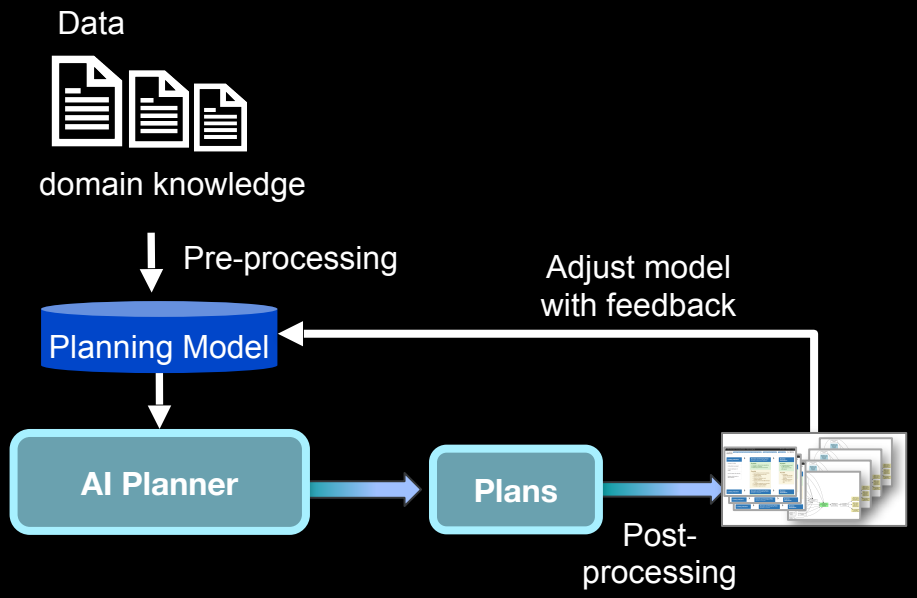
Modeling

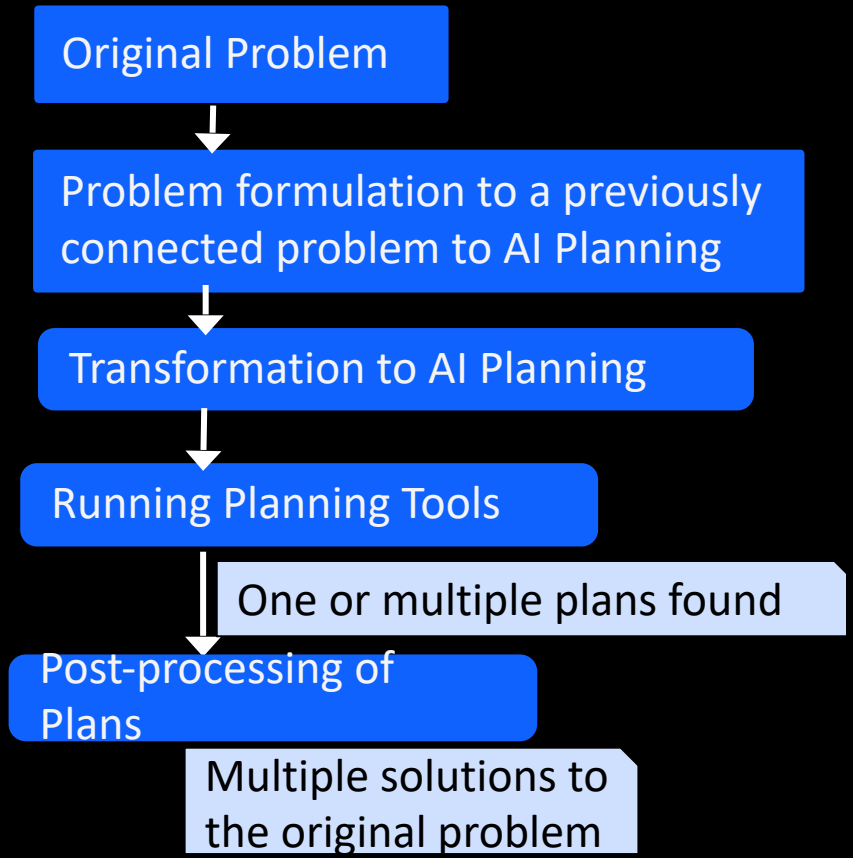


1. Theory
2. Modeling
3. Practice

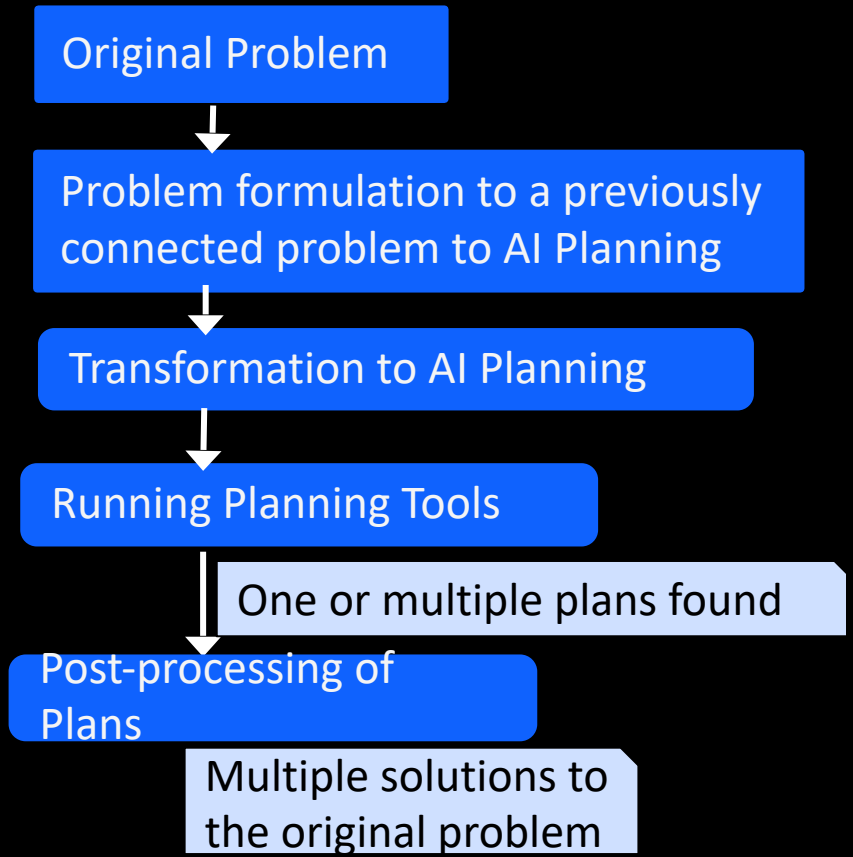
Modeling Challenges

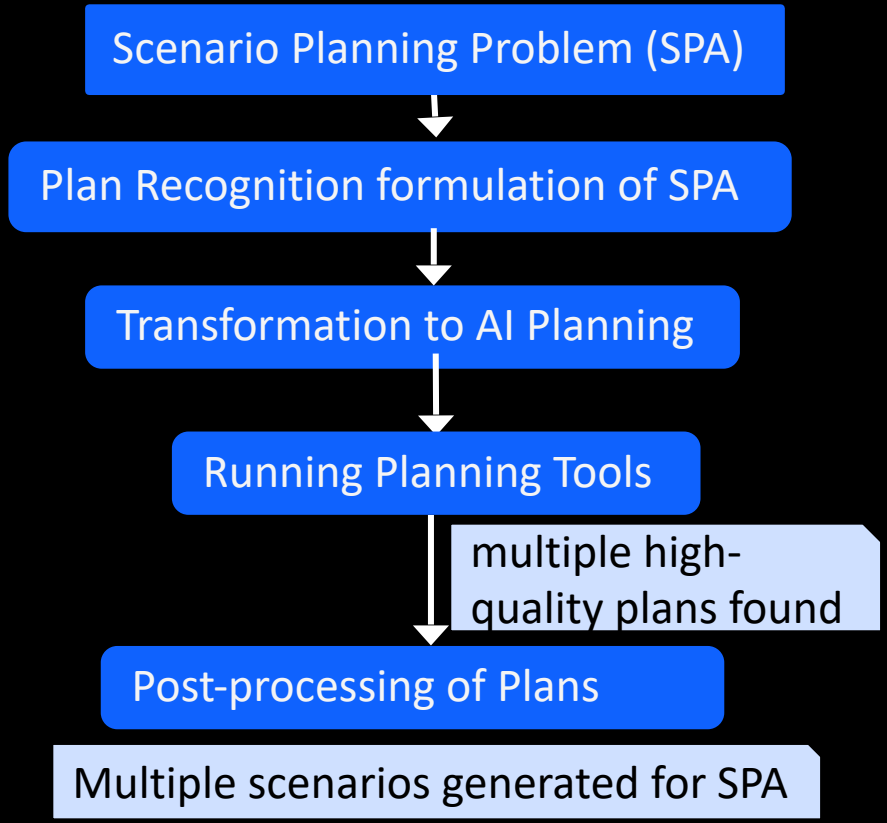
1. Often need to transform the original problem to a problem with known/easier to compute solutions
2. Access to domain experts familiar with input to planners (e.g., PDDL) is rare.
3. Any form of knowledge is practically guaranteed to be incomplete and often inconsistent. Even human validation can be ambiguous
4. Current state-of-art learning approaches may not scale and may have assumptions that may not hold for the practical setting. Further, the learned knowledge may be not consumable

