On Modeling Formalisms for Automated Planning

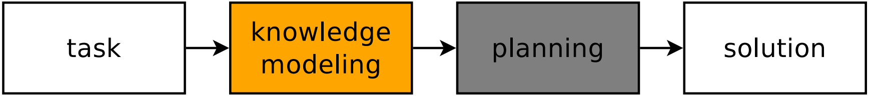
J. Vodrážka R. Barták

Faculty of Mathematics and Physics
Charles University in Prague

AIMSA, 2014

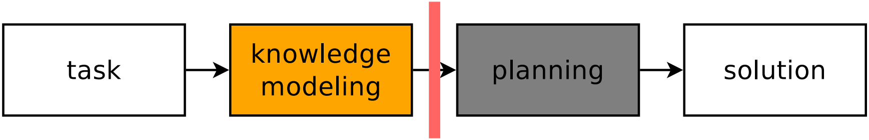
Automated planning

General model based approach to problem solving.



Automated planning

General model based approach to problem solving.



modeling formalism = interface

Real world example

Petrobras planning challenge

model: object classes, system state, operations

problem: initial state, desired state

solution: ordered set of actions

Academic approach

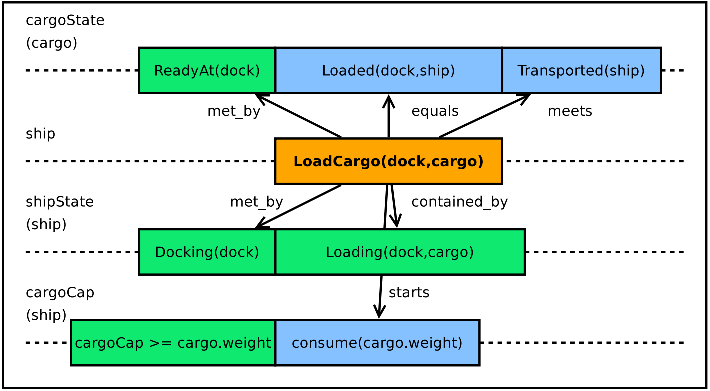
- PDDL (IPC)
- state = set of propositions (+ fluent values)
- changed by actions

PDDL operator sample

```
(:action load-cargo
  :parameters (?s - Ship ?c - Cargo
               ?loc - Location )
  :precondition (and
                 (at ?s ?loc)
                 (cargo-at ?c ?loc)
                 (isDocked ?s ?loc)
                 (>= (free-cargo-capacity ?s)
                     (cargo-weight ?c)))
  :effect (and
           (not (cargo-at ?c ?loc))
           (cargo-at ?c ?s)
           (decrease (free-cargo-capacity ?s)
                     (cargo-weight ?c)))
)
```

Engineering approach

- NDDL (Europa)
- timelines + intervals
- temporal constraints



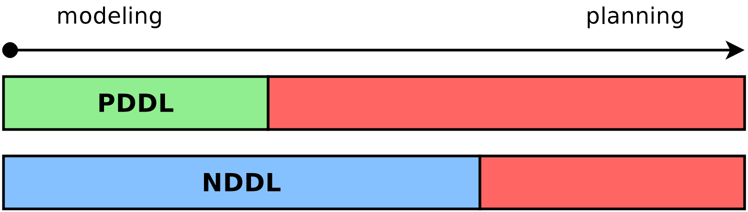
Comparison

PDDL

- predicate logic
- operators
- + clear model
- - sequencing

NDDL

- timelines
- compatibility rules
- - complex structures
- + expressivity



Alternative formalism

Key concepts:

1. State variables - representation of system state
2. Domain rules - domain specific knowledge
3. Operators - state transition

Alternative formalism

Key concepts:

1. State variables - representation of system state
2. Domain rules - domain specific knowledge
3. Operators - state transition

Alternative formalism

Key concepts:

1. State variables - representation of system state
2. Domain rules - domain specific knowledge
3. Operators - state transition

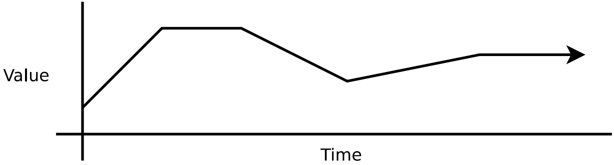
State variables

How to model a ship?

state variable declarations

```
shipLoc(Ship):{Location}
shipFuel(Ship):{Number}
shipAvailCap(Ship):{Number}
```

$$s(a_1, \ldots, a_k) : \{R\}$$



Domain rules

Domain specific knowledge

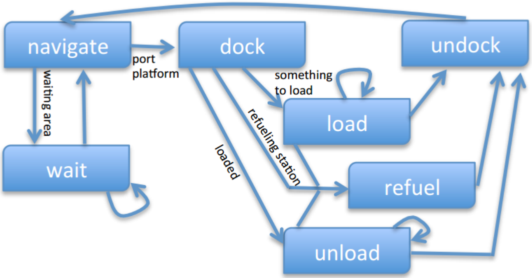
- numeric computation
- advanced check

fuelConsumption(L1,L2,Weight)

Domain rules

Domain specific knowledge

- numeric computation
- advanced check



Operators

Two types of expressions:

- conditional
- transitional

operator load cargo

```
loadCargo(C - Cargo; S - Ship; L - LogisticLoc;
          X - Number; A - ShipState)
     fsaCheck(load,A) = true
     cargoWeight(C) <= X
     shipLoc(S) = L
     shipStatus(S): load -> A
     shipAvailCap(S): X -> (X - cargoWeight(C))
     cargoLoc(C): L -> S
```

The end

Thank you.