Answer Set Programming Approach to Deep Reasoning

Joohyung Lee

Associate Professor, CIDSE

Automated Reasoning Group

Automated Reasoning Group at ASU

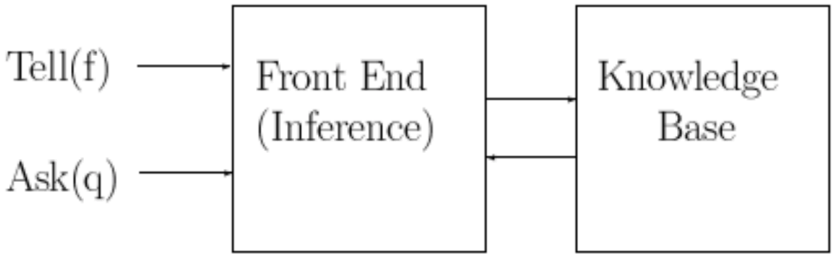
- We're interested in designing and building intelligent systems, which can represent knowledge in a formal language and manipulate it in an autonomous way. Such a system should be able to handle open-ended tasks, acting intelligently by "thinking."

Gottfried Wilhelm Leibniz (1646-1716): Early AI Researcher?

- Leibniz believed that much of human reasoning could be reduced to calculations of a sort:
- *"The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right."*
- How do we build a calculator that computes deep reasoning, not just numbers?

AI and KRR

- AI can be described as: the study of intelligent behavior achieved through computational means
- Knowledge representation and reasoning (KRR) could then be viewed as the study of how to reason (compute) with knowledge in order to decide what to do
  - Think about knowledge, not only data



Learning vs. Reasoning

- Learning: learn models/knowledge from data
- Reasoning: manipulate models/knowledge to derive new information
- Currently there is a significant gap between models learned by Machine Learning and knowledge required by Knowledge Representation and Reasoning.

"Google's TensorFlow Alone Will Not Revolutionize AI"

- Erik Muller, Wired, 11/13/2015
- "*Suppose we train an ANN to recognize cats. When it recognizes a previously unseen cat in an image, it can't explain to us why or how it did this. And if the ANN fails to recognize a spotted cat, it's hard for us to fix the problem. We're not going to tell it something like, "change element 341375's value from 0.3265 to 0.4271, element 1954236's value from 0.9218 to 0.8612, …"*
- *"When AI systems make bad decisions (as they've done before, and inevitably will again), we need to be able to understand why they made those decisions and communicate with them to fix the problem."*
- How do we trust AI's decision if it can't tell us why?
  - Knowledge-poor AI can't explain the reason

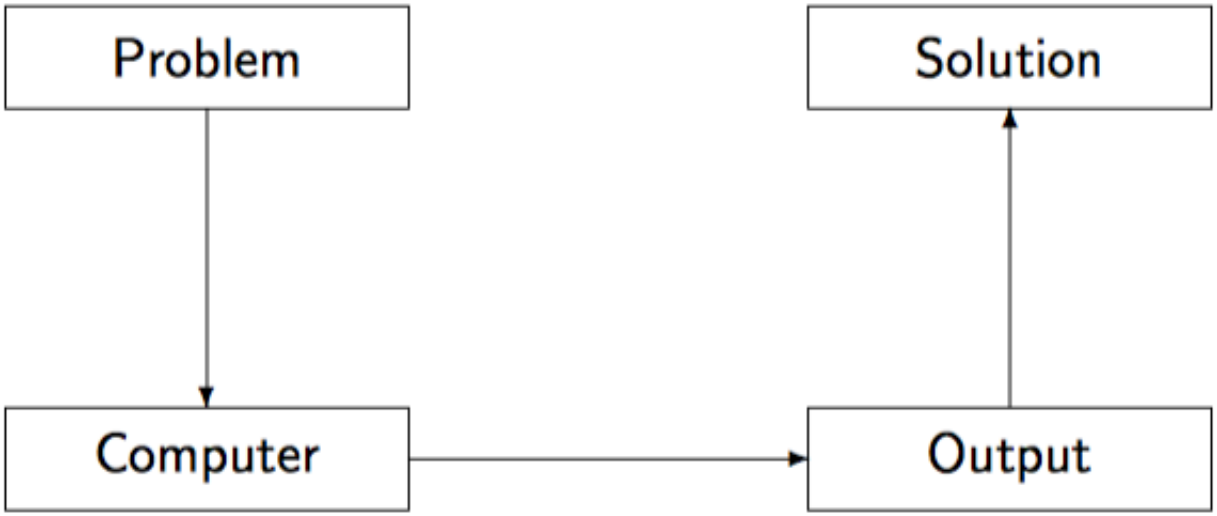
Outline

- Introduction to ASP
- Language of ASP
- Extensions of ASP by ARG@ASU
  - ASPMT
  - LPMLN
- Recent Applications by ARG@ASU
  - QA System
  - Smart Home

