tBurton: Model-based Temporal Generative Planning

David Wang

PI: Brian Williams

August 1st, 2012, Keck Institute for Space Studies

Model-based Embedded & Robotic Systems

CSAIL

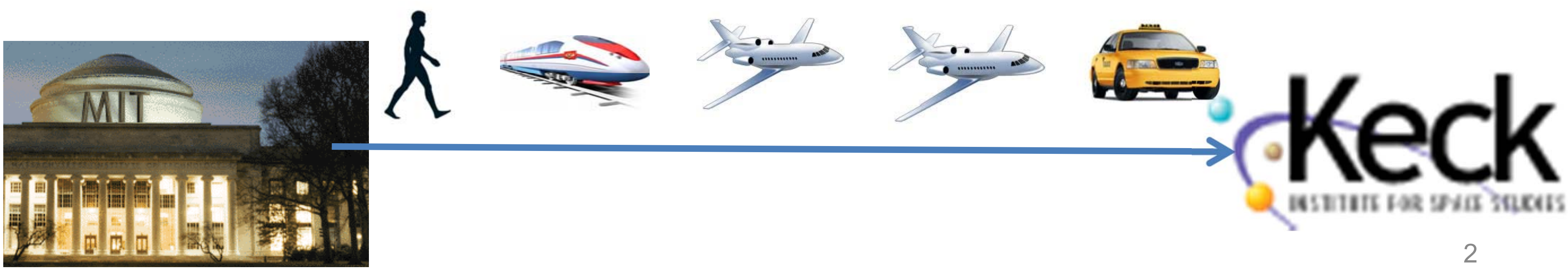
Generative Planning

initial state:

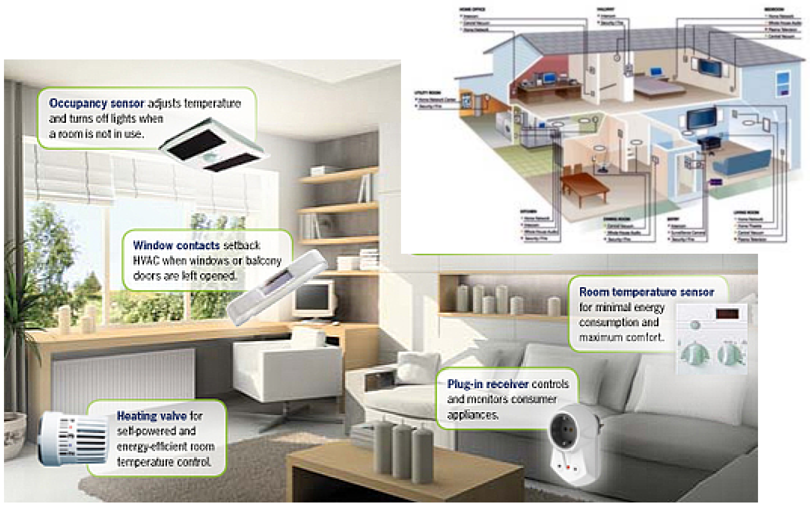
goals:

actions/model:

plan:

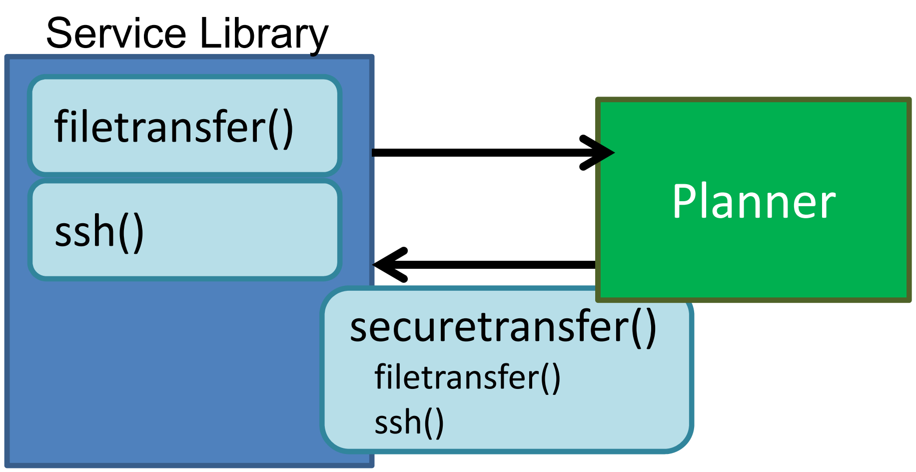


Applications



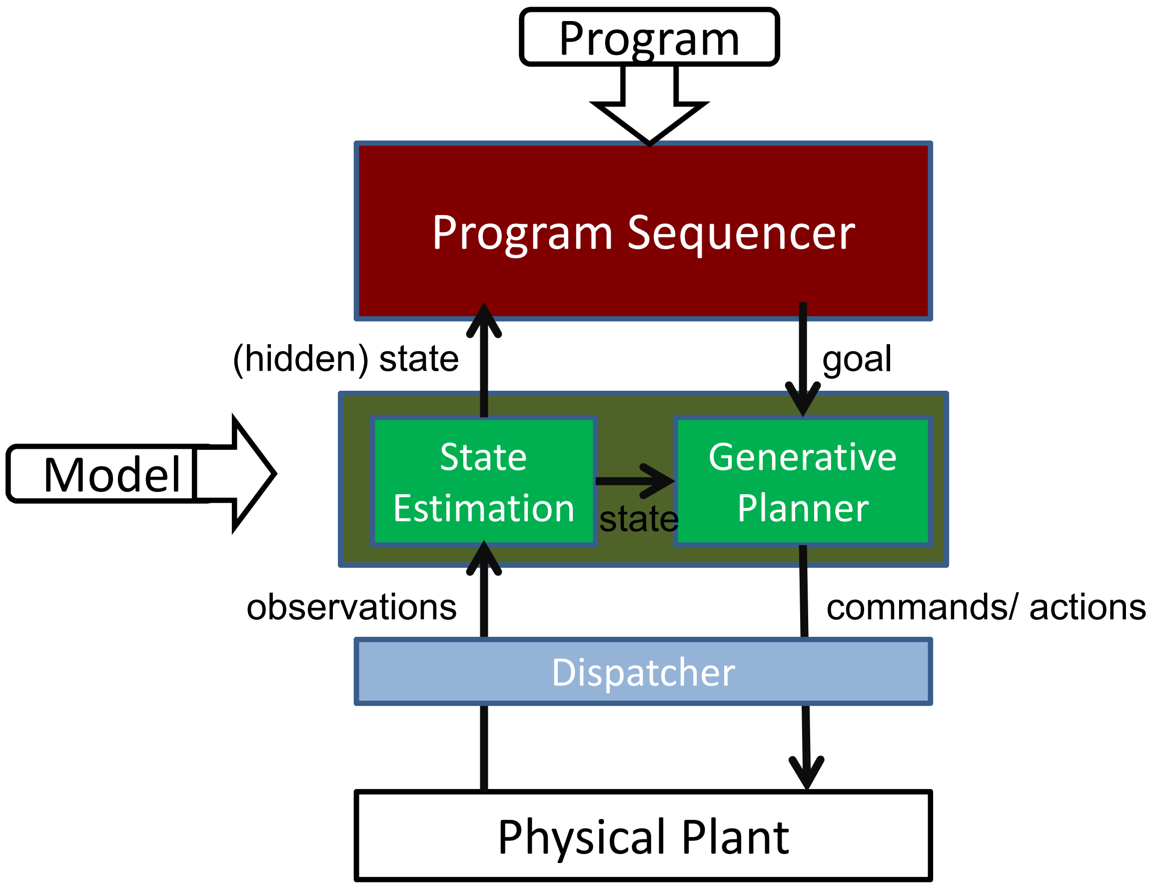
Intent Recognition

Fault Recovery



Service Composition

Model-based Executive



Program the nominal behavior, leave the exceptional behavior to the executive.

tBurton

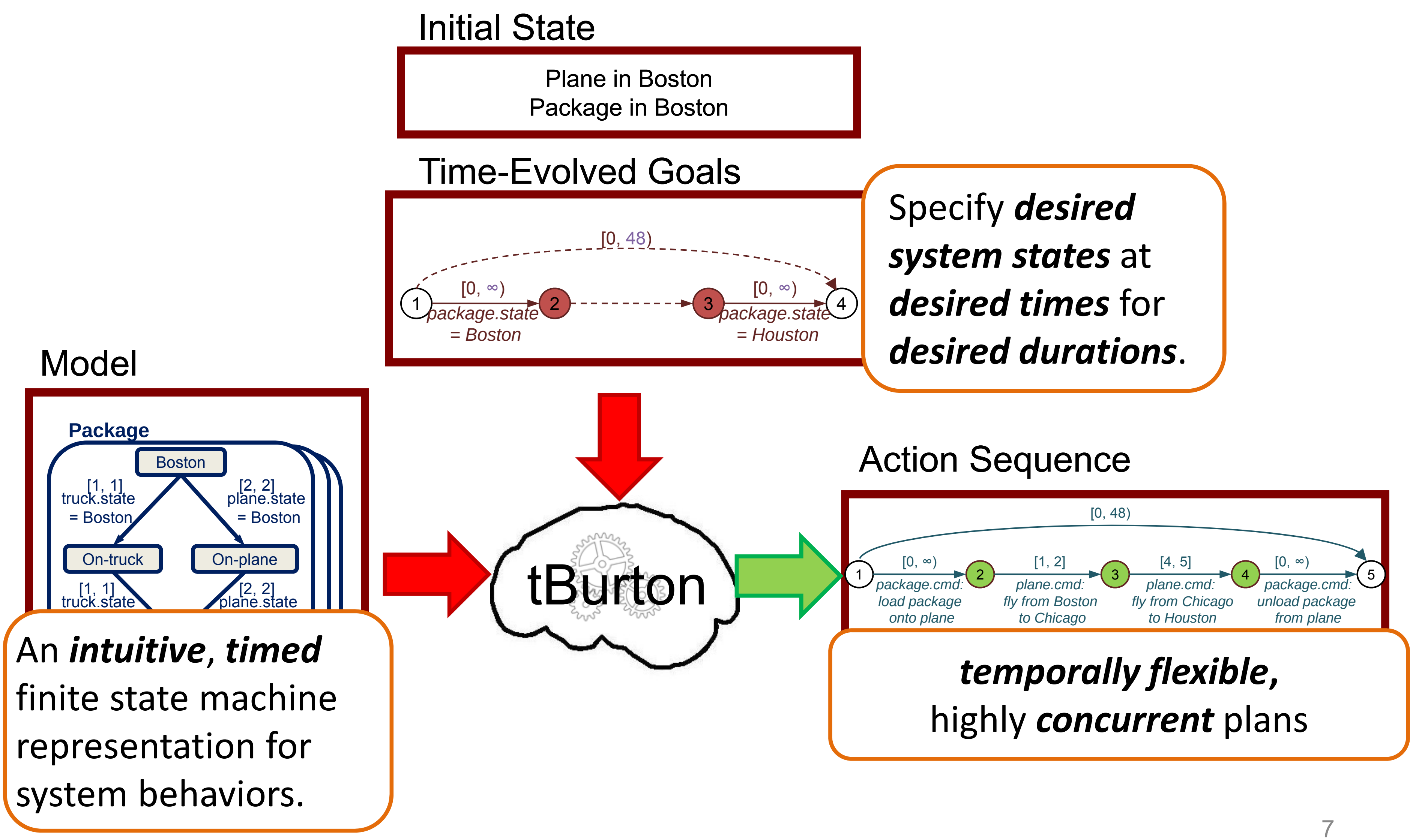
- Motivation
- **Overview**
- Example

Problem Traits

(Desired Expressivity)

- Time and concurrency are important
- Devices can transition automatically after timed intervals (timed light switch).
- Transitions in one device can cause transitions in another. (throwing a breaker switch, can turn off an appliance)
- Actions do not have to be reversible

tBurton


Initial State
Plane in Boston
Package in Boston
Time-Evolved Goals
[0, 48)
[0, ∞)
package.state = Boston
[0, ∞)
package.state = Houston
1
2
3
4
Specify *desired system states* at *desired times* for *desired durations*.
Model
Package
Boston
[1, 1] truck.state = Boston
[2, 2] plane.state = Boston
On-truck
On-plane
[1, 1] truck.state
[2, 2] plane.state
An *intuitive, timed* finite state machine representation for system behaviors.
tBurton
Action Sequence
[0, 48)
[0, ∞) package.cmd: load package onto plane
[1, 2] plane.cmd: fly from Boston to Chicago
[4, 5] plane.cmd: fly from Chicago to Houston
[0, ∞) package.cmd: unload package from plane
1
2
3
4
5
temporally flexible, highly *concurrent* plans

Planner Features

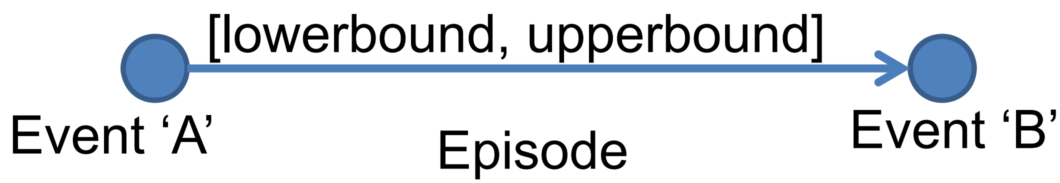
Time Robust Execution	Least-commitment planning through Temporal Networks
Natural Encoding	~~Use Finite State Machines (Timed Concurrent Constraint Automata)~~
Fast	Causal Graph Decomposition

