1</td>
<td>30.00</td>
<td>2.00</td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>005 C</td>
<td>EC1</td>
<td>30.00</td>
<td>2.00</td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>005 D</td>
<td>EC1</td>
<td>30.00</td>
<td>20.00</td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
<tr>
<td>005 T</td>
<td>EC1</td>
<td>90.00</td>
<td>54.77</td>
<td></td>
<td>Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Spread photoresist from 18000 RPM spinner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Photolithography of photoresist using phototool in &quot;real.iges&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Etching of copper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total time on VMC1</td>
</tr>
</tbody>
</table>

---

Part of a manufacturing process plan

[A representation] of future behavior ... usually a set of actions, with temporal and other constraints on them, for execution by some agent or agents. - Austin Tate

[MIT Encyclopedia of the Cognitive Sciences, 1999]
Generating Plans of Action

- Computer programs to aid human planners
  - Project management (consumer software)
  - Plan storage and retrieval
    - e.g., *variant process planning* in manufacturing
  - Automatic schedule generation
    - various OR and AI techniques
  - For some problems, we would like generate plans (or pieces of plans) automatically
    - Much more difficult
    - Automated-planning research is starting to pay off
  - Here are some examples …
Space Exploration

- Mars rovers
  - Autonomous planning, scheduling, control
  - NASA (JPL and Ames)
Manufacturing

- Sheet-metal bending machines
  - Amada Corporation
  - Software to plan the sequence of bends
Games

- **Bridge Baron - Great Game Products**
  - Won 1997 world championship of computer bridge by using HTN planning to generate game trees [Smith et al.: AAAI 1998, AI Magazine 1998]
- Current version of **Bridge Baron** is still one of the best bridge programs

```
Finesse(P₁; S)

LeadLow(P₁; S)

PlayCard(P₁; S, R₁)

EasyFinesse(P₂; S)

StandardFinesse(P₂; S)

FinesseTwo(P₂; S)

BustedFinesse(P₂; S)

StandardFinesseTwo(P₂; S)

StandardFinesseThree(P₃; S)

FinesseFour(P₄; S)

PlayCard(P₂; S, R₂)

PlayCard(P₃; S, R₃)

PlayCard(P₄; S, R₄)

PlayCard(P₄; S, R₄’)
```

Us: East declarer, West dummy
Opponents: defenders, South & North
Contract: East – 3NT
On lead: West at trick 3

East: ♠KJ74
West: ♠A2
Out: ♠QT98653

North—♠3
East—♠J
South—♠5
South—♠Q
Outline

» Conceptual model for planning
» Example planning algorithms
» What’s bad, what’s good
» Directions and trends

• This talk is deliberately non-technical
• For technical details:
  » Ghallab, Nau, and Traverso
    Automated Planning: Theory and Practice
    Morgan Kaufmann, May 2004
  » First comprehensive
textbook & reference book
  on automated planning
  » http://www.laas.fr/planning
Conceptual Model

1. Environment

- Initial state(s)
- Objectives
- Execution status
- Observations
- Actions
- Events

System $\Sigma$

Description of $\Sigma$

Planner

Controller

State transition system

...
State Transition System

\[ \Sigma = (S, A, E, \gamma) \]

- **S** = \{states\}
- **A** = \{actions\}
- **E** = \{exogenous events\}
- **\gamma** = state-transition function

**Example:**
- \( S = \{s_0, \ldots, s_5\} \)
- \( A = \{\text{put, take, load, \ldots}\} \)
- \( E = \emptyset \)
- \( \gamma: \) see the arrows
Observation function

\( h: S \rightarrow O \)

Conceptual Model

2. Controller

Initial state(s) → Description of \( \Sigma \)

Objectives

Execution status

Plan(s)

Controller

Observations → Actions

System \( \Sigma \)

Events

Given observation \( o \) in \( O \), produces action \( a \) in \( A \)
Conceptual Model
3. Planner’s Input

Planning problem

Omit unless planning is online
Planning Problem

- Description of $\Sigma$
- Initial state or set of states
  - Initial state $= s_0$
- Objective
  - Goal state, set of goal states, set of tasks, "trajectory" of states, objective function, …
  - Goal state $= s_5$
Conceptual Model
4. Planner’s Output