Intro to ASP

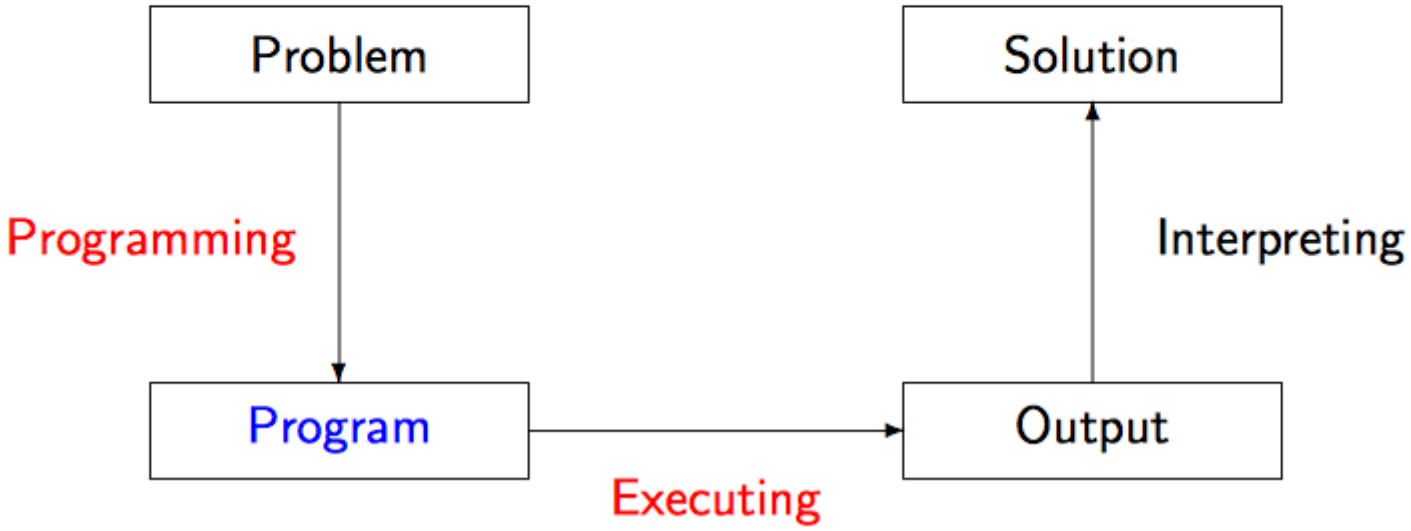
Problem Solving

"What is the problem?" versus "How to solve the problem?"



Procedural Programming

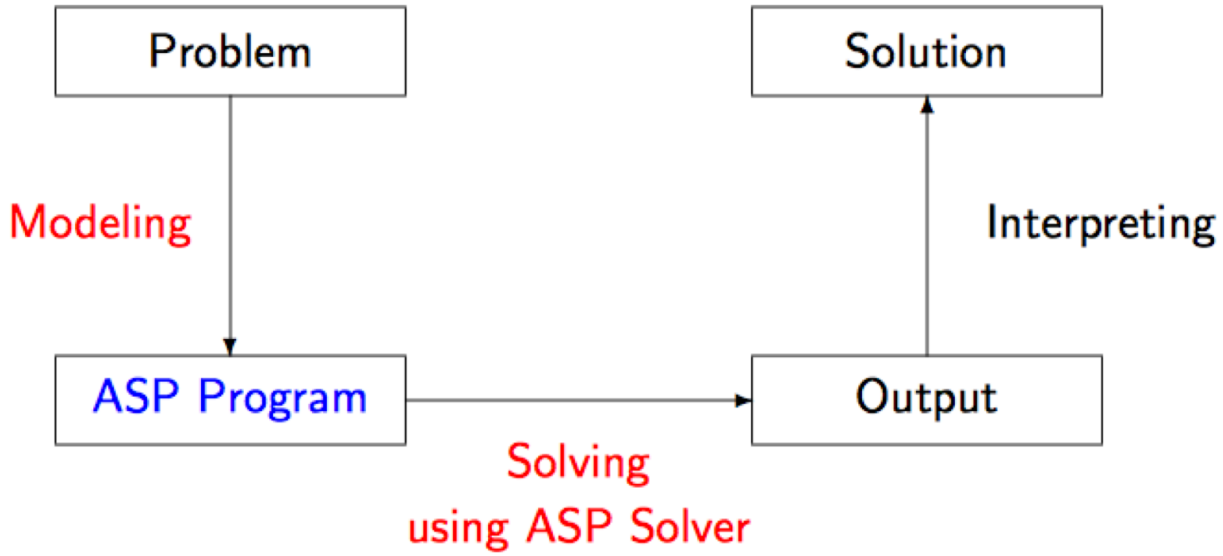
"What is the problem?" versus "How to solve the problem?"



In procedural programming, we specify the sequence of operations to be executed to generate an answer

Declarative Problem Solving using ASP

"What is the problem?" versus "How to solve the problem?"



In ASP, we specify the properties of the answers using logic rules and let ASP solver generate the answers

What is Answer Set Programming (ASP)

- Declarative programming paradigm combining
  - a rich yet simple modeling language
  - with high-performance solving capacities
- ASP is useful for knowledge-intensive tasks and combinatorial search problems
- ASP has its roots in
  - logic programming
  - knowledge representation
  - constraint solving (in particular SAT)
  - (deductive) databases

$$\text{ASP} = \text{LP} + \text{KR} + \text{SAT} + \text{DB}$$

What is Answer Set Programming (ASP)

- Many efficient ASP solvers are available: clingo, smodels, dlv, dlv-hex, cmodels, pbmodels, ASPeRIX, JASP, WASP, ME-ASP, etc.
- ASP Core 2: standard language
- Biennial ASP Solver Competition

ASP Applications

- information integration
- constraint satisfaction
- planning, routing
- robotics
- diagnosis
- security analysis
- configuration
- computer-aided verification
- biology / biomedicine
- knowledge management ...