Continuous Operation

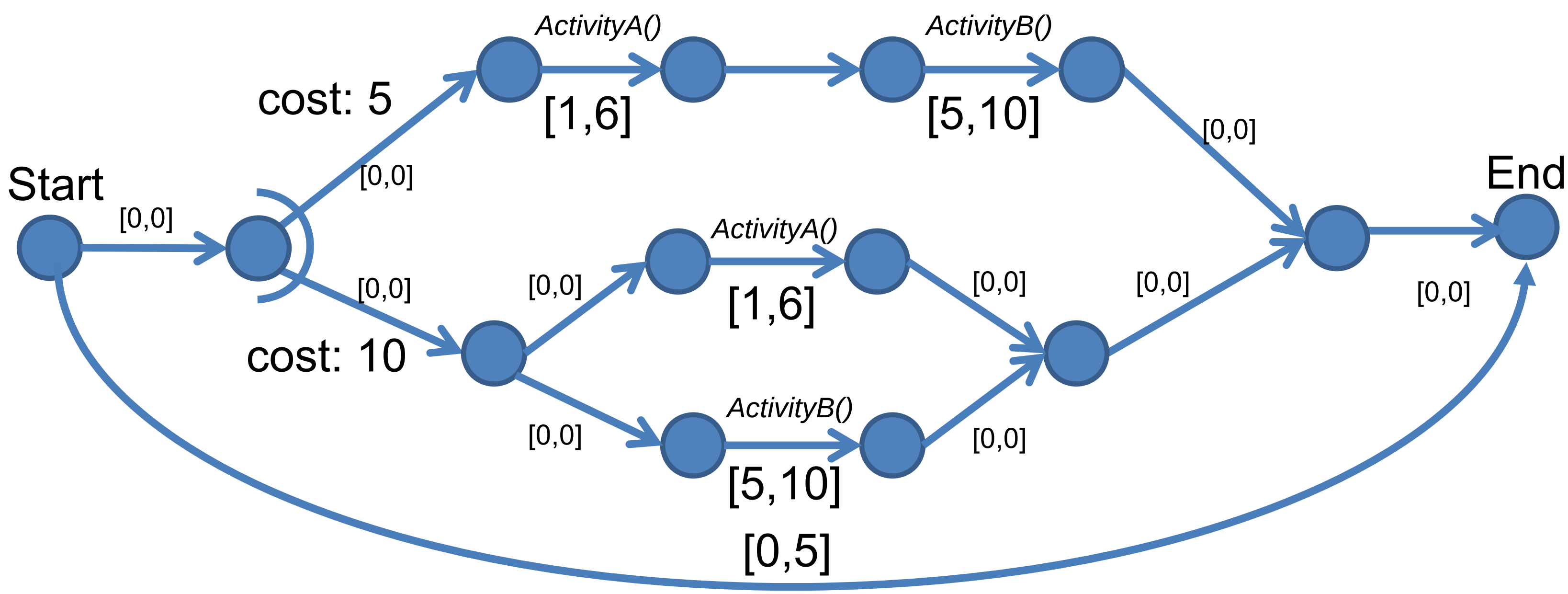
Risk-aware & Probabilistic Transitions

Parallel/Distributed

Temporal Plan Networks



Read: "Event B must occur between lowerbound and upperbound time after A"



Rina Dechter (1991), Muscettola, Paul Morris , Ioannis Tsamardinos (1998), Williams (2001), ...

Temporal Flexibility

(Temporal Least Commitment)

Resilient
Idea 2

Would you rather operate using the plan:

 12:00pm – walk to subway station

 12:15pm – take subway to airport

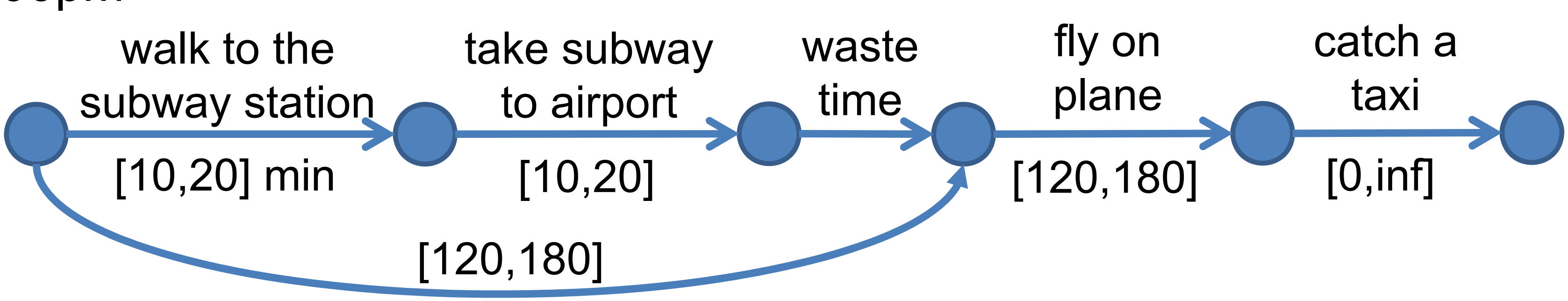
12:30pm – waste time at airport

 2:30pm – get on plane

 4:45pm – catch a taxi

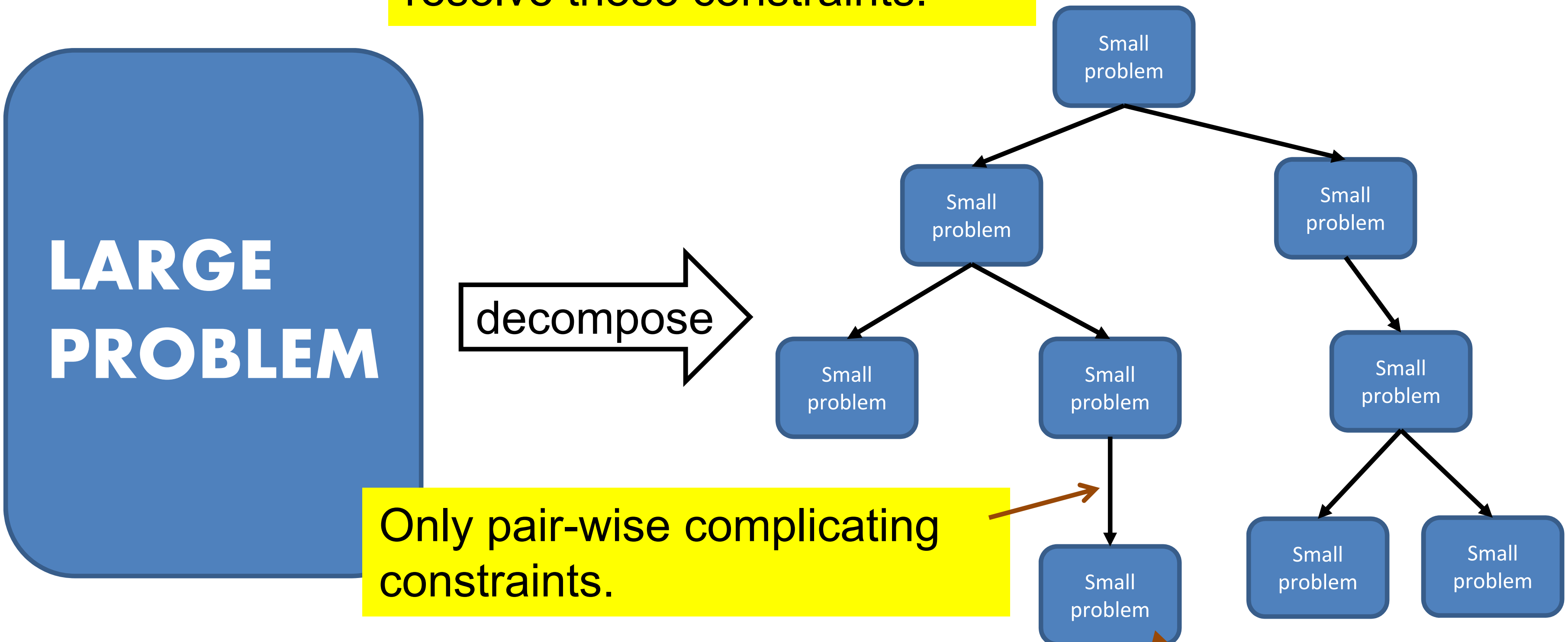
Or:

Starting at
12:00pm



Wouldn't it be great if…

There is an order in which to resolve those constraints.

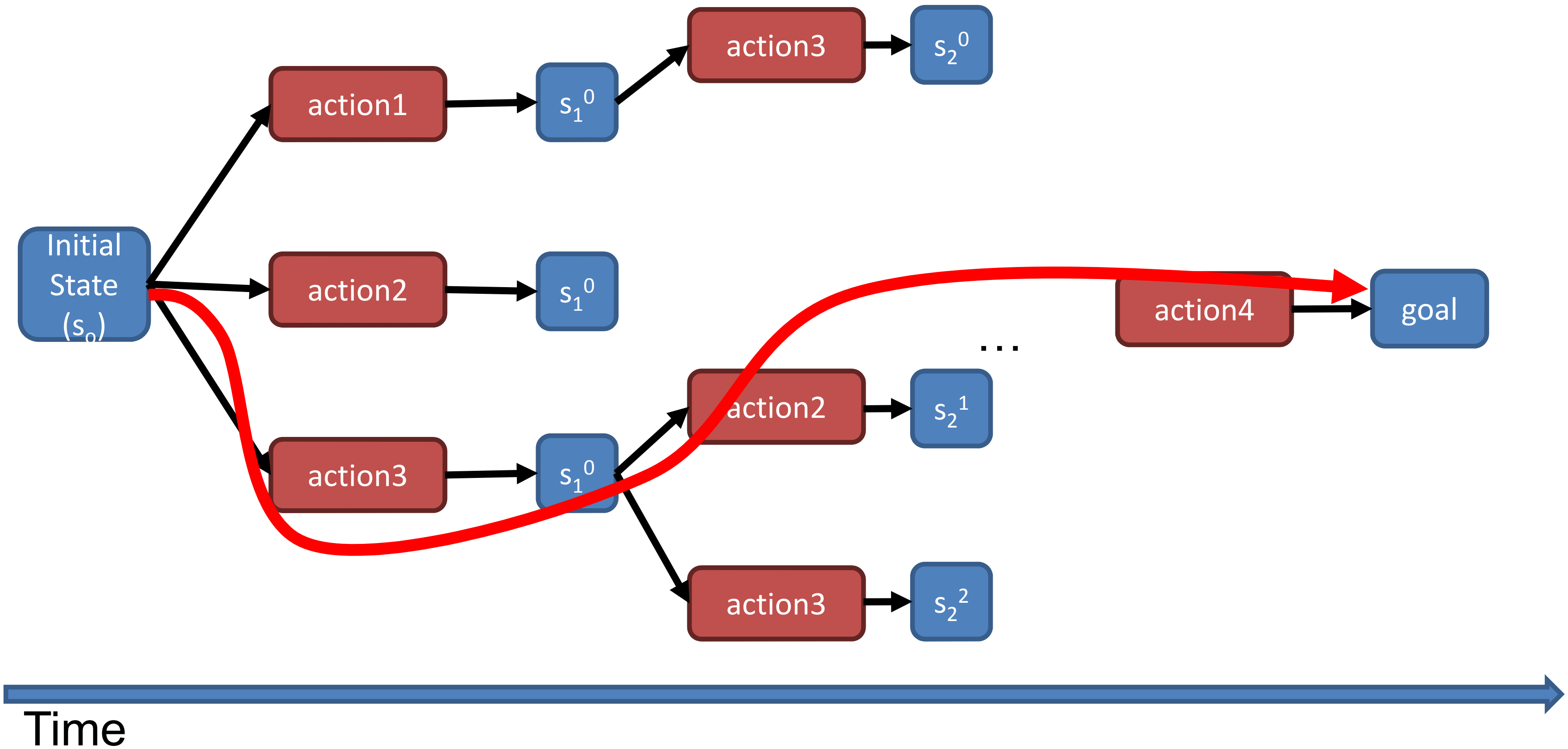


Only pair-wise complicating constraints.

only has sequential plans*

Forward-Search Planners

Very fast, but only generates sequential plans.