Relationship to Planning



Relationship to Planning





Established relationships to AI Planning

- Finding excuses (G¨obelbecker et al. 2010)
- Diagnosis (Sohrabi et al., 2010)
- Explanation generation (Sohrabi et al., 2011)
- Plan Recognition (Ramirez & Geffner IJCAI’09, AAAI'10, Sohrabi et al., IJCAI’16)

Which in turn helped with:

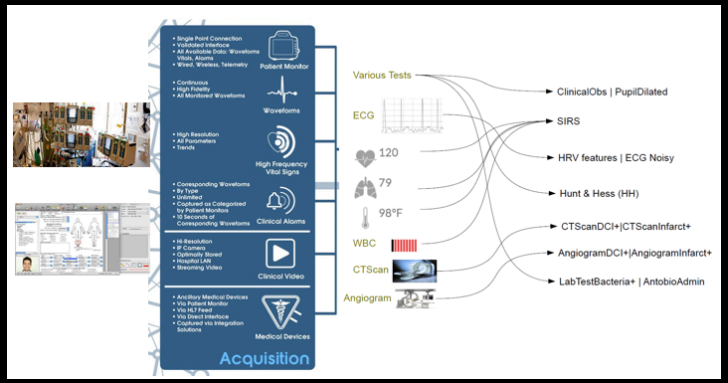
- Hypothesis Generation and Exploration (Sohrabi et al., 2013, 2015)
- Future State Projection (Sohrabi et al., 2017)
- Multi-agent Plan Recognition (Shvo et al., 2018)
- Scenario Planning (Sohrabi et al., 2018)

Additional Compilations

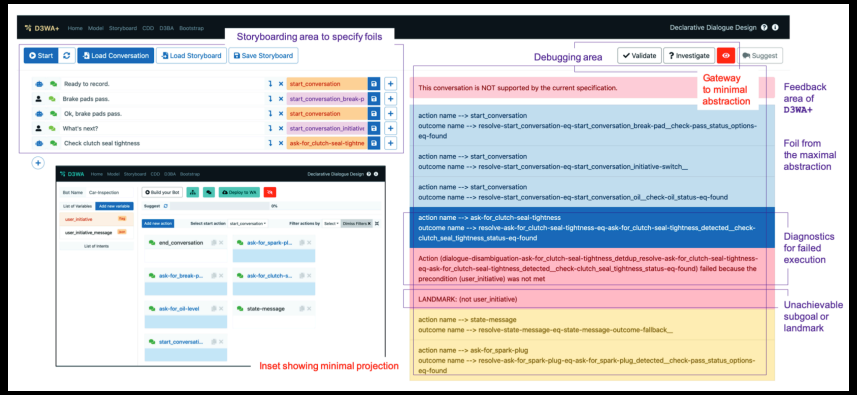
- Compiling away soft goals (Keyder and Geffner, 2009)
  - Use any classical planner to obtain a solution for a planning problem with soft goals (i.e., simple preferences)

- Translate HTN Problems into classical planning problems (Alford et al., 2016)
  - Use classical planners (heuristic search) to find solutions for an HTN planning problem.

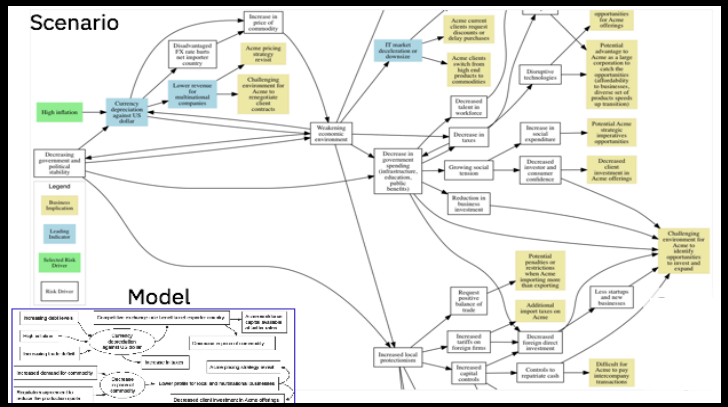
Examples



[Automated large-scale data analysis, ICAPS 2015]



[D3WA+: A Case Study of XAIP in a Model Acquisition Task, ICAPS 2020]



[Scenario planning for enterprise risk management, AAAI 2018]

[Exploring Context-Free Languages via Planning: The Case for Automating Machine Learning, ICAPS 2020]

AutoAI: Automating ML Pipeline Generation

Problem

Space of possible pipelines is huge

Humans can explore only a tiny portion of it

Humans are biased towards the pipelines they already are familiar with

Benefits

- Generation of pipelines of high accuracy automatically
- Reduces the need for human data scientists

Solution

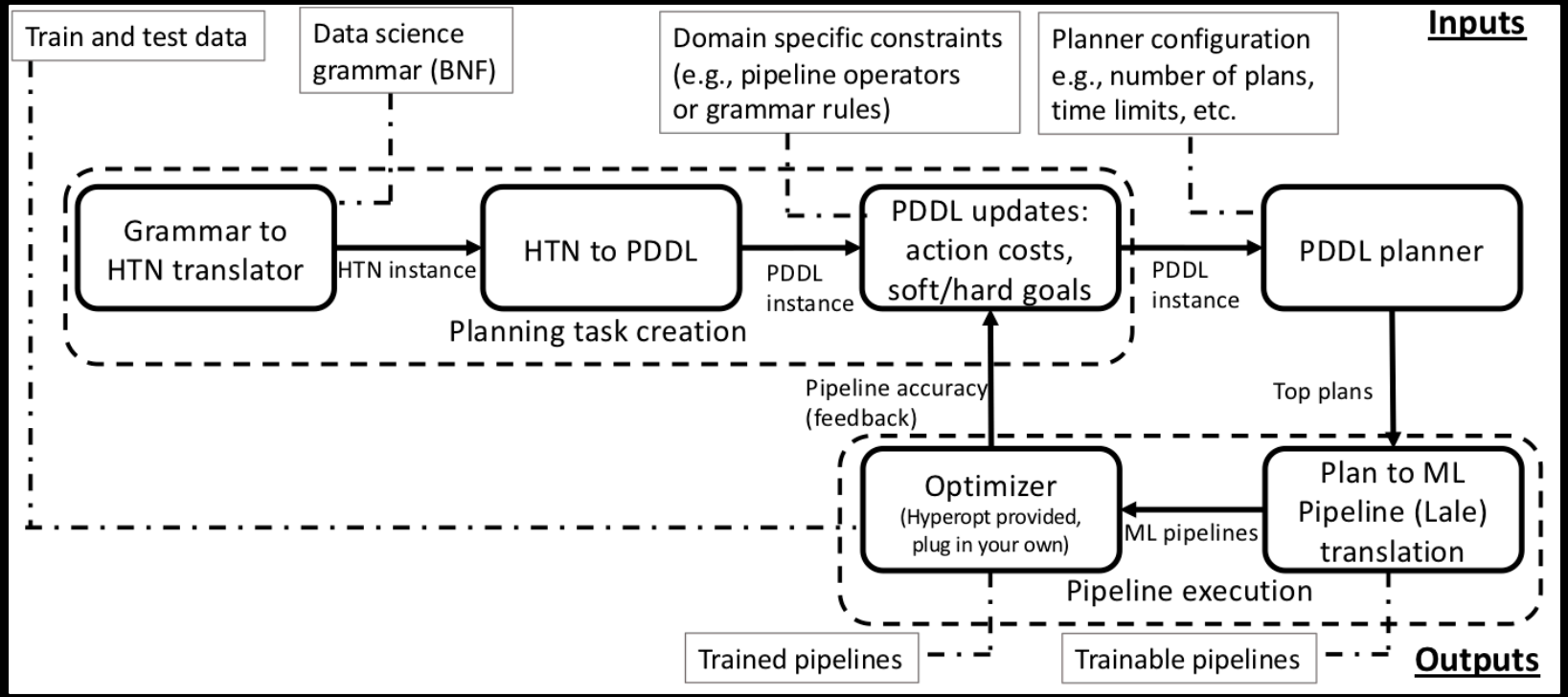
– Use regular grammars to define the possible pipeline compositions

– Translate regular grammars to HTN planning model, and then to classical planning model

– Enrich planning model with user defined constraints

– Use AI Planners to solve the planning model and translate plans to pipelines

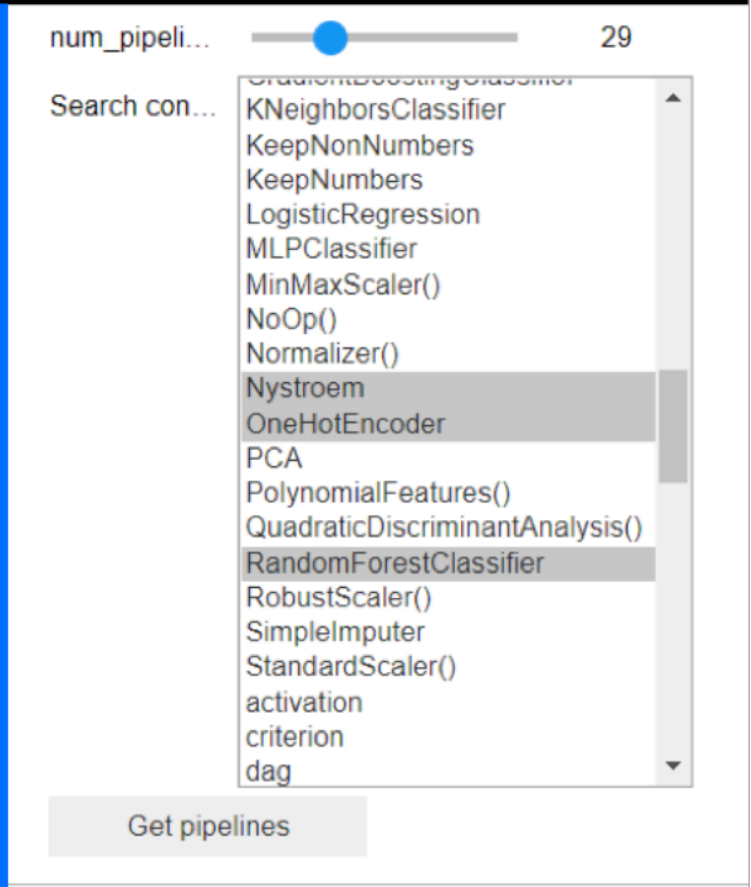
Automating ML Pipeline Generation: Solution