But this is not to be too emphasized

Outline

- Introduction to ASP
- Language of ASP
- Extensions of ASP by ARG@ASU
  - ASPMT
  - LPMLN
- Recent Applications by ARG@ASU
  - QA System
  - Smart Home

Answer Sets of Traditional Logic Programs

$$A_0 \leftarrow A_1 \wedge \cdots \wedge A_m \wedge \neg A_{m+1} \wedge \cdots \wedge \neg A_n$$

- means intuitively that
  - generate A_0 if you have generated $A_1, \ldots, A_m$
  - provided that none of the atoms $A_{m+1}, \ldots, A_n$ can be generated using the rules of the program.
- Vicious circle: how do you know some atoms cannot be generated while you're generating some other atoms?
- Negation as failure

Answer Sets

Program	Answer Sets
$p \leftarrow not\ q$	$\{p\}$
$p \leftarrow not\ q$ $q \leftarrow not\ r$	$\{q\}$
$p \leftarrow not\ q$ $q \leftarrow not\ p$	$\{p\}, \{q\}$

Prolog vs. ASP

$$p \leftarrow not\ q$$
$$q \leftarrow not\ p$$

Prolog does not terminate on query p or q.

```
?- p.
ERROR: Out of local stack
   Exception: (729,178)
```

SMODELS returns

```
Answer: 1
Stable Model: p
Answer: 2
Stable Model: q
```

Finite ASP programs are guaranteed to terminate.

Answer Sets

Program	Answer Sets
$\{p, q, r\}$	$\emptyset, \{p\}, \{q\}, \{r\}, \{p, q\}, \{q, r\}, \{p, r\}, \{p, q, r\}$
$1\{p, q, r\}2$	$\{p\}, \{q\}, \{r\}, \{p, q\}, \{q, r\}, \{p, r\}$
$1\{p, q, r\}2$ $\leftarrow p$	$\{q\}, \{r\}, \quad \{q, r\}$

Traveling Salesman

```
% each node has exactly one outgoing edge and one incoming edge
{in(X,Y): edge(X,Y)} = 1 :- node(X).
{in(X,Y): edge(X,Y)} = 1 :- node(Y).

% every node is reachable from some fixed vertex 1.
reached(Y) :- in(1,Y).
reached(Y) :- in(X,Y), reached(X).
:- node(Y), not reached(Y).

% minimize the cost of travel
#minimize { C,X,Y : in(X,Y), cost(X,Y,C) }.
```

Reviewer Assignment

```
reviewer(r1). paper(p1). classA(r1,p1). classB(r1,p2). coi(r1,p3). reviewer(r2).
paper(p2). classA(r1,p3). classB(r1,p4). coi(r1,p6).
[...]

% each paper is assigned to 3 reviewers
{assigned(P,R): reviewer(R)} = 3 :- paper(P).

% do not assign coi and no expertise papers
:- assigned(P,R), coi(R,P).
:- assigned(P,R), not classA(R,P), not classB(R,P).
% assign 6 to 9 papers to each reviewer
:- not 6{assigned(P,R) : paper(P)}9, reviewer(R).

% do not assign >=3 classB papers
assignedB(P,R) :- classB(R,P), assigned(P,R).
:- 3{assignedB(P,R): paper(P)}, reviewer(R).

% minimize the number of classB assignment
#minimize { 1,P,R : assignedB(P,R), paper(P), reviewer(R) }.
```

Outline

- Introduction to ASP
- Language of ASP
- Extensions of ASP by ARG@ASU
  - ASPMT
  - LPMLN
- Recent Applications by ARG@ASU
  - QA System
  - Smart Home

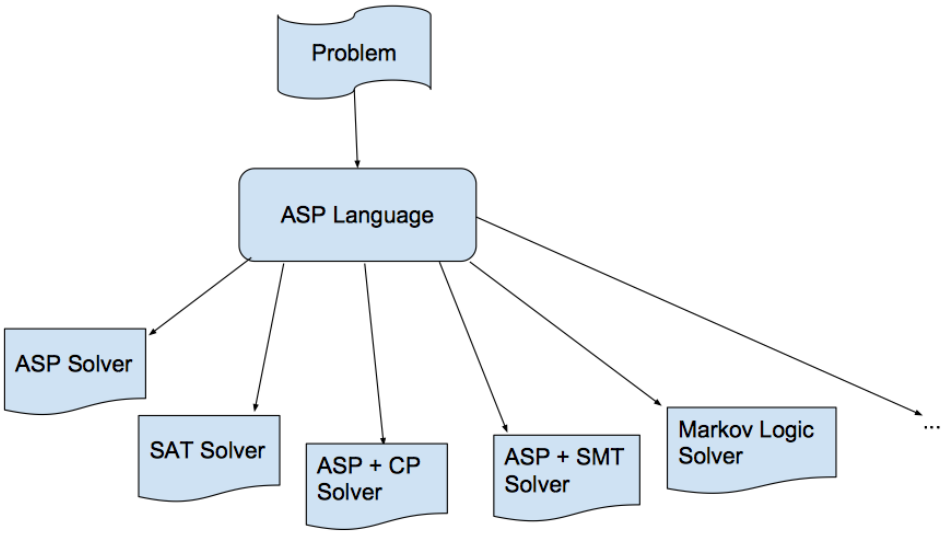
Extensions of ASP

ASPMT (extension to first-order)