Initial state(s)
Objectives
Execution status
Planner

Description of $\Sigma$

Plans
Controller

Observations
Actions
System $\Sigma$

Events

Instructions to the controller
Plans

- Classical plan: a sequence of actions

\[ \langle \text{take, move1, load, move2} \rangle \]

- Policy: partial function from \( S \) into \( A \)

\[ \{(s_0, \text{take}), (s_1, \text{move1}), (s_3, \text{load}), (s_4, \text{move2})\} \]
Planning Versus Scheduling

- **Scheduling**
  - When and how to perform a given set of actions
    - Time constraints
    - Resource constraints
    - Objective functions
  - Typically NP-complete

- **Planning**
  - Decide what actions to use to achieve some set of objectives
  - Can be much worse than NP-complete; worst case is undecidable
Three Main Types of Planners

1. Domain-specific
2. Domain-independent
3. Configurable

I’ll briefly discuss each
Types of Planners: 1. Domain-Specific

- Made or tuned for a specific domain
- Won’t work well (if at all) in any other domain
- Most successful real-world planning systems work this way
Types of Planners: 2. Domain-Independent

- In principle:
  - Works in any planning domain
  - Only domain-specific knowledge is the definitions of the basic actions

- In practice:
  - Not feasible to develop domain-independent planners that work in every possible domain
    - Could you use a bridge program to explore Mars?
  - Restrictive assumptions to simplify the set of domains
    - Classical planning
    - Historical focus of most research on automated planning
Restrictive Assumptions

- **A0: Finite system:**
  - finitely many states, actions, events

- **A1: Fully observable:**
  - controller always knows Σ’s current state

- **A2: Deterministic:**
  - One initial state, one outcome for each action

- **A3: Static** (no exogenous events):
  - no changes but the controller’s actions

- **A4: Attainment goals:**
  - a set of goal states $S_g$

- **A5: Sequential plans:**
  - linear sequence of actions $(a_1, a_2, ... a_n)$

- **A6: Implicit time:**
  - no time durations; linear sequence of instantaneous states

- **A7: Off-line planning:**
  - planner doesn’t know the execution status
Classical Planning

- Classical planning requires all eight restrictive assumptions
  - Offline generation of action sequences for a deterministic, static, finite system, with complete knowledge, attainment goals, and implicit time
- Reduces to the following problem:
  - Given \((\Sigma, s_0, S_g)\)
  - Find a sequence of actions \(\langle a_1, a_2, \ldots, a_n \rangle\) that produces a sequence of state transitions \(\langle s_1, s_2, \ldots, s_n \rangle\) such that \(s_n\) is in \(S_g\).
- This is just path-searching in a graph
  - Nodes = states
  - Edges = actions
- Is this trivial?
Classical Planning

- Generalize the earlier example:
  - Five locations, three robot carts, 100 containers, three piles
    - Then there are $10^{277}$ states
  - Number of particles in the universe is only about $10^{87}$
    - The example is more than $10^{190}$ times as large!

- Automated-planning research has been heavily dominated by classical planning
  - Dozens (hundreds?) of different algorithms
  - I’ll briefly mention a few of the best-known ones
Partial-Order Planning

- Decompose sets of goals into the individual goals
- Plan for them separately
  » Bookkeeping info to detect and resolve interactions

- In classical planning, not used much any more
- The Mars rovers use temporal-planning extensions of it
Graphplan

- Planning graph: problem relaxation
  - Apply all applicable actions simultaneously
  - Next “level” = \{effects of all of those actions\}
- Restrict the planner to search within the planning graph
- Graphplan’s many children
  - IPP, CGP, DGP, LGP, PGP, SGP, TGP, …

Running out of names
Heuristic Search

- Do an A*-style heuristic search guided by a *heuristic function* that estimates the distance to a goal
  - Can use problem relaxations to compute the heuristic function

- Problem: A* quickly runs out of memory
  - So do a greedy search

- Greedy search can get trapped in local minima
  - Greedy search plus local search at local minima

- HSP, HSP2 [Bonet & Geffner]
- FastForward [Hoffmann]
Translation to Other Domains