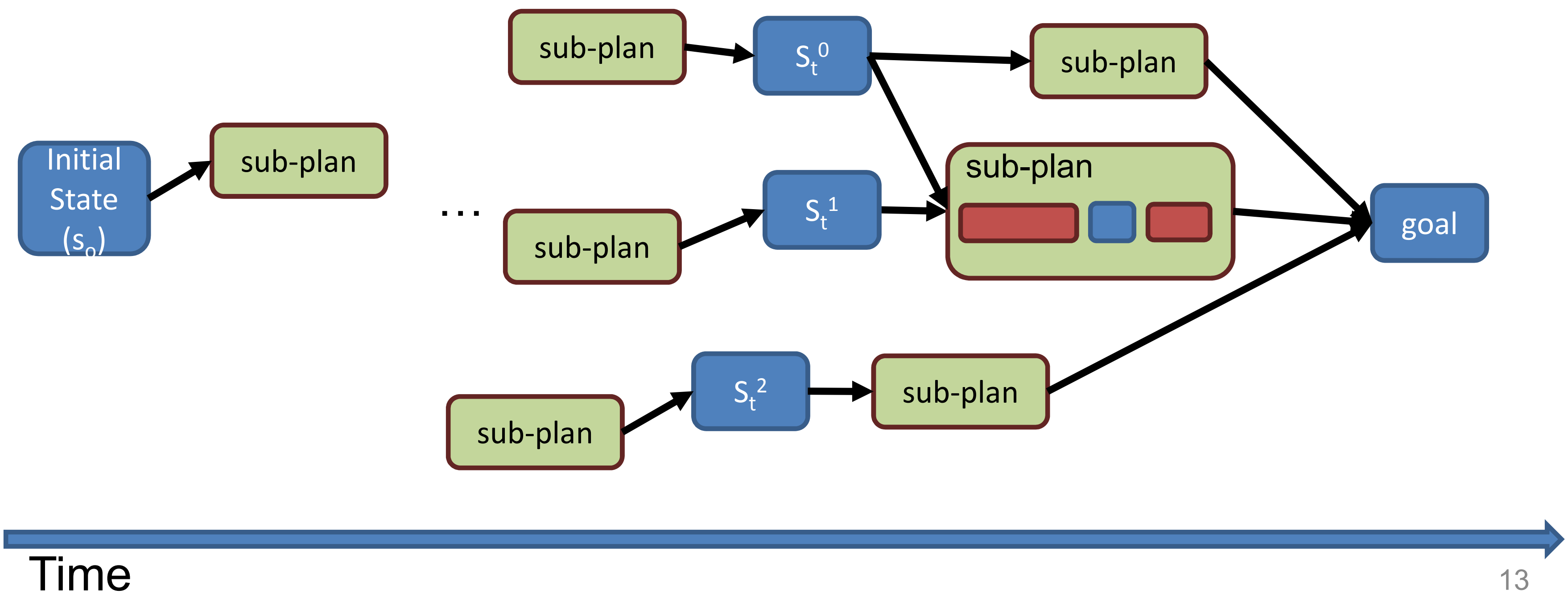


State of the art generative planners use heuristics to guide the search.

tBurton

Ideas:

- Concurrent Constraint Automata Encoding
- ***Decomposition based on Causal Graphs***
- Goal Regression Search
- Decomposed Prime Implicate Generation
- Systematic Goal Orderings by Topological Sort
- Incremental Temporal Consistency Checking



Why is this a good idea?

- Threat-resolution in goal-regression planning is the slowest part.
  - Causal graphs remove a lot of constraint processing
- Heuristic forward-search planners are good at solving cycles in a causal graph.
  - Current search strategies throw away cycles.

Why is this innovative?

- Output temporal, least commitment plan.
- Increased expressiveness of system models.
- Explicit use of Causal Graph.