LPMLN (extension to probabilistic reasoning)

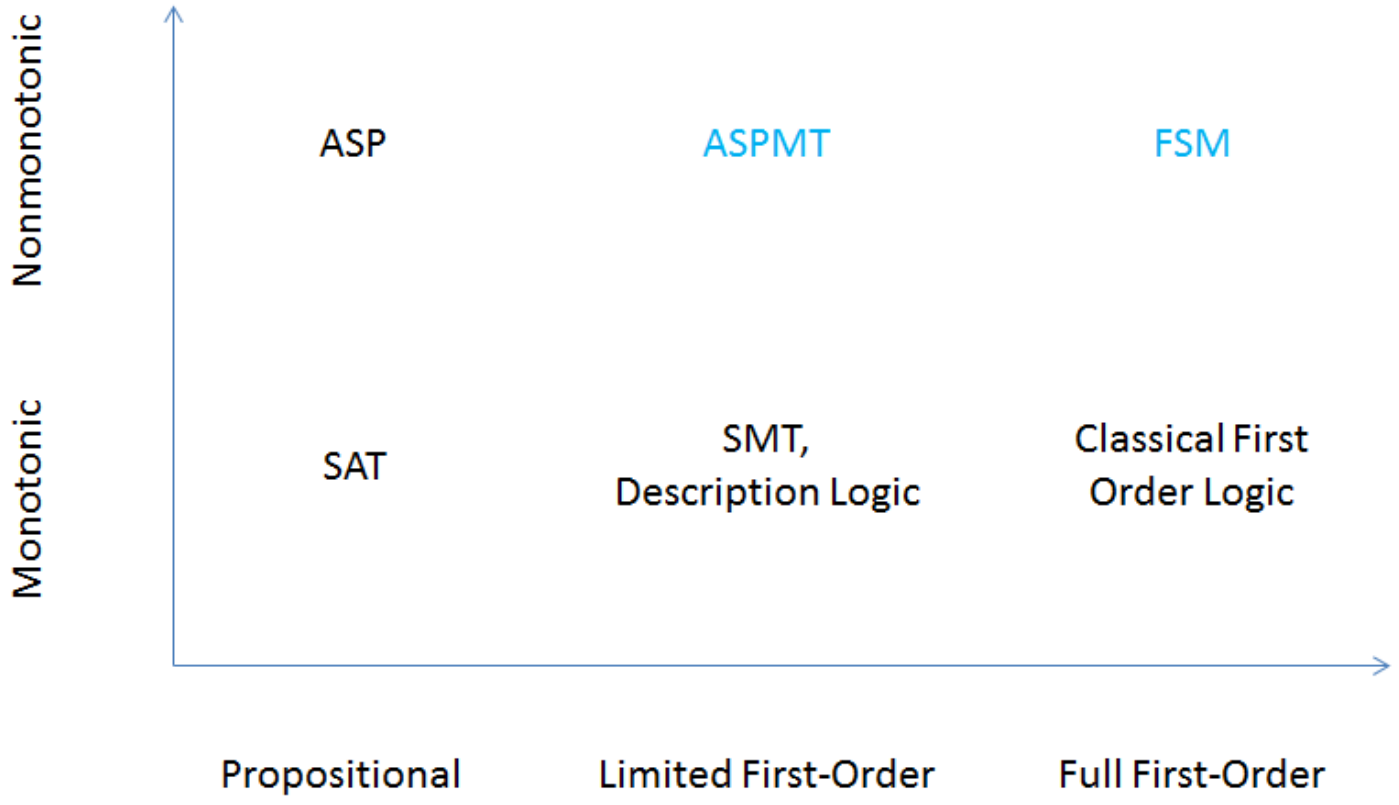
ASP as Interface Language for Declarative Paradigms

- ASP language serves as a specification language for AI.
- Computation is carried out by compilation to different engines



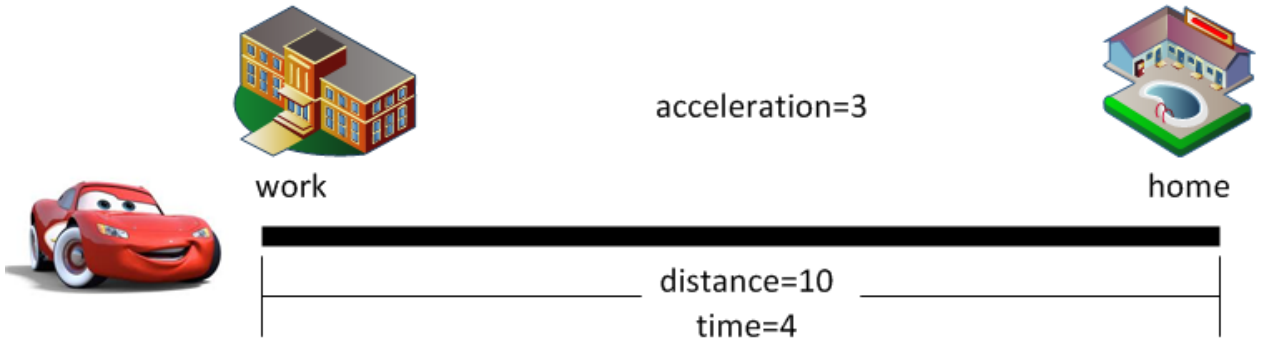
- Annual workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP) since 2008