- Translate the planning problem or the planning graph into another kind of problem for which there are efficient solvers
  - Find a solution to that problem
  - Translate the solution back into a plan

- Satisfiability solvers, especially those that use local search
  - Blackbox, Satplan [Kautz & Selman]

- Integer programming solvers such as Cplex
  - [Vossen et al.]
Types of Planners: 3. Configurable

- Domain-independent planners are quite slow compared with domain-specific planners
  - Blocks world in linear time [Slaney and Thiébaux, *A.I.*, 2001]
  - Can get analogous results in many other domains
- But we don’t want to write a whole new planner for every domain!
- Configurable planners
  - Domain-independent planning engine
  - Input includes info about how to solve problems in the domain
    - Hierarchical Task Network (HTN) planning
    - Planning with control formulas
**HTN Planning**

- Problem reduction
  - *Tasks* (activities) rather than goals
  - *Methods* to decompose tasks into subtasks
  - Enforce constraints, backtrack if necessary
- Real-world applications
- Noah, Nonlin, O-Plan, SIPE, SIPE-2, SHOP, SHOP2

**Method:** taxi-travel\((x,y)\)

- get-taxi
- ride\((x,y)\)
- pay-driver

**Method:** air-travel\((x,y)\)

- get-ticket\((a(x),a(y))\)
- fly\((a(x),a(y))\)
- travel\((a(y),y)\)

**Task:** travel\((x,y)\)
Example

- SHOP2
  - My group’s HTN planning system
  - Won one of the top four awards in the 2002 International Planning Competition
  - Freeware, open source
    - Several thousand downloads - I stopped keeping track
  - Used in hundreds of projects worldwide
    [IEEE Intelligent Systems, 2005]
Planning with Control Formulas

- Forward search
- At each state $s_i$, we have a control formula $f_i$ in temporal logic
  
  \[
  ontable(x) \land \neg \exists y: \text{GOAL(on}(x, y)) \Rightarrow \Box(\neg \text{holding}(x))
  \]

  “never pick up $x$ from table unless $x$ needs to be on another block”

- For each successor of $s$, derive a control formula using logical progression
- Prune any successor state in which the progressed formula is false
  - TLPlan [Bacchus & Kabanza]
  - TALplanner [Kvarnstrom & Doherty]
Comparisons

- **Domain-specific planner**
  - Write an entire computer program - lots of work
  - Lots of domain-specific performance improvements
- **Domain-independent planner**
  - Just give it the basic actions - not much effort
  - Not very efficient
A domain-specific planner only works in one domain

In principle, configurable and domain-independent planners should both be able to work in any domain

In practice, configurable planners work in a larger variety of domains
  » Partly due to efficiency
  » Partly due to expressive power
Example

- The planning competitions
  - All of them included domain-independent planners
- In addition, AIPS 2000 and IPC 2002 included configurable planners
- The configurable planners
  - Solved the most problems
  - Solved them the fastest
  - Usually found better solutions
  - Worked in many non-classical planning domains that were beyond the scope of the domain-independent planners
But Wait …

- *IPC* 2004 and *IPC* 2006 included *no* configurable planners.
  - Why not?
But Wait …

- *IPC 2004* and *IPC 2006* included *no* configurable planners.
  - Why not?
- Hard to enter them in the competition
  - Must write all the domain knowledge yourself
  - Too much trouble except to make a point
  - The authors of TLPlan, TALplanner, and SHOP2 felt they had already made their point
But Wait …

- *IPC 2004* and *IPC 2006* included no configurable planners.
  - Why not?
- Hard to enter them in the competition
  - Must write all the domain knowledge yourself
  - Too much trouble except to make a point
  - The authors of TLPlan, TALplanner, and SHOP2 felt they had already made their point
- Why not provide the domain knowledge?
But Wait …

- *IPC* 2004 and *IPC* 2006 included *no* configurable planners.
  - Why not?
- Hard to enter them in the competition
  - Must write all the domain knowledge yourself
  - Too much trouble except to make a point
  - The authors of TLPlan, TALplanner, and SHOP2 felt they had already made their point
- Why not provide the domain knowledge?
  - Drew McDermott proposed this at *ICAPS-05*
  - Many people didn’t like this idea
    - Cultural bias against it
Cultural Bias