User Guided Exploration: Choosing Preferred Symbols

- **Non-terminal symbols — rules**
- **Terminal symbols — specific algorithms and hyper-param values**

- **Translated to soft goals**
- **Soft goals compiled away (Keyder & Geffner 2009)**



Representing the Knowledge (Example: PDDL)

```
(define (problem mixed)
   (:domain miconic)
   (:objects p0 p1 p2 p3 f0 f1 f2 f3 f4 f5 f6 f7)

(:init
(passenger p0)(passenger p1)(passenger p2)(passenger p3)
(floor f0)(floor f1)(floor f2)(floor f3)(floor f4)
(floor f5)(floor f6)(floor f7)

(above f0 f1)(above f0 f2)(above f0 f3)(above f0 f4)
(above f0 f5)(above f0 f6)(above f0 f7)(above f1 f2)
(above f1 f3)(above f1 f4)(above f1 f5)(above f1 f6)
(above f1 f7)(above f2 f3)(above f2 f4)(above f2 f5)
(above f2 f6)(above f2 f7)(above f3 f4)(above f3 f5)
(above f3 f6)(above f3 f7)(above f4 f5)(above f4 f6)
(above f4 f7)(above f5 f6)(above f5 f7)(above f6 f7)

(origin p0 f0)(destin p0 f5)(origin p1 f7)(destin p1 f4)
(origin p2 f0)(destin p2 f7)(origin p3 f1)(destin p3 f6)

(lift-at f0))

(:goal (and
(served p0)(served p1)(served p2)(served p3))))
```

Problem file

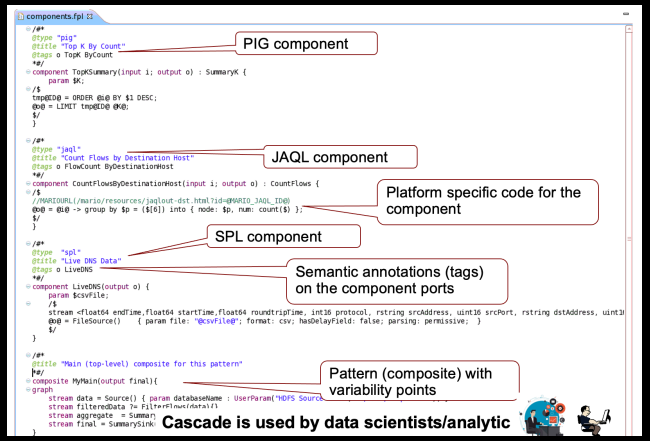
```
(define (domain miconic)
  (:requirements :strips)

(:predicates
(origin ?person ?floor)
(floor ?floor)
(passenger ?passenger)
(destin ?person ?floor)
(above ?floor1 ?floor2)
(boarded ?person)
(served ?person)
(lift-at ?floor))

(:action board
  :parameters (?f ?p)
  :precondition (and (floor ?f) (passenger ?p)(lift-at ?f) (origin ?p ?f))
  :effect (boarded ?p))

(:action depart
  :parameters (?f  ?p)
  :precondition (and (floor ?f) (passenger ?p) (lift-at ?f) (destin ?p ?f)
             (boarded ?p))
  :effect (and (not (boarded ?p))
           (served ?p)))

(:action up
  :parameters (?f1 ?f2)
  :precondition (and (floor ?f1) (floor ?f2) (lift-at ?f1) (above ?f1 ?f2))
  :effect (and (lift-at ?f2) (not (lift-at ?f1))))

(:action down
  :parameters (?f1 ?f2)
  :precondition (and (floor ?f1) (floor ?f2) (lift-at ?f1) (above ?f2 ?f1))
  :effect (and (lift-at ?f2) (not (lift-at ?f1)))))
```

Domain file

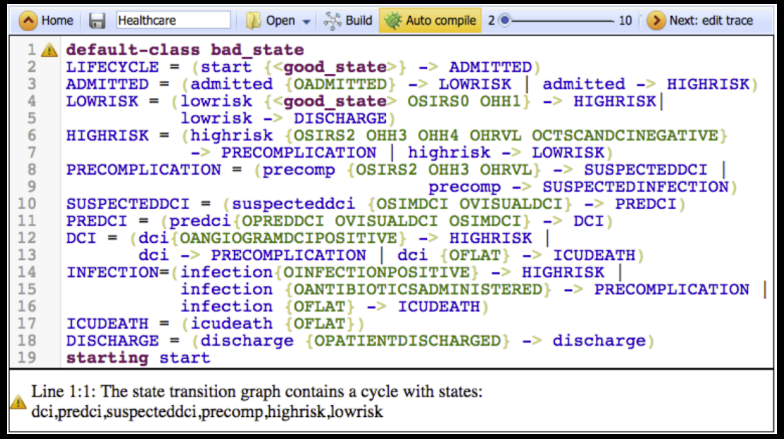
Plan

(up f0 f1)
(up f1 f7)
(board f7 p1)
(down f7 f4)
(depart f4 p1)
(down f4 f0)
(board f0 p0)
(up f0 f5)
(depart f5 p0)
(up f5 f6)
(down f6 f0)
(board f0 p2)
(up f0 f7)
(depart f7 p2)
(down f7 f1)
(board f1 p3)
(up f1 f6)
(depart f6 p3)

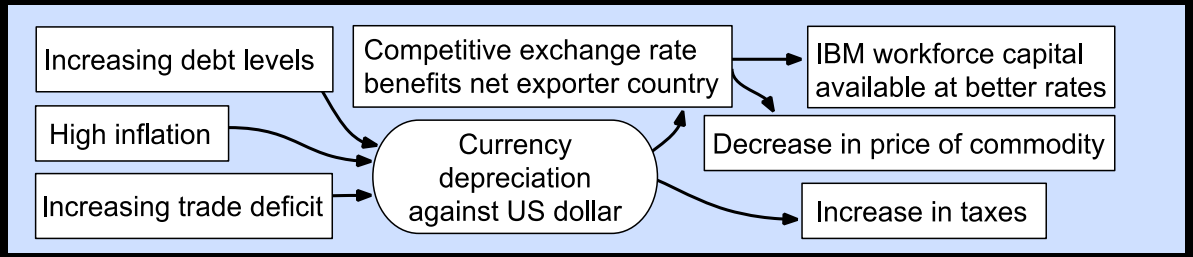
Modeling Examples



Cascade (Ranganathan et al., 2009)

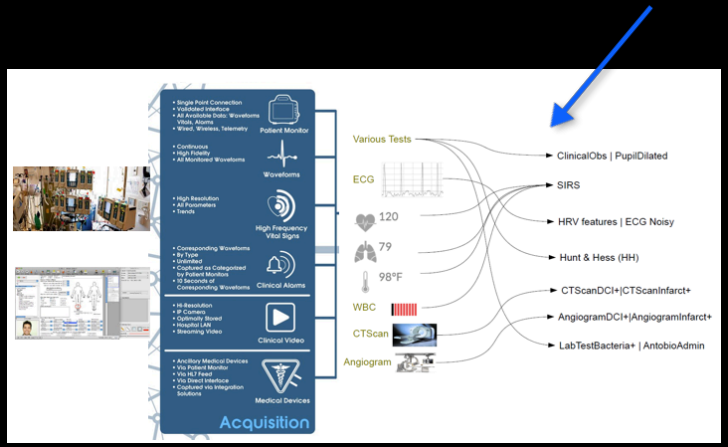


LTS++ (Sohrabi et al., 2015, 2020)

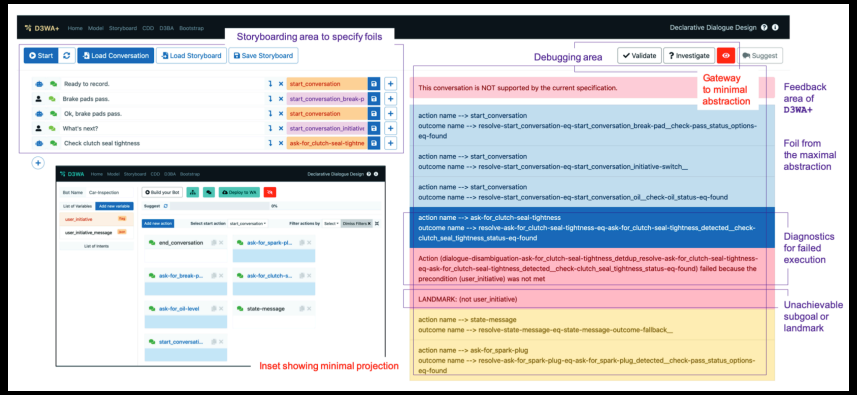


MindMaps
(Sohrabi et al., 2018)