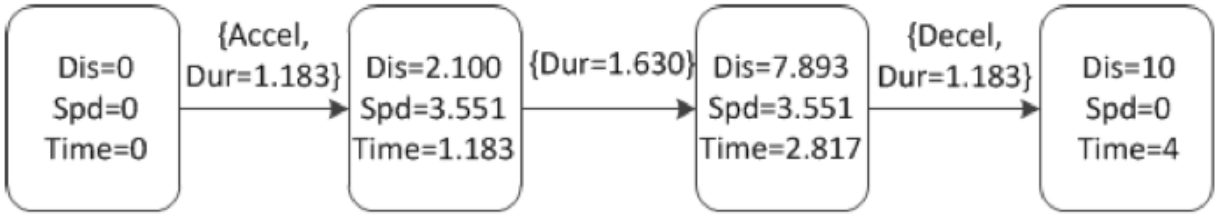
Answer Set Programming Modulo Theories [Bartholomew, L_ & Meng]



Planning with Continuous Time

- Give a formal representation of the domain to generate a plan





Car Example in ASPMT

Intensional constants:		Domains:
$i:Speed$, $i:Distance$	$(0 \leq i \leq maxstep)$	$\mathcal{R}_{\geq 0}$

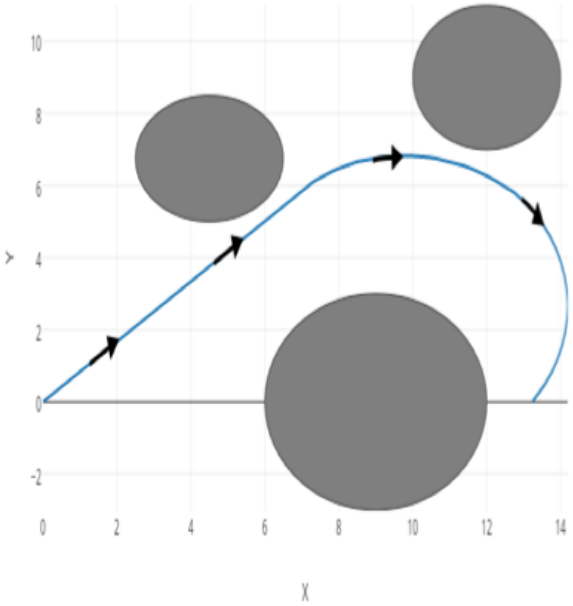
Nonintensional constants:		
$i:Time$	$(0 \leq i \leq maxstep)$	$\mathcal{R}_{\geq 0}$
$i:Accelerate$, $i:Decelerate$	$(0 \leq i < maxstep)$	Boolean
$i:Dur$	$(0 \leq i < maxstep)$	$\mathcal{R}_{\geq 0}$

Axioms:

$$i{+}1:Speed = v + \mathrm{A} \times t \leftarrow i:(Accelerate \wedge Speed = v \wedge Dur = t)$$
$$i{+}1:Speed = v - \mathrm{A} \times t \leftarrow i:(Decelerate \wedge Speed = v \wedge Dur = t)$$
$$\{i{+}1:Speed = v\} \leftarrow i:Speed = v$$
$$i{+}1:Distance = d + 0.5 \times (v + v') \times t \leftarrow i{+}1:Speed = v' \wedge i:(Distance = d \wedge Speed = v \wedge Dur = t)$$

ASP Modulo ODE [L_& Loney, 2017]