- Most automated-planning researchers feel that using domain knowledge is “cheating”
- Researchers in other fields have trouble comprehending this
  - Operations research, control theory, engineering, …
  - Why would anyone not want to use the knowledge they have about a problem they’re trying to solve?
- In the past, the bias has been very useful
  - Without it, automated planning wouldn’t have grown into a separate field from its potential application areas
- But it’s not useful any more
  - The field has matured
  - The bias is too restrictive
Example

- Typical characteristics of application domains
  - Dynamic world
  - Multiple agents
  - Imperfect/uncertain info
  - External info sources
    - users, sensors, databases
  - Durations, time constraints, asynchronous actions
  - Numeric computations
    - geometry, probability, etc.
- Classical planning excludes all of these

Nau: Plans, 2006
In Other Words …

- We *like* to think classical planning is domain-independent planning
- **But it isn’t!**
  - Classical planning only includes domains that satisfy some *very* specific restrictions
  - Classical planners depend heavily on those restrictions
In Other Words ...

- We *like* to think classical planning is domain-independent planning
- **But it isn’t!**
  - Classical planning only includes domains that satisfy some *very* specific restrictions
  - Classical planners depend heavily on those restrictions
- This is fine for “toy problems” like the *blocks world*
In Other Words …

- We *like* to think classical planning is domain-independent planning
- **But it isn’t!**
  - Classical planning only includes domains that satisfy some **very** specific restrictions
  - Classical planners depend heavily on those restrictions
- This is fine for “toy problems” like the **blocks world**
- *Not* so fine for the **real world**
Good News, Part 1

- We’re already moving away from classical planning
- Example: the planning competitions
- Increasing divergence from classical planning
  - 1998, 2000: classical planning
  - 2002: added elementary notions of time durations, resources
  - 2004: added inference rules, derived effects, and a separate track for planning under uncertainty
  - 2006: added soft goals, trajectory constraints, preferences, plan metrics
Good News, Part 2

- Success in high-profile applications
  - A success like the Mars rovers is a big deal
  - Creates excitement about building planners that work in the real world
Good News, Part 3

- These successes provide opportunities for synergy between theory and practice
  - Understanding real-world planning leads to better theories
  - Better theories lead to better real-world planners
Good News, Part 4

- Classical planning research has produced some very powerful techniques for reducing the size of the search space
- We can generalize these techniques to work in non-classical domains
- Examples:
  1. Partial order planning
     - Extended to do temporal planning
       - Mars rovers
  2. HTN planning
     - Lots of applications
  3. Planning under uncertainty …
Digression: What planning under uncertainty is

- Actions with several possible outcomes
  - Action failures, e.g., *gripper drops its load*
  - Exogenous events, e.g., *road closed*
- Primary models
  - Markov Decision Processes (MDPs)
    - Probabilities, costs, rewards, optimize expected utility
    - Dynamic programming
  - Nondeterministic planning domains
    - No numbers
    - Solutions: weak, strong, strong-cyclic, …
    - Symbolic model checking
  - Game-theoretic
    - game-tree search (e.g., minimax)
3. General way to \textit{nondeterminize} forward-chaining planners

» Rewrite them to work in nondeterministic domains
  • TLPlan $\rightarrow$ ND-TLPlan
  • TALplanner $\rightarrow$ ND-TALplanner
  • SHOP2 $\rightarrow$ ND-SHOP2

» Big (exponential) speedups compared to previous planners for nondeterministic domains [Kuter and Nau, \textit{AAAI-04}]

» Even bigger speedups if we use the BDD representation used in the previous planners for nondeterministic domains
  • [Kuter, Nau, Pistore, and Traverso, \textit{ICAPS-05}]
  • Analogous results for MDPs [Kuter and Nau, \textit{AAAI-05}]
  • Possible extension to game-theoretic environments?
Important Trends, and Directions for Growth
Planning in Multi-Agent Environments

- Traditional assumption: the planner is alone in the world
- In reality:
  - The planner is part of a larger system
  - Other agents: human or automated or both
- The planner needs to
  - Recognize what those agents are trying to accomplish
  - Generate an appropriate response
- Examples
  - Mixed-initiative and embedded planning
  - Assisted cognition
  - Customer service hotlines
  - Surveillance applications
  - Games
Temporal Planning