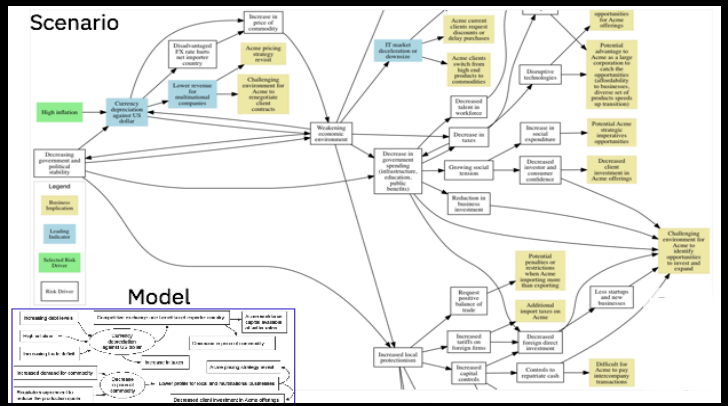
Examples



[Automated large-scale data analysis, ICAPS 2015]



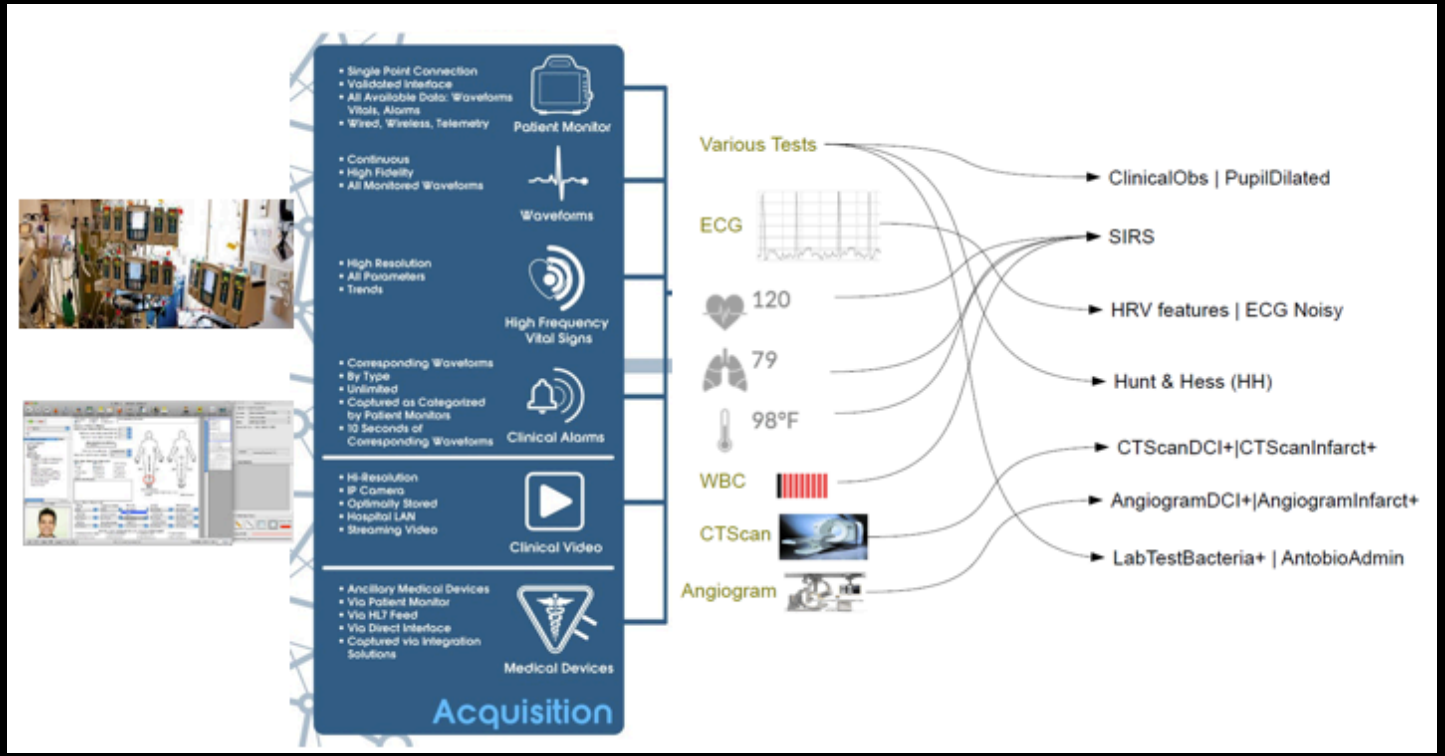
[D3WA+: A Case Study of XAIP in a Model Acquisition Task, ICAPS 2020]



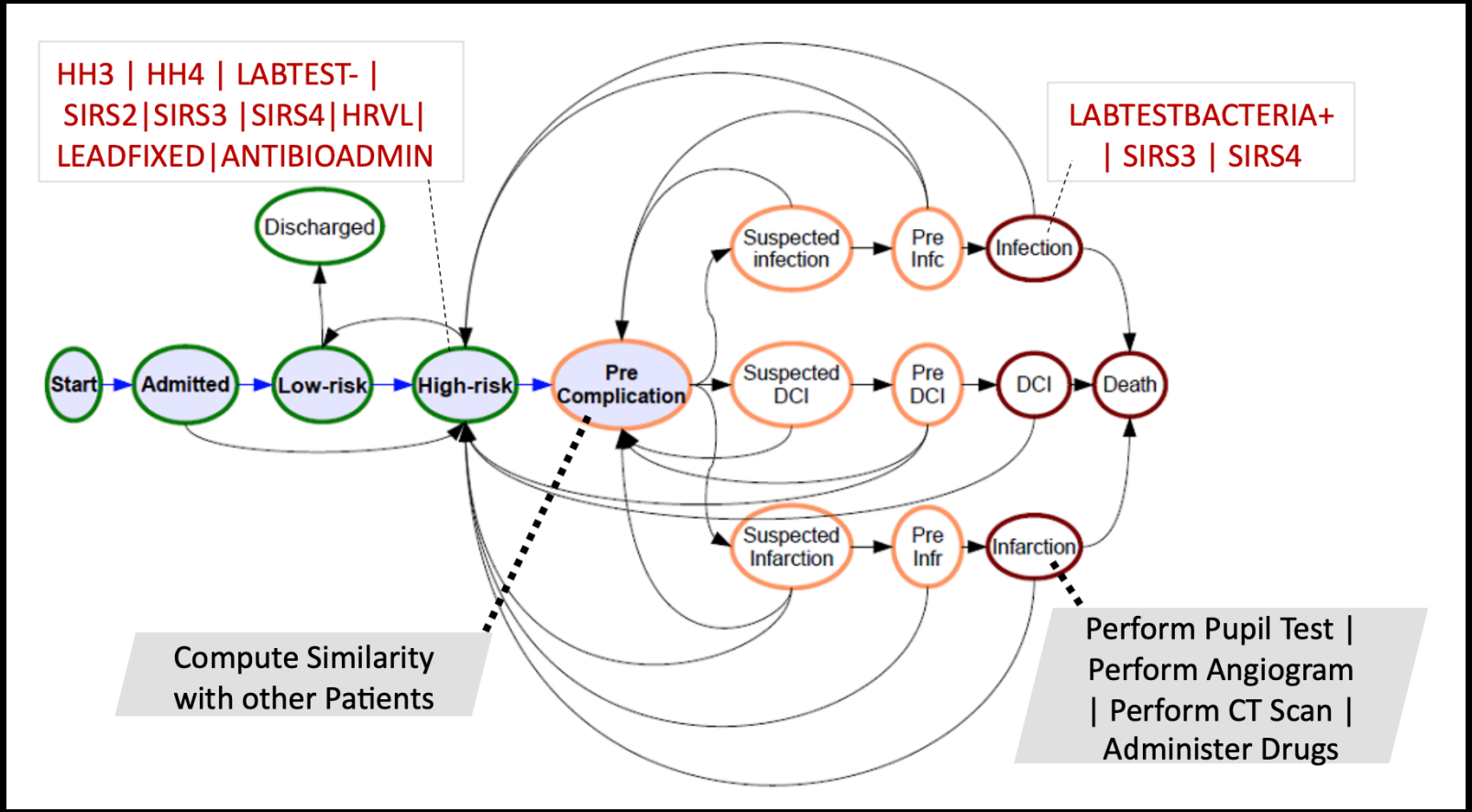
[Scenario planning for enterprise risk management, AAAI 2018]

[Exploring Context-Free Languages via Planning: The Case for Automating Machine Learning, ICAPS 2020]