Can encode hybrid automata succinctly



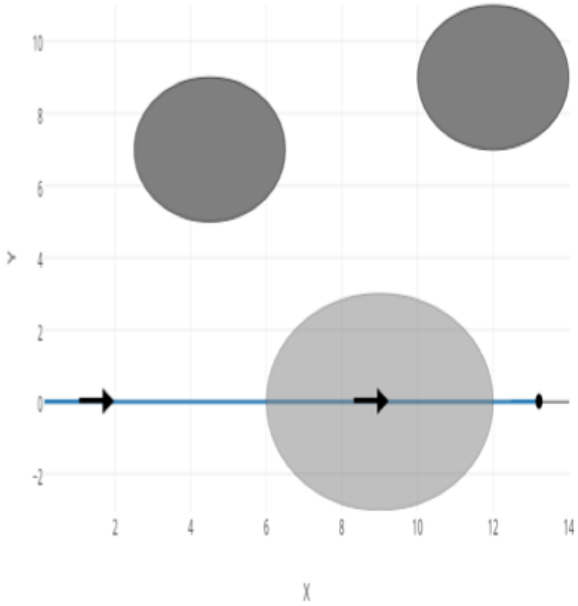
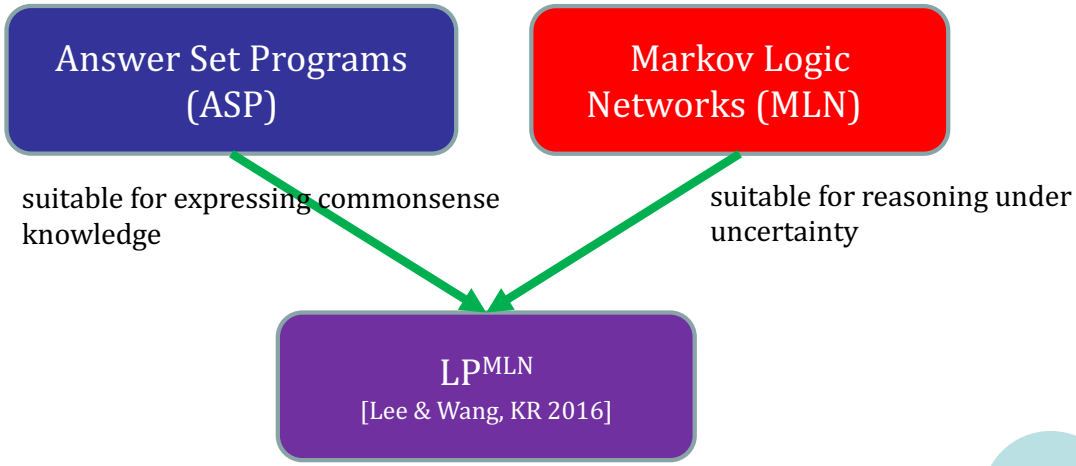
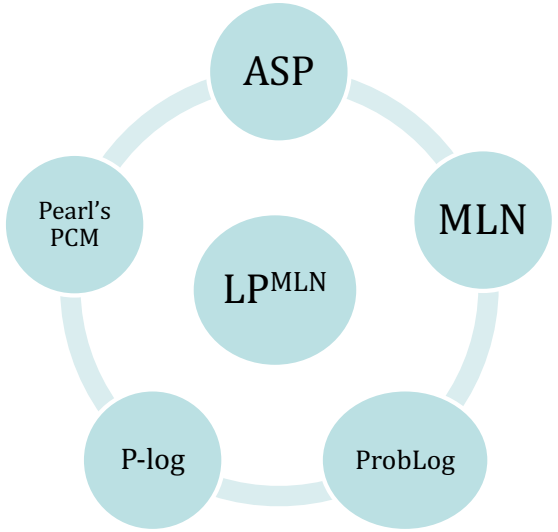


Fig. 1. (a) feasible plan (b) infeasible plan

LPMLN

Language LP^{MLN} [L_ & Wang]

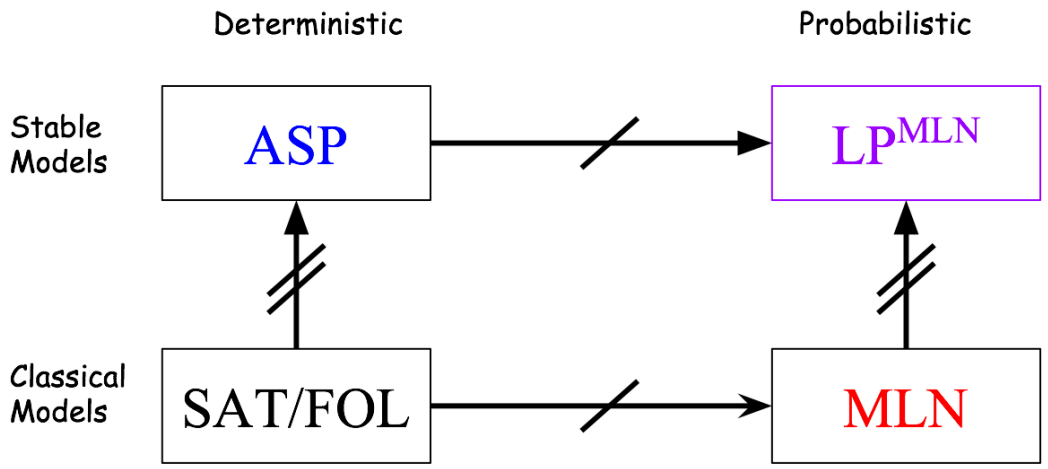




Relationship between LP^{MLN} and several other formalisms were established:
[L_ & Wang, 2016; L_, Meng & Wang 2015]

Language LP^{MLN}

- Overcomes the weakness of ASP in handling uncertainty.
- Overcomes the weakness of MLN in handling expressive commonsense reasoning.