- Classical planning uses a trivial model of time
  - Linear sequence of instantaneous states $s_0, s_1, s_2, \ldots$
  - Several “temporal” logics do the same thing
- Need
  - Time durations, overlapping actions
  - Integrated planning/scheduling (e.g., space exploration)
  - Continuous change (e.g., vehicle movement)
  - Temporally extended goals - “trajectories” of states
- Growth is already occurring
  - E.g., the planning competitions
- Still more to be done
Dynamic External Information

- Traditional assumption
  - Information is static; planner starts with all of it
- Real-world planning
  - Acquire information during planning and execution
    - Applications: web services, many others
  - What info to look for? Where to get it?
  - How to deal with lag time and information volatility?
  - What if the query itself causes change in the world?
- Candidate for a new IPC track?
Acquiring Domain Knowledge

- How to get the domain knowledge needed to plan efficiently?
  - One of the most neglected topics for planning research, but one of the most important
  - If we could do this well on real-world problems, planners would be hundreds of times more useful
- Researchers are starting to realize this
  - E.g., the “Knowledge Engineering Competition” at ICAPS-05
- Overlap with HCI, ML, and CBR
Real Plans in Real Domains

- Data mining has become an important field very quickly
  » One reason: researchers can easily get real-world data
  » E.g., go to the web
- One reason automated-planning researchers have concentrated on “toy” problems:
  » Trouble getting access to real plans for real problems
  » Need a source of real-world planning data
- Half-baked idea: could we data-mine plans and domains from web sources?
  » My lab is starting to look for ways to do this
Various kinds of planning are studied in many different fields

- AI planning, computer games, game theory, OR, economics, psychology, sociology, political science, industrial engineering, systems science, control theory

The research groups are often nearly disjoint

- Different terminology, assumptions, ideas of what’s important
- Hard to tell what the similarities and differences are

Potential for cross-pollination

- Combine ideas and approaches from different fields

Example: applications to social and behavioral sciences
Cross-disciplinary research laboratory at the University of Maryland

- Faculty from CS, Business, EE, Government & Politics, International Development, Conflict Management

Very ambitious goals

- Develop theory and algorithms needed for tools to support decision making in cultural contexts.
- Help understand how/why other cultures make decisions
  - More effective cross-cultural interactions
  - Better governance when different cultures are involved
  - Recovery from conflicts and disasters
  - Improve quality of life in developing countries

Example: research by Tsz-Chiu Au, a graduate student at UMD
Prisoner’s Dilemma

- One of the best-known examples of a non-zero-sum game
- Two players, each has two possible moves:
  - Cooperate (C) with the other player
  - Defect (D), i.e., take advantage of the other player
- Nash equilibrium strategy: (D, D)

Payoff matrix:

<table>
<thead>
<tr>
<th></th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player 1</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>3, 3</td>
</tr>
<tr>
<td>D</td>
<td>5, 0</td>
</tr>
</tbody>
</table>

My best move is “defect,” regardless of whether he cooperates or defects.
Iterated Prisoner’s Dilemma (IPD)

- Axelrod (1984), *The Evolution of Cooperation*
- Two players, finite number of iterations of the Prisoner’s Dilemma
- Widely used to study emergence of cooperative behavior among agents
  - No optimal strategy
  - Performance depends on the strategies of all of the players
- The best strategy in Axelrod’s tournaments:
  - *Tit-for-Tat (TFT)*
    - On 1st move, cooperate. On *n*th move, repeat the other player’s (*n*–1)-th move
    - Could establish and maintain cooperations with many other players
    - Could prevent malicious players from taking advantage of it

Payoff matrix:

<table>
<thead>
<tr>
<th></th>
<th>Player₁</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player₂</td>
<td>C</td>
<td>3, 3</td>
<td>0, 5</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>5, 0</td>
<td>1, 1</td>
</tr>
</tbody>
</table>