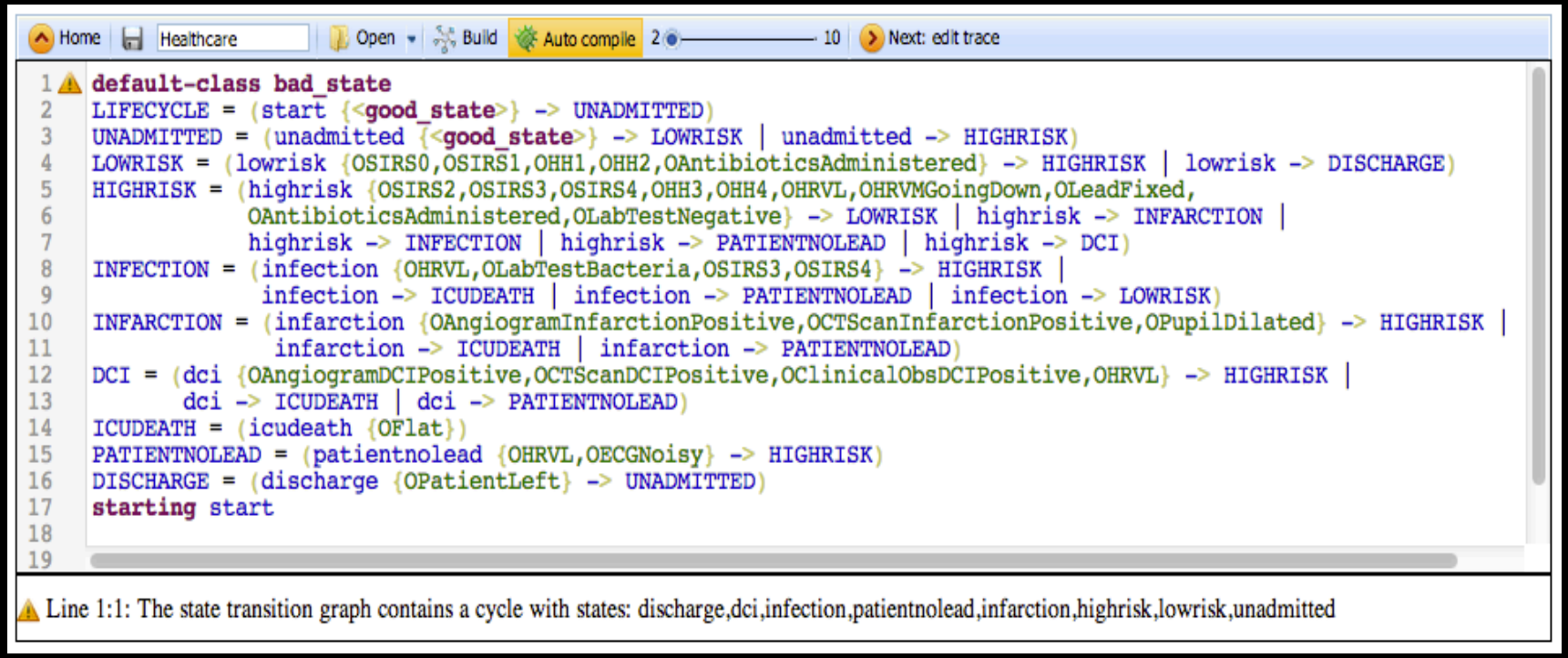
Hypothesis Generation



Patient Complication Detection



LTS++



- Derived from an existing language, Labeled Transition System (LTS)
- Can associate an observation with a state, can specify class type

Partial encoding of the sample example

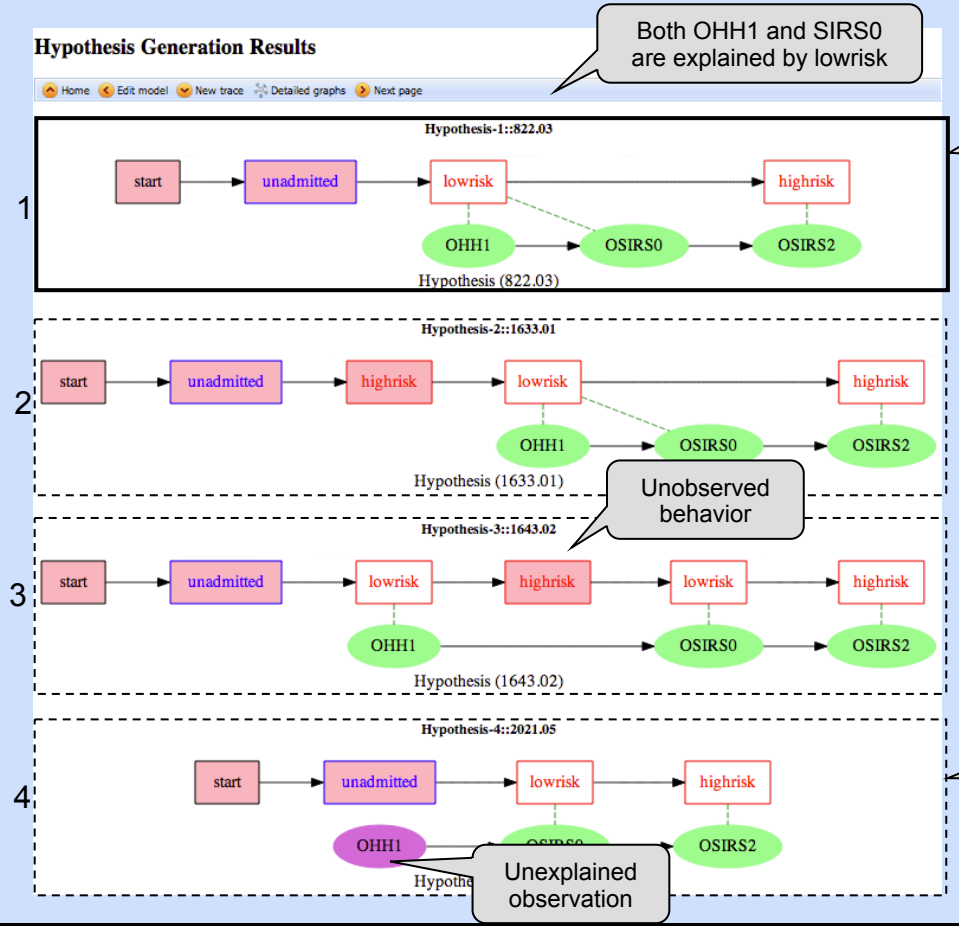
```
(:action explain-observation
  :parameters(?x - state ?obs1 - obs ?obs2 - obs ?cat - obs-type)
  :precondition (and (is-next-obs ?obs1 ?obs2)(matches ?obs1 ?cat)
                     (explains ?x ?cat)(at-state ?x)(at-obs ?obs1))
  :effect (and  (not (at-obs ?obs1)) (at-obs ?obs2) (ready)
                     (increase (total-cost) 0)))

 (:action discard-observation
  :parameters(?x - state ?obs1 - obs ?obs2 - obs)
  :precondition (and (is-next-obs ?obs1 ?obs2)(at-state ?x)(at-obs ?obs1))
  :effect (and (not (at-obs ?obs1))(at-obs ?obs2)(ready)
                 (increase (total-cost) 2000)))

 (:action state-change
  :parameters(?x - state ?y - state ?obs - obs)
  :precondition (and (is-next-state ?x ?y)(at-obs ?obs)(at-state ?x)(ready))
  :effect (and (not (at-state ?x))(not (ready))(entering-state ?y)
                   (increase (total-cost) 0)))

 (:action enter-state-good
  :parameters(?y - state ?obs - obs)
  :precondition (and (at-obs ?obs) (entering-state ?y) (good-state ?y))
  :effect (and (at-state ?y)(not (entering-state ?y))
               (increase (total-cost) 1)))

 (:action enter-state-bad
  :parameters(?y - state ?obs - obs)
  :precondition (and (at-obs ?obs)(entering-state ?y)(bad-state ?y))
  :effect (and (at-state ?y)(not (entering-state ?y))
               (increase (total-cost) 10)))

 (:action allow-unobserved
  :parameters(?x - state ?obs - obs)
  :precondition (and (at-obs ?obs)(at-state ?x))
  :effect (and (ready)(increase (total-cost) 1100)))
```

Partial encoding of the sample example

```
(:init
  (at-state admitted) (at-obs o_1)(ready)

  (matches o_1 OHH1)(matches o_2 OSIRS0)(matches o_3 OSIRS2)

  (explains lowrisk OSIRS0) (explains highrisk OSIRS2)
  (explains precomp OSIRS2) (explains lowrisk OHH1)
  (explains dci OANGIOGRAMDCIPOSITIVE)
  (explains highrisk OHRVL) (explains precomp OHRVL)

  (is-next-state admitted highrisk) (is-next-state admitted lowrisk)
  (is-next-state lowrisk highrisk) (is-next-state highrisk lowrisk)
  (is-next-state highrisk precomp) (is-next-state dci highrisk)
  (is-next-state dci icudeath) (is-next-state dci precomp)

  (bad-state dci) (bad-state highrisk) (good-state lowrisk)

  (is-next-obs o_1 o_2)(is-next-obs o_2 o_3) (is-next-obs o_3 o_end))

(:goal (and (at-obs o_end) (ready)))
```

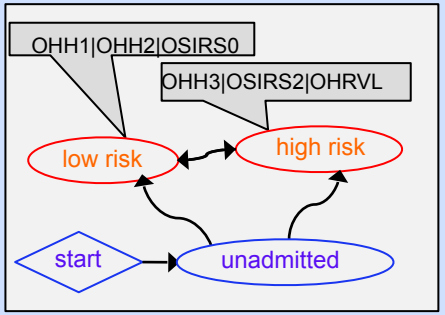
```
(:action explain-observation
  :parameters(?x - state ?obs1 - obs ?obs2 - obs ?cat - obs-type)
  :precondition (and (is-next-obs ?obs1 ?obs2)(matches ?obs1 ?cat)
                     (explains ?x ?cat)(at-state ?x)(at-obs ?obs1))
  :effect (and  (not (at-obs ?obs1)) (at-obs ?obs2) (ready)
                     (increase (total-cost) 0)))

 (:action discard-observation
  :parameters(?x - state ?obs1 - obs ?obs2 - obs)
  :precondition (and (is-next-obs ?obs1 ?obs2)(at-state ?x)(at-obs ?obs1))
                     bs ?obs1))(at-obs ?obs2)(ready)
                     se (total-cost) 2000)))


                     ?y - state ?obs - obs)
                     next-state ?x ?y)(at-obs ?obs)(at-state ?x)(ready))
                     tate ?x))(not (ready))(entering-state ?y)
                     ease (total-cost) 0)))

                     d
                     ?obs - obs)
                     obs ?obs) (entering-state ?y) (good-state ?y))
                     ?y)(not (entering-state ?y))
                     (total-cost) 1)))


                     ?obs - obs)
                     obs ?obs)(entering-state ?y)(bad-state ?y))
                     ?y)(not (entering-state ?y))
                     (total-cost) 10)))

(:action allow-unobserved
 :parameters(?x - state ?obs - obs)
 :precondition (and (at-obs ?obs)(at-state ?x))
 :effect (and (ready)(increase (total-cost) 1100)))
```

Generated hypotheses for the critical care application



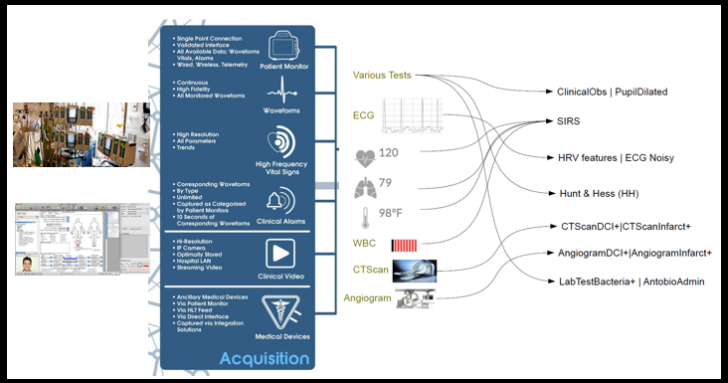
Each hypothesis is shown as sequence of states matched to observed event sequence

State Transition Diagram

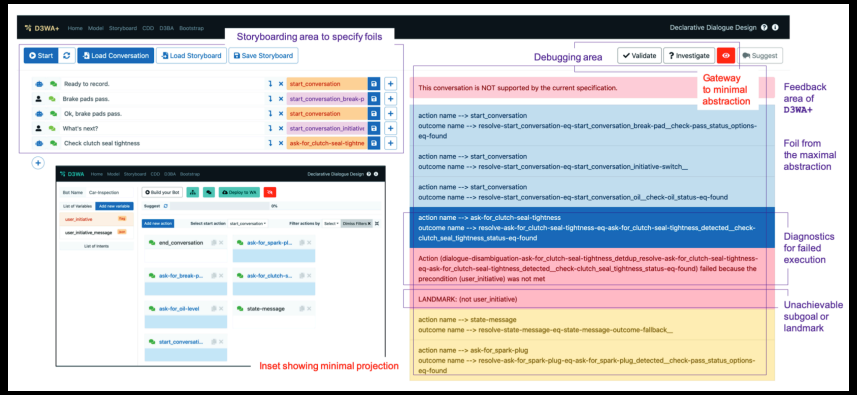


Less plausible hypothesis

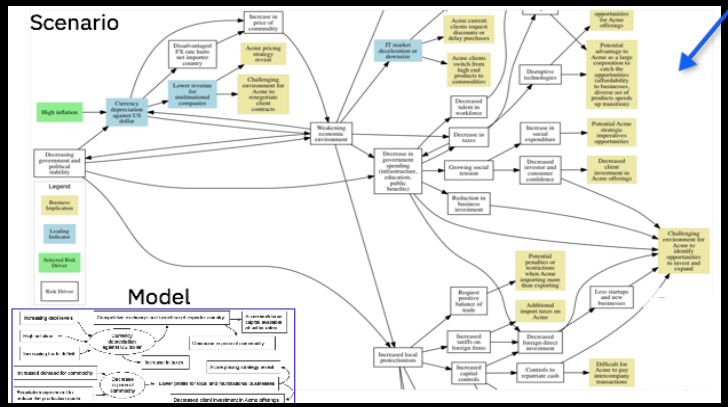
Examples



[Automated large-scale data analysis, ICAPS 2015]



[D3WA+: A Case Study of XAIP in a Model Acquisition Task, ICAPS 2020]



[Scenario planning for enterprise risk management, AAAI 2018]

[Exploring Context-Free Languages via Planning: The Case for Automating Machine Learning, ICAPS 2020]

Scenario Planning for Enterprise Risk Management

Main forces:

Economic environment

Technology

Currency

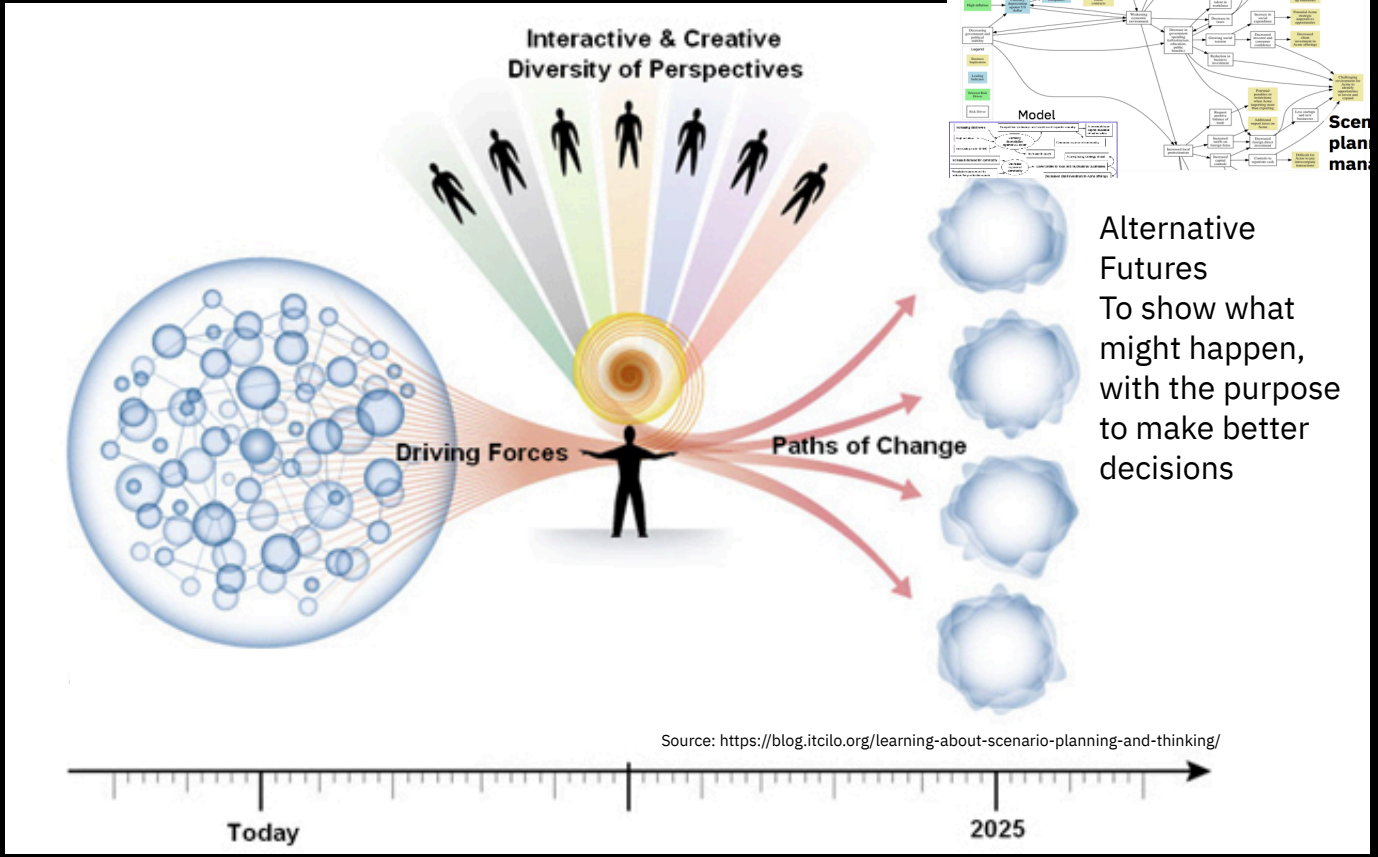
Social order/unrest

Corruption

Natural disaster

Market disruptors

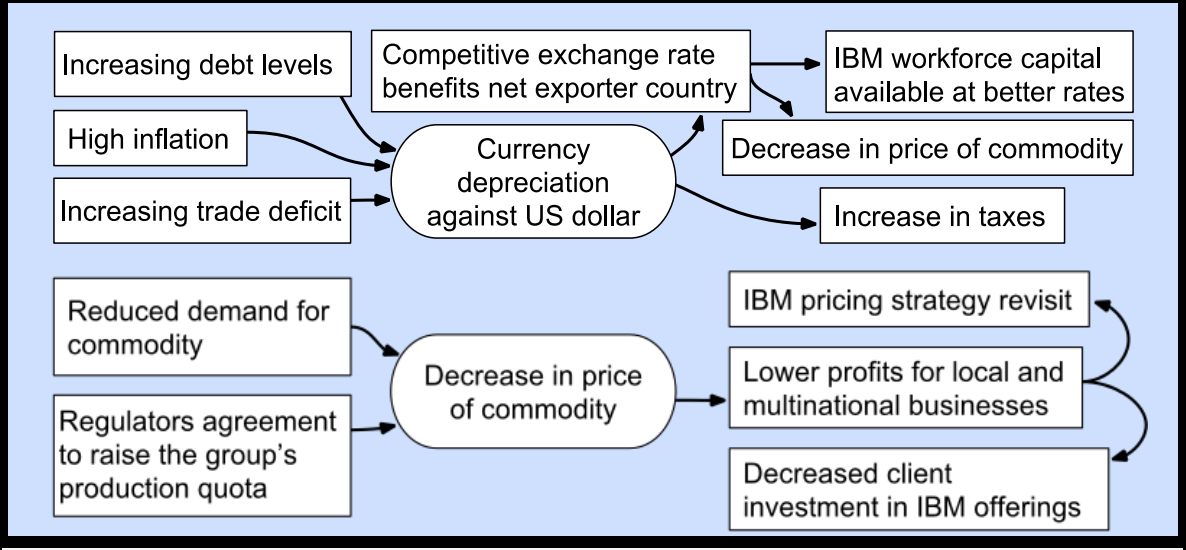
Government Stability

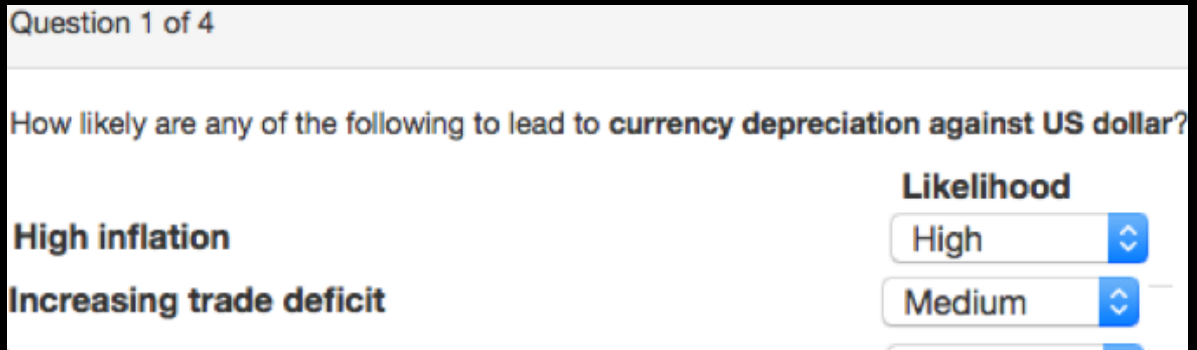


Alternative Futures
To show what might happen, with the purpose to make better decisions

Source: https://blog.itcilo.org/learning-about-scenario-planning-and-thinking/

Mindmaps (in the context of Scenario Planning Advisor)





Planning Models for Scenario Planning – How Acquired?

Manual creation of scenario planning models is impracticable.

Automated extraction of causal models (Hassanzadeh et al. 2019, Bhandari et al., 2021)

- use learning-based natural language understanding techniques to identify risk drivers and extract causal pairs
- AI QA for automated reading comprehension
- with causal questions (“what causes X”) no supervision is needed
- authoritative documents that are rich in causal content (10-K, NATO SFA)
- with open-ended questions based on seed sets of risks, discover new candidate risks, bootstrap

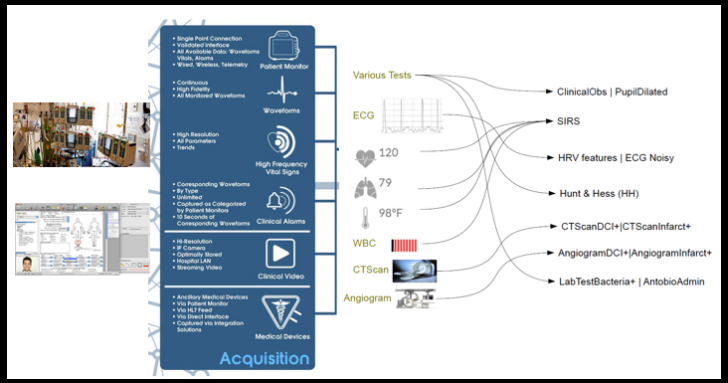
Causal Extraction in Support of Planning Model Generation

CEFAS – **C**ausal **E**xtraction **F**rom **A**uthoritative **S**ources

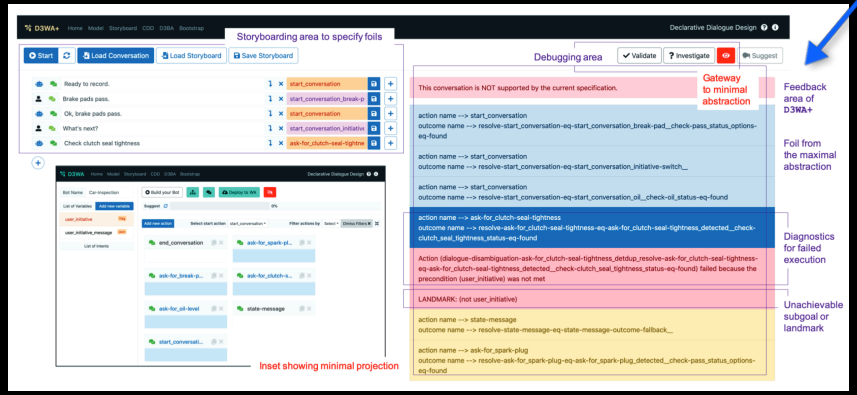
- uses AI Question Answering to answer causal questions about risk drivers…
- relies on relatively small set of authoritative documents, e.g.,
  - SEC Form 10-K – Item 1A Risk Factors
  - NATO Strategic Foresight Analysis doc, 2017
- CEFAS processes all input documents, asking of each paragraph of each document and for each known risk driver:

What causes D?
What does D cause?

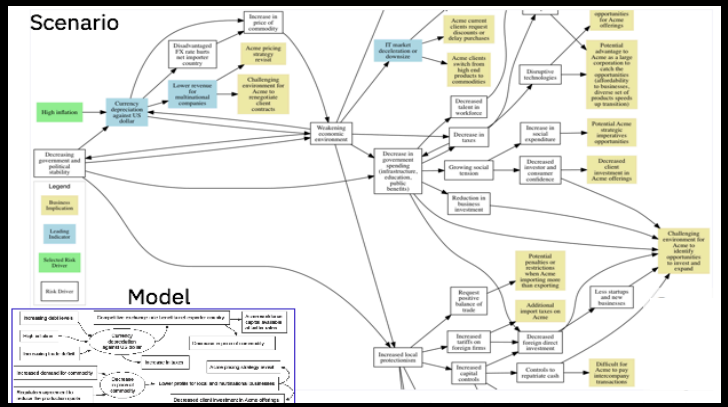
Examples



[Automated large-scale data analysis, ICAPS 2015]



[D3WA+: A Case Study of XAIP in a Model Acquisition Task, ICAPS 2020]

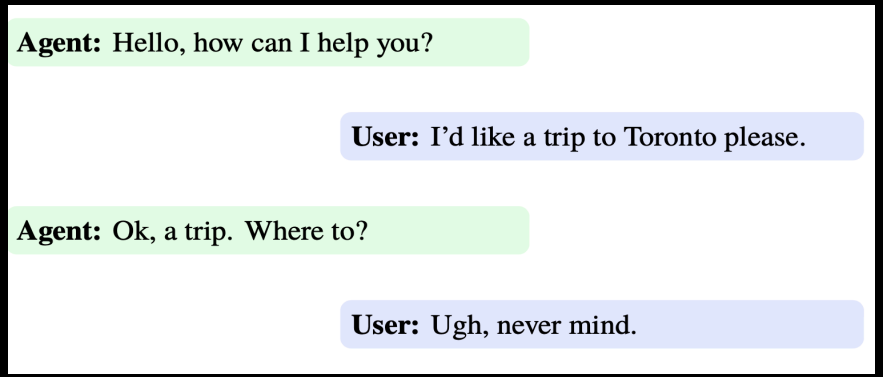


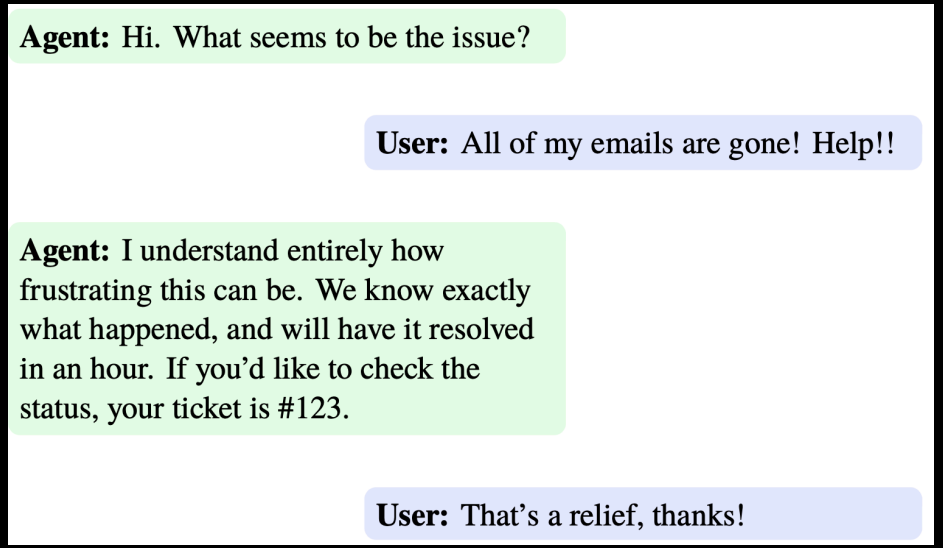
[Scenario planning for enterprise risk management, AAAI 2018]

[Exploring Context-Free Languages via Planning: The Case for Automating Machine Learning, ICAPS 2020]

State of Conversational Agents Today

- **Players:** Google's DialogueFlow, IBM's Watson Assistant, Apple's Siri, Amazon's Alexa, Microsoft's Cortana, etc.

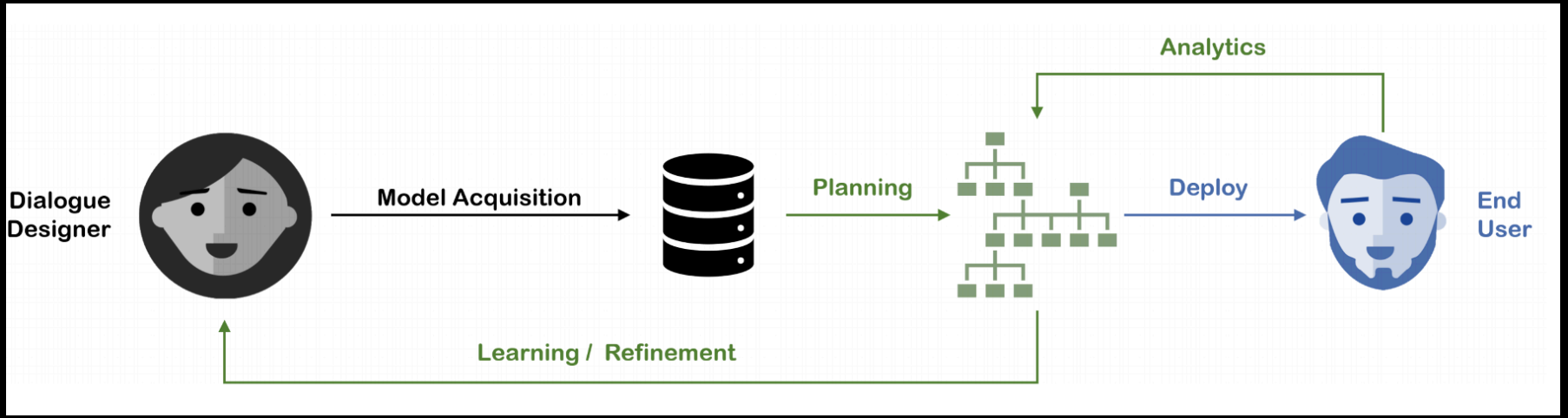




Dialog as Planning

Setting : Multi-turn goal-oriented dialogue

Challenge: How to scale beyond explicit trees



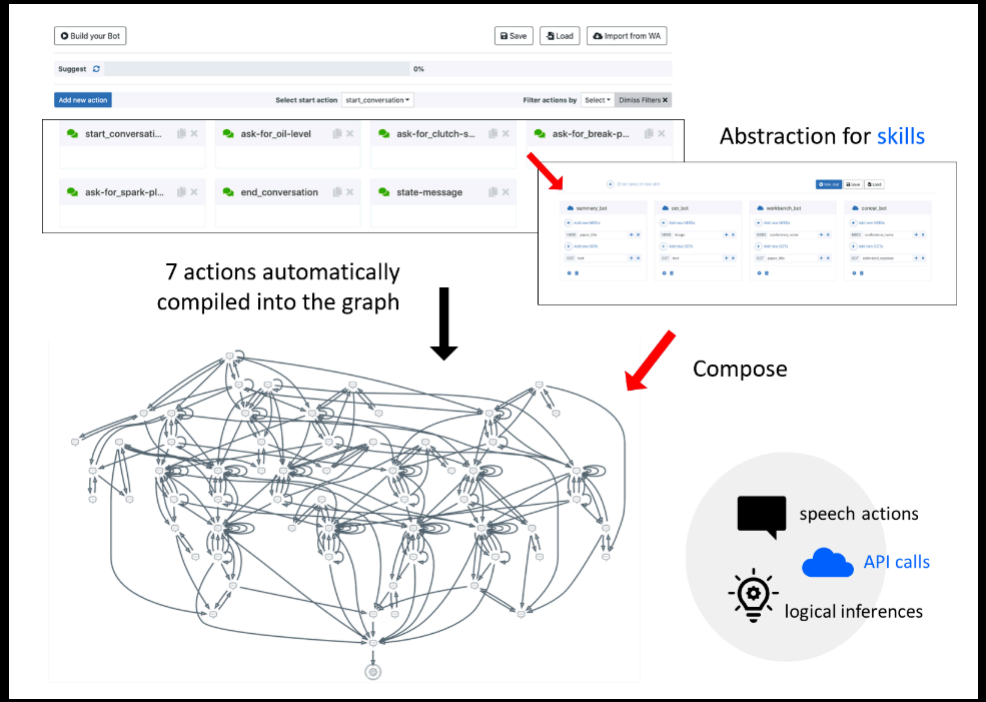
Muise et al., 2019. Planning for Goal-Oriented Dialogue Systems.
Sreedharan et al., 2020. Explainable Composition of Aggregated Assistants
Rizk et al., 2020. A Unified Conversational Assistant Framework for Business Process Automation.

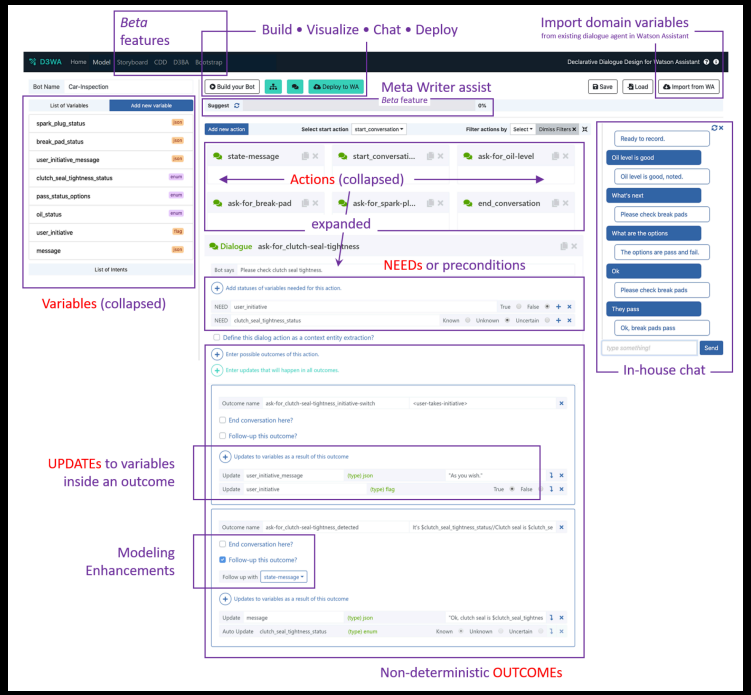
D3WA Exponential scale-up of sophistication of composed conversations

Exponential scale-up means

Increased sophistication of dialogue tree with same size of specification

Decreased size of specification for same sophistication of dialogue tree

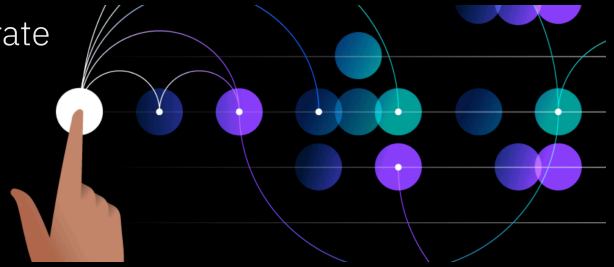




Natural Language Processing x AI Planning

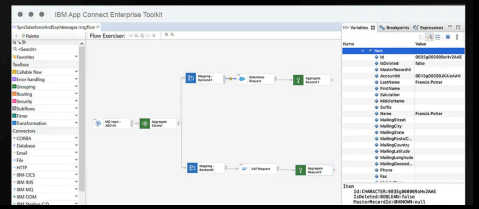
IBM Watson Orchestrate

It's time to work differently by giving teams their own digital employee



IBM App Connect

Connect your apps to automate your business



- Conversational agents realized as an aggregation of skill
- At runtime, planners can be used to organize assets dynamically to fulfill user goals extracted from natural language utterances.

- ICAPS 2021 Demo: https://youtu.be/K7FPcl-IYgE
- AAAI 2022 Demo: ibm.biz/gofa-aaai-demo
- BPM 2021 Tutorial: ibm.biz/bpm-2021-tutorial

Related Work

- ItSimple (Vaquero et al., 2007)
- PDDL Editor http://editor.planning.domains/
- PDDL in Python https://github.com/IBM/pddl-in-python
- Planimation https://github.com/AI-Planning/planimation
- PDDL plugin for VSCode
- Workshop on Knowledge Engineering for Planning and Scheduling (KEPS) @ ICAPS
- PDDL book titled *"An Introduction to the Planning Domain Definition Language"* (Haslum et al., 2020)
- Extensive literature on planning and learning (Aineto et al., AIJ 2019)

Modeling Summary

- Modeling in an important aspect of use and adaptation of AI Planning
- Investing in transformation/compilation techniques pays off
- Learning consumable planning models or even enhancing a planning model automatically is important