LP^{MLN}

- Syntactically, it's a simple extension of ASP where each rule is prepended by weights
  - infinite weight (∞) tells the rule is a definite knowledge
- Each answer set gets weights from the rules that are true in the answer set.
  - an answer set does not have to satisfy all rules
  - the more rules are true, the more likely the answer set is
- Adopting the log-linear models of MLN, language LP^{MLN} provides a simple and intuitive way to incorporate the concept of weights into the stable model semantics
  - It overcomes some semantic weakness of MLN.
  - While MLN is an undirected approach, LP^{MLN} is a directed approach, where the directionality comes from the stable model semantics
- Probabilistic answer set computation can be reduced to sampling and optimization problems studied in machine learning

KB_1

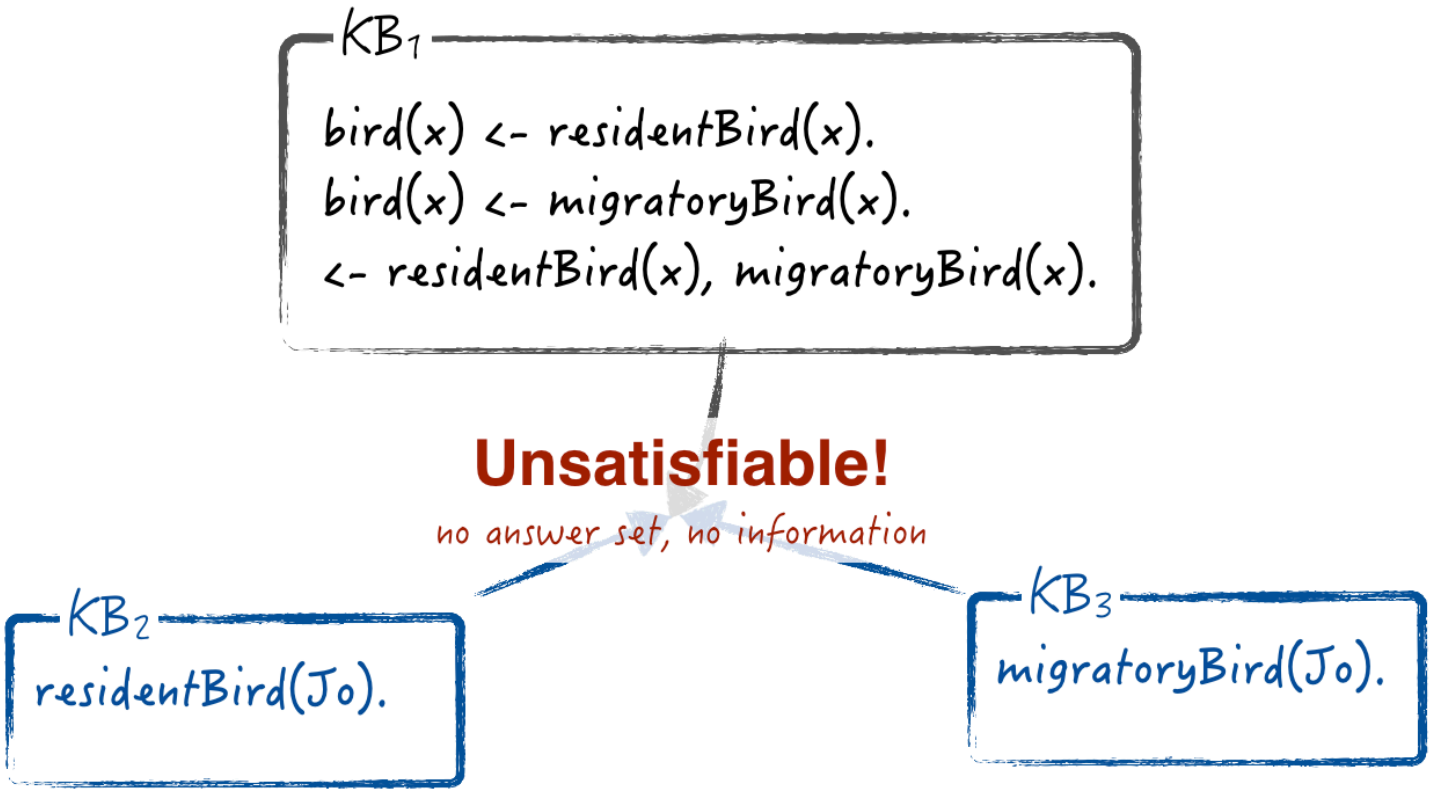
```
bird(x) <- residentBird(x).
bird(x) <- migratoryBird(x).
<- residentBird(x), migratoryBird(x).
```