If I defect now, he might punish me by defecting next time
IPD with Noise

- Models accidents and misinterpretations
- There’s a nonzero probability (e.g., 10%) that a “noise gremlin” will change some of the actions
  - *Cooperate* (C) will become *Defect* (D), and vice versa
- Tit-for-Tat and other strategies fail to maintain cooperation
- Tsz-Chiu Au’s **DBS strategy:**
  - Build a model of the other player’s strategy by observing his/her behavior
  - Use this model to detect noise
  - Use it to plan DBS’s actions
  - Detect when the other player’s strategy changes
    - Update the model
The 20th-Anniversary Iterated Prisoner’s Dilemma Competition

http://www.prisoners-dilemma.com

- Category 2: IPD with noise
  - 165 programs participated

- DBS dominated the top 10 places

- Only two programs beat DBS
  - Both used a strategy that was dangerously close to cheating

<table>
<thead>
<tr>
<th>Rank</th>
<th>Program</th>
<th>Avg. score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BWIN</td>
<td>433.8</td>
</tr>
<tr>
<td>2</td>
<td>IMM01</td>
<td>414.1</td>
</tr>
<tr>
<td>3</td>
<td>DBSz</td>
<td>408.0</td>
</tr>
<tr>
<td>4</td>
<td>DBSy</td>
<td>408.0</td>
</tr>
<tr>
<td>5</td>
<td>DBSpl</td>
<td>407.5</td>
</tr>
<tr>
<td>6</td>
<td>DBSx</td>
<td>406.6</td>
</tr>
<tr>
<td>7</td>
<td>DBSf</td>
<td>402.0</td>
</tr>
<tr>
<td>8</td>
<td>DBSft</td>
<td>401.8</td>
</tr>
<tr>
<td>9</td>
<td>DBSd</td>
<td>400.9</td>
</tr>
<tr>
<td>10</td>
<td>lowESTFT_classic</td>
<td>397.2</td>
</tr>
<tr>
<td>11</td>
<td>TFTIm</td>
<td>397.0</td>
</tr>
<tr>
<td>12</td>
<td>Mod</td>
<td>396.9</td>
</tr>
<tr>
<td>13</td>
<td>TFTlz</td>
<td>395.5</td>
</tr>
<tr>
<td>14</td>
<td>TFTlc</td>
<td>393.7</td>
</tr>
<tr>
<td>15</td>
<td>DBSe</td>
<td>393.7</td>
</tr>
<tr>
<td>16</td>
<td>TTFT</td>
<td>393.4</td>
</tr>
<tr>
<td>17</td>
<td>TFTIa</td>
<td>393.3</td>
</tr>
<tr>
<td>18</td>
<td>TFTIb</td>
<td>393.1</td>
</tr>
<tr>
<td>19</td>
<td>TFTIx</td>
<td>393.0</td>
</tr>
<tr>
<td>20</td>
<td>mediumESTFT_classic</td>
<td>392.9</td>
</tr>
</tbody>
</table>
How BWIN and IMM01 worked

- Each participant could submit up to 20 programs
- Some participants submitted 20 programs that worked as a team
  - 1 *master* + 19 *slaves*
    - When slaves play with master
      - they cooperate and master defects
      - master gets all the points
    - When slaves play with anyone not in their team, they defect
- Analysis
  - The average score of each master-and-slaves team was much lower than DBSz’s average score
  - If BWIN and IMM01 each had \( \leq 10 \) slaves, DBS would have placed 1st
  - If BWIN and IMM01 had no slaves, they would have done badly

My strategy? I order my goons to beat them up

I order my goons to give me all their money
DBS cooperates, not coerces

- Unlike BWIN and IMM01, DBS had *no* slaves
  - None of the DBS programs even knew the others were there
- DBS worked by establishing cooperation with *many* other agents
- DBS could do this *despite* the noise, because it could filter out the noise

- We’re trying this idea in other games
  - Joint work with Sarit Kraus
Conclusion

● Advances in automated planning
  » Historically, limited by focus on classical planning
  » Scope is broadening to include things important for real-world planning
  » Increased use in practical settings

● Important areas for future growth
  » multi-agent environments
    • reasoning about other agents
  » time durations
  » dynamic external information
  » acquiring domain knowledge
    • data mining of plans
  » cross-pollination with other fields
Any questions?