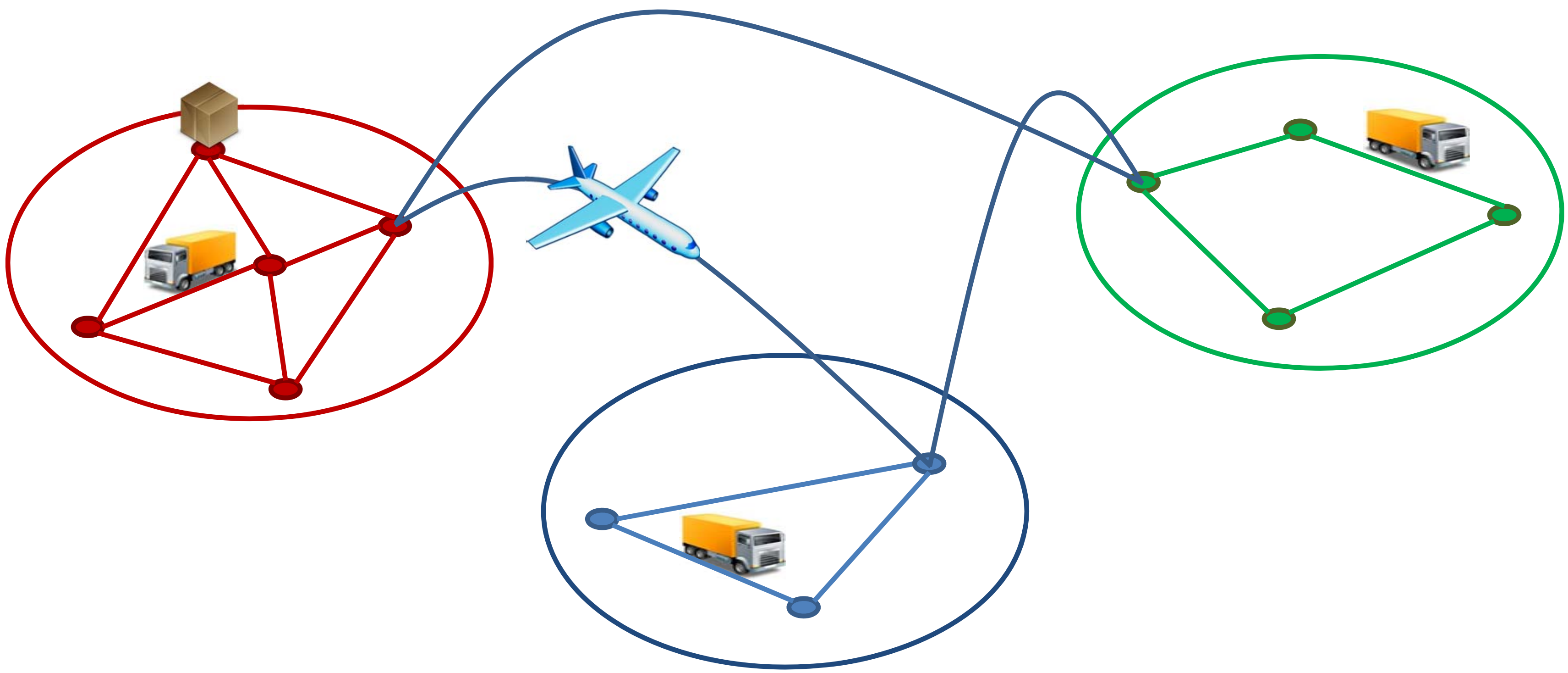
tBurton

- Motivation
- Overview
- **Example**

Example

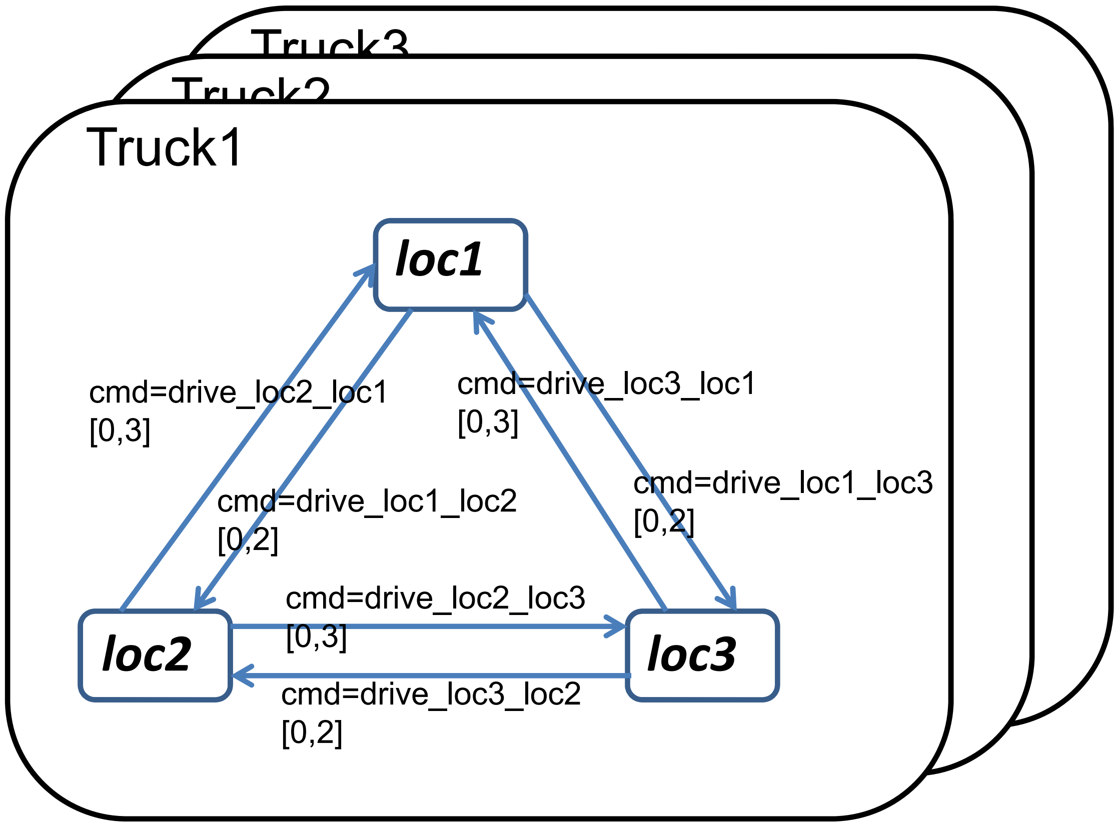
Logistics Example:

- Objective: move packages from locations in cities to other locations.
- Planes move *between* cities
- Trucks move *in* cities



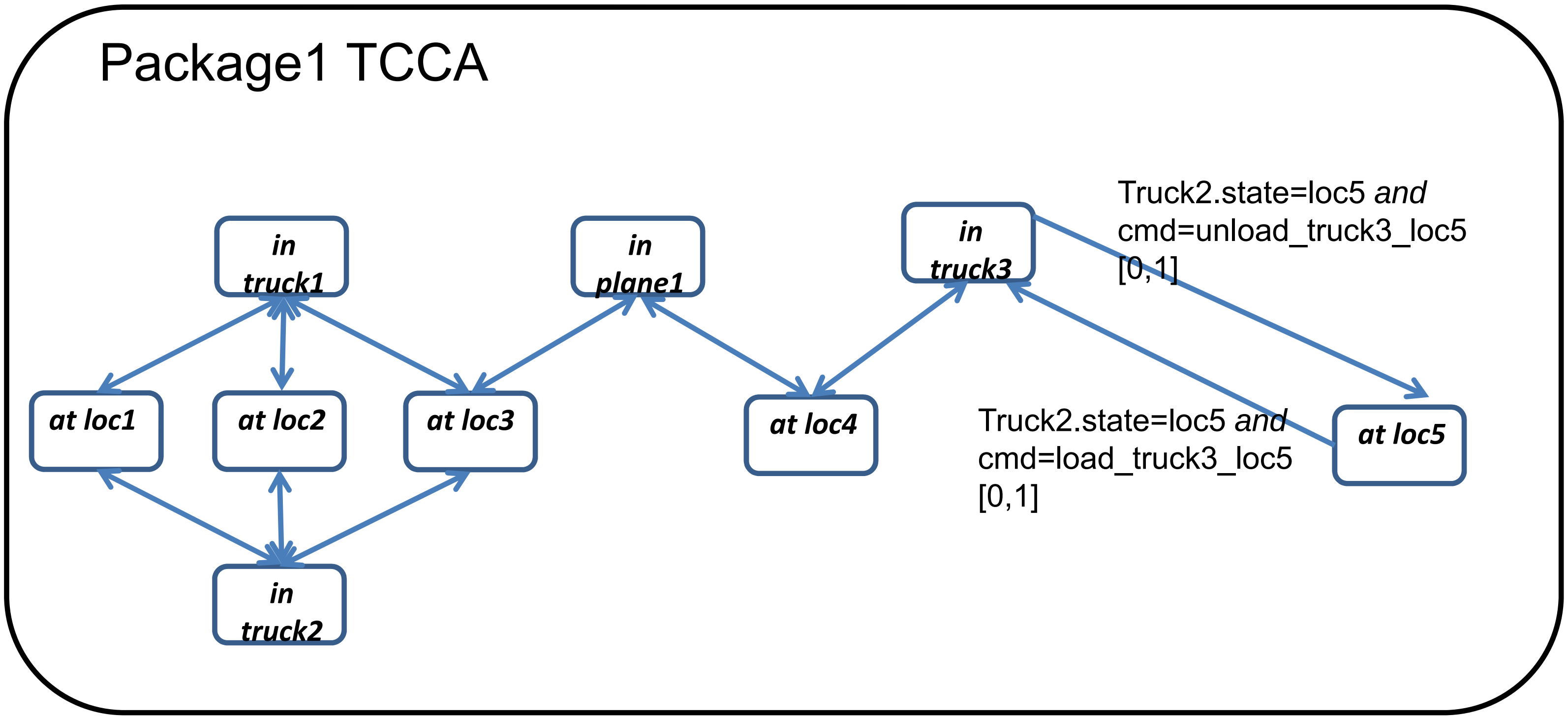
tBurton's Encoding

Model: Timed Concurrent Constraint Automata (TCCA)



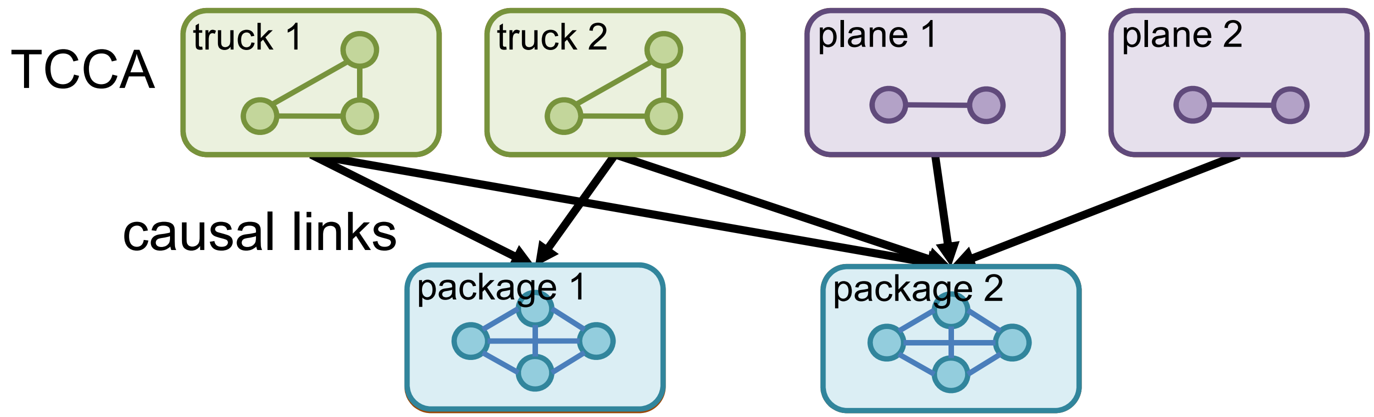
tBurton's Encoding

Model: Timed Concurrent Constraint Automata



Causal Graph

Use a causal graph:

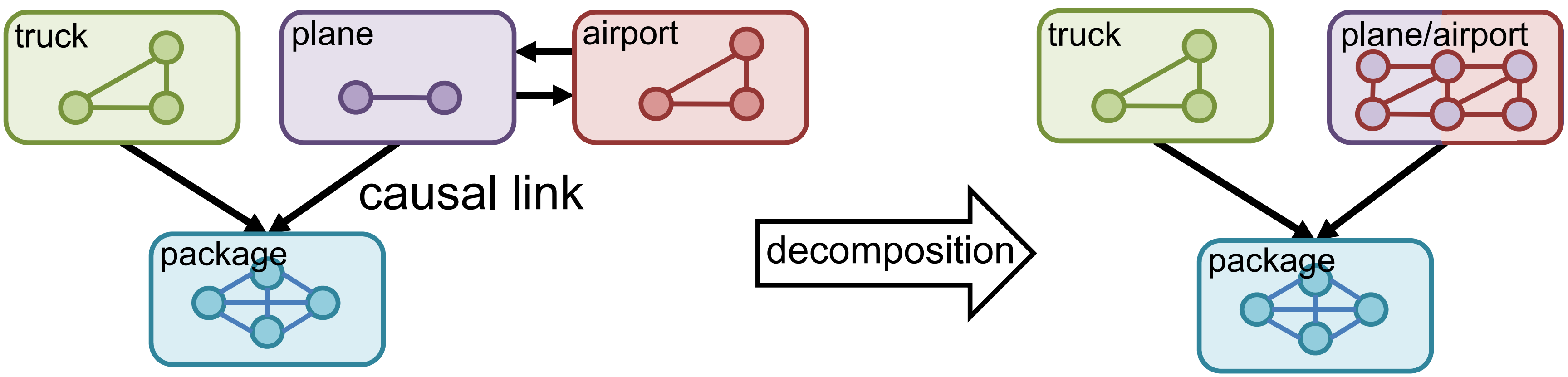


Benefits:
 - Decomposes/composes the problem into planning tasks suitable for heuristic based searches.
 - Captures information, to prioritize planning goals.

i.e. to move a package, we must first determine the route the package must travel, then move the trucks and planes accordingly.

Causal Graph Decomposition

Causal graph – captures device dependencies



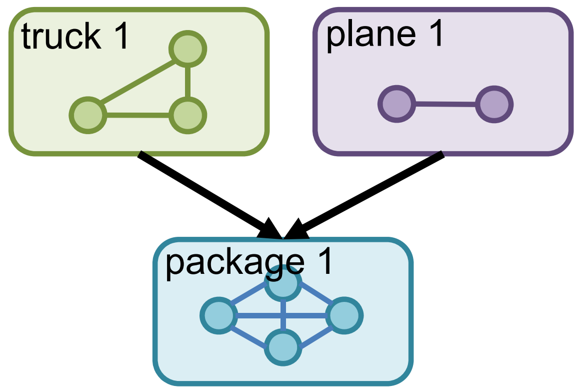
Benefits:

- Decomposes/composes the problem into planning tasks suitable for heuristic based searches.
- Captures information, to prioritize planning goals.

  i.e. to move a package, we must first determine the route the package must travel, then move the trucks and planes accordingly.

Example

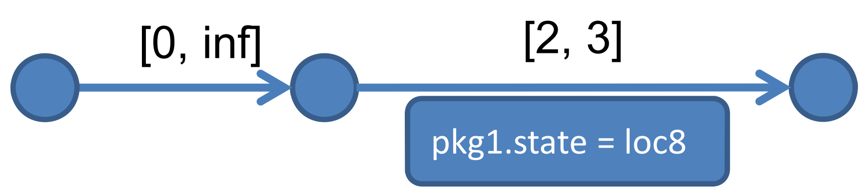
Model: TCCA with Causal Graph:



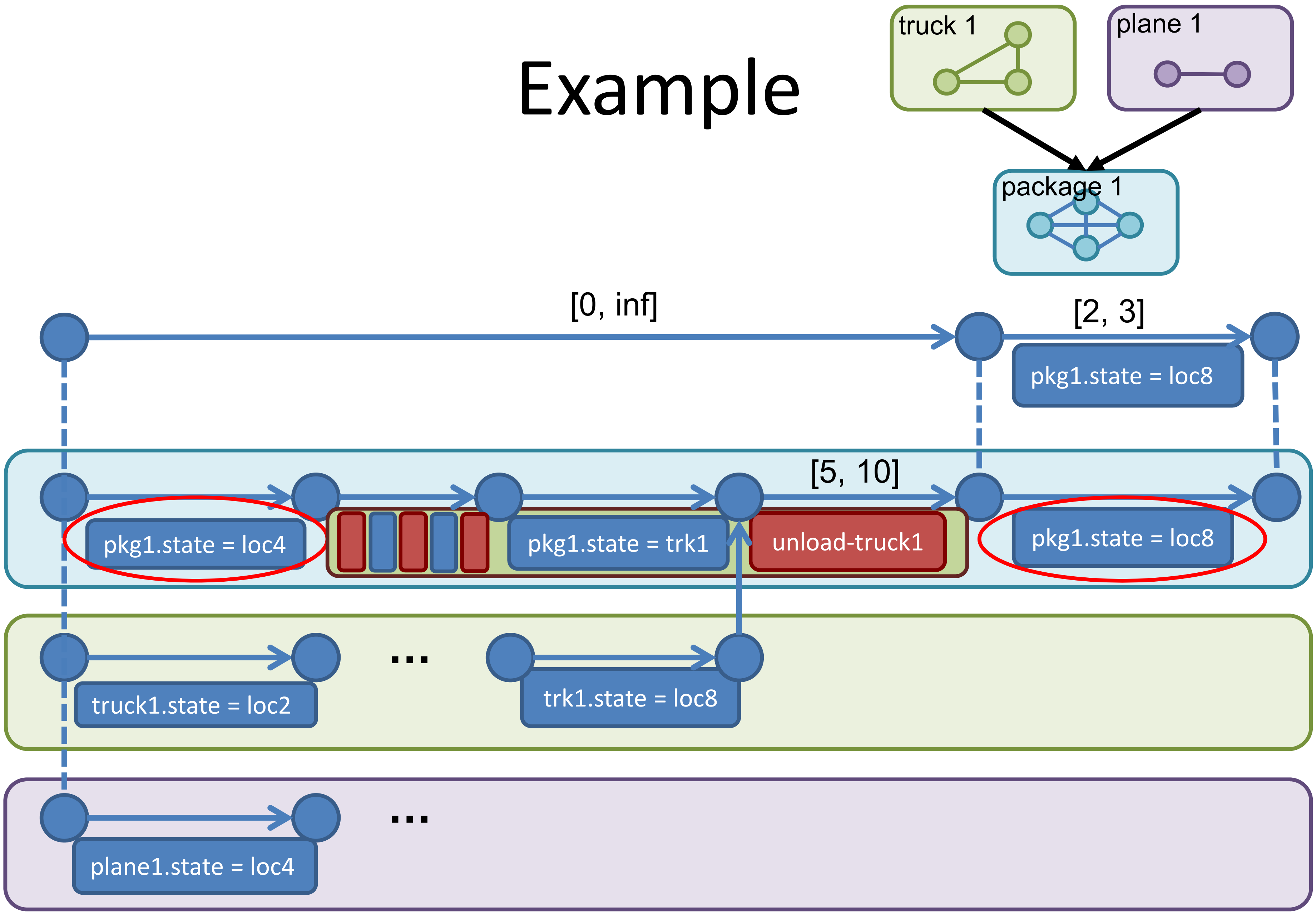
Initial State:

pkg1.state = loc4
truck1.state = loc2
plane1.state = loc4

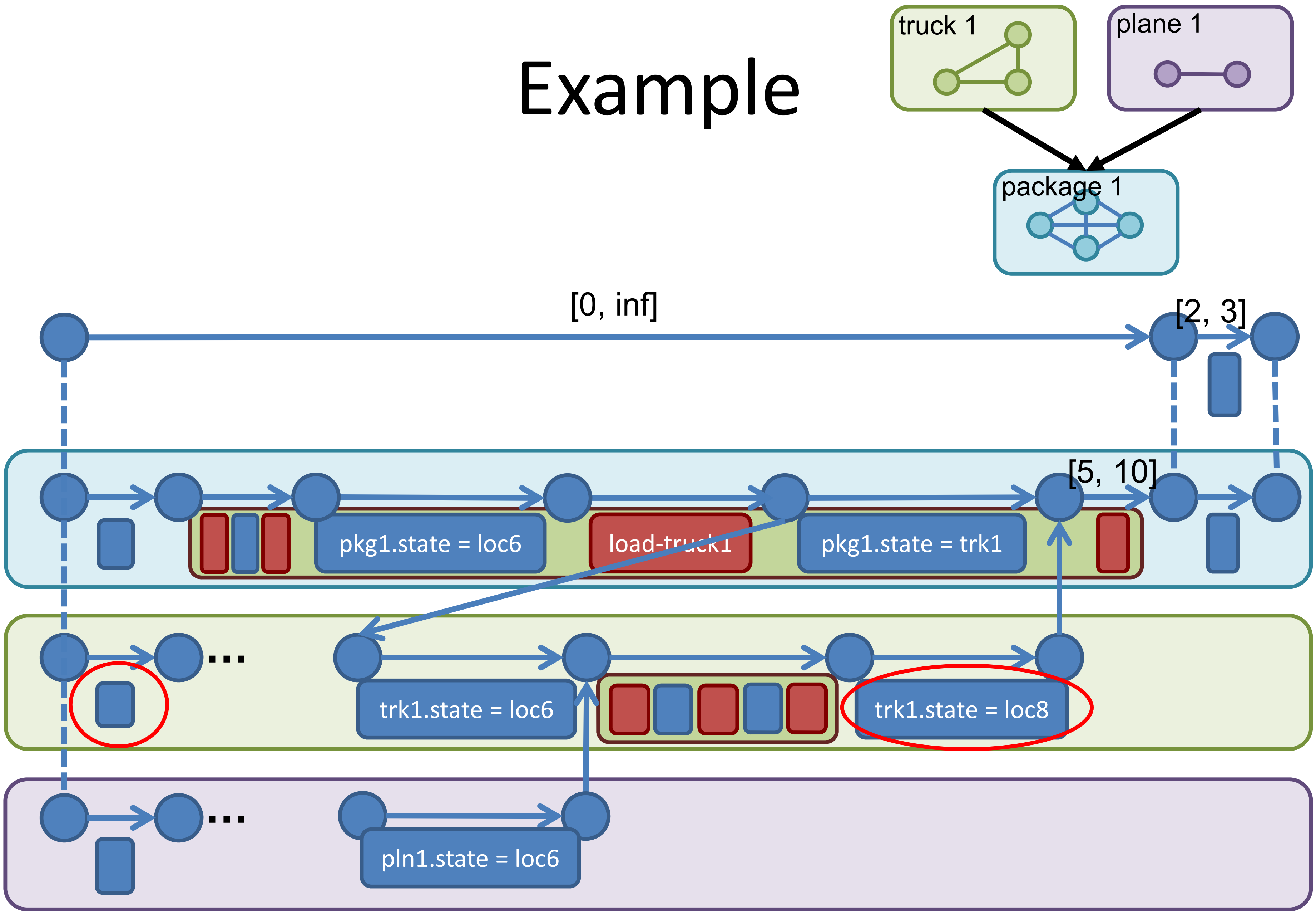
Goal:



Example


truck 1
plane 1
package 1
[0, inf]
[2, 3]
pkg1.state = loc8
[5, 10]
pkg1.state = loc4
pkg1.state = trk1
unload-truck1
pkg1.state = loc8
...
truck1.state = loc2
trk1.state = loc8
...
plane1.state = loc4

Example



truck 1
plane 1
package 1
[0, inf]
[2, 3]
[5, 10]
pkg1.state = loc6
load-truck1
pkg1.state = trk1
…
trk1.state = loc6
trk1.state = loc8
…
pln1.state = loc6

Take-aways

- A model-based executive can alleviate the task of handling off-nominal behavior.
- The **product of an algorithm can support resiliency** as well as the algorithm. (Temporal networks allow flexible execution)
- We can ***exploit the structure of engineered systems*** to quickly plan for a reasonably expressive models.
- The plans produced reflect the ***concurrent*** nature of systems.

[redacted]