KB_2

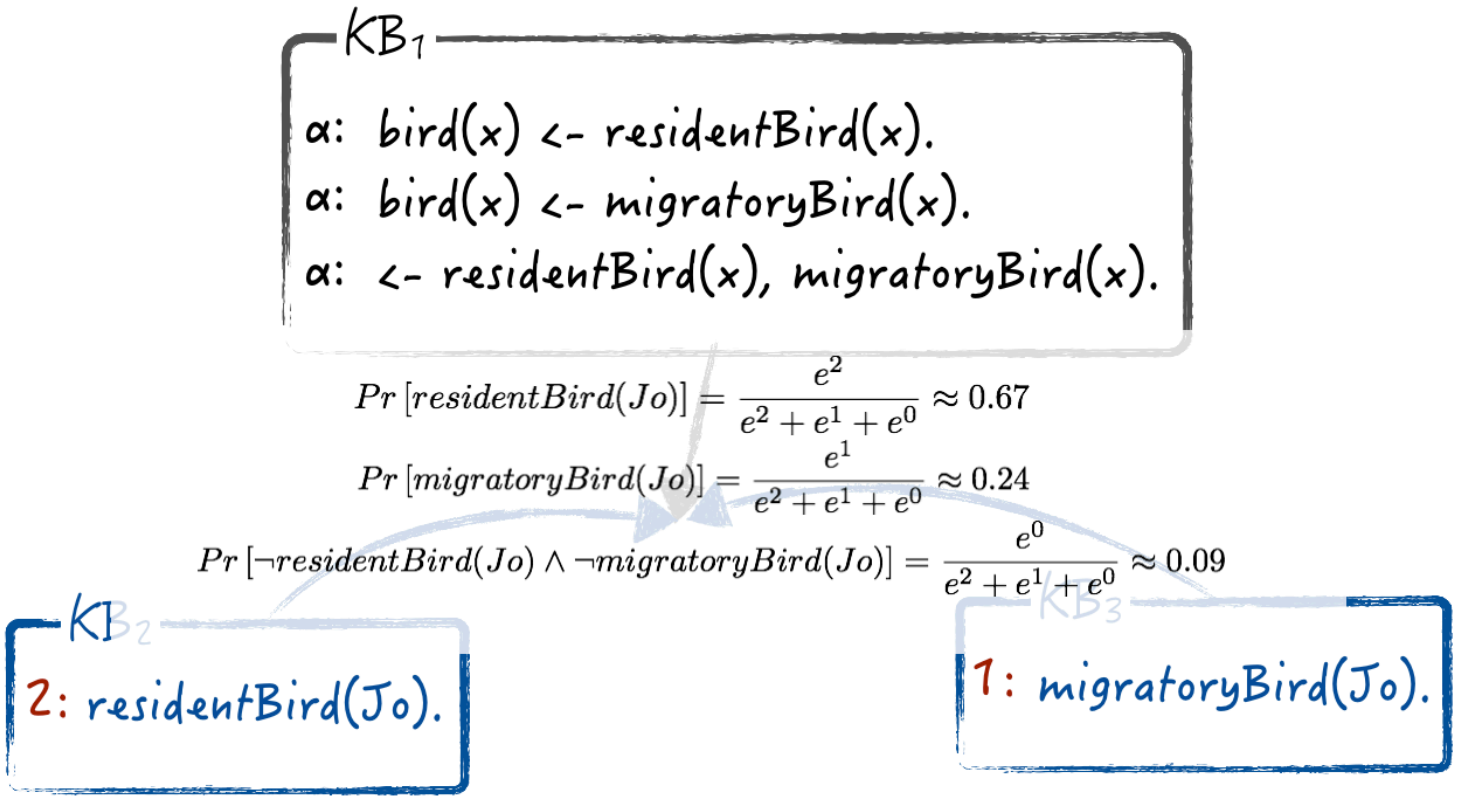
```
residentBird(Jo).
```

KB_3

```
migratoryBird(Jo).
```



But still useful to conclude that Jo is a bird!


KB1
α: bird(x) <- residentBird(x).
α: bird(x) <- migratoryBird(x).
α: <- residentBird(x), migratoryBird(x).
$Pr\,[residentBird(Jo)] = \frac{e^2}{e^2+e^1+e^0} \approx 0.67$
$Pr\,[migratoryBird(Jo)] = \frac{e^1}{e^2+e^1+e^0} \approx 0.24$
$Pr\,[\neg residentBird(Jo) \wedge \neg migratoryBird(Jo)] = \frac{e^0}{e^2+e^1+e^0} \approx 0.09$
KB2
2: residentBird(Jo).
KB3
1: migratoryBird(Jo).

Probabilistic ASP

- Network connectivity: "what is the probability that there exists a path between the nodes?"

  0.7: edge(0,0),
  0.3: edge(2,3), …,

  ∞: path(X, Y) :- edge(X, Y).
  ∞: path(X, Y) :- path(X, Z), path(Z, Y), Y != Z.

- Probabilistic Traveling Salesman: "Given a graph with uncertain edges, what is the probability that there is a Hamiltonian circuit? "

Probabilistic ASP

Consider a variant in which the wolf does not always eat the sheep.

$$
\begin{aligned}
\alpha: &\quad Loc_i(o,l) \leftarrow OnBoat_i(o), LocBoat_i(l)\\
\alpha: &\quad \{Loc_{i+1}(o,l)\}^{\mathrm{ch}} \leftarrow Loc_i(o,l)\\
\alpha: &\quad OnBoat_{i+1}(o) \leftarrow GetOnBoat_i(o)\\
\alpha: &\quad LocBoat_{i+1}(l) \leftarrow MoveBoat_i(l)\\
w: &\quad SheepEaten \leftarrow Loc_i(Wolf,l), Loc_i(Sheep,l), not\ LocBoat_i(l)\\
\alpha: &\quad Success \leftarrow Loc_{\max}(Wolf,L_2), Loc_{\max}(Sheep,L_2),\\
&\qquad\qquad Loc_{\max}(Cabbage,L_2), not\ SheepEaten\\
\alpha: &\quad \leftarrow not\ Success.
\end{aligned}
$$

The minimal length plan for the original puzzle involves 17 actions of loading, moving and unloading,

The elaboration has a new minimal length plan involving 11 actions only with some chance of failure.

LP^{MLN} System [Talsania, Wang & L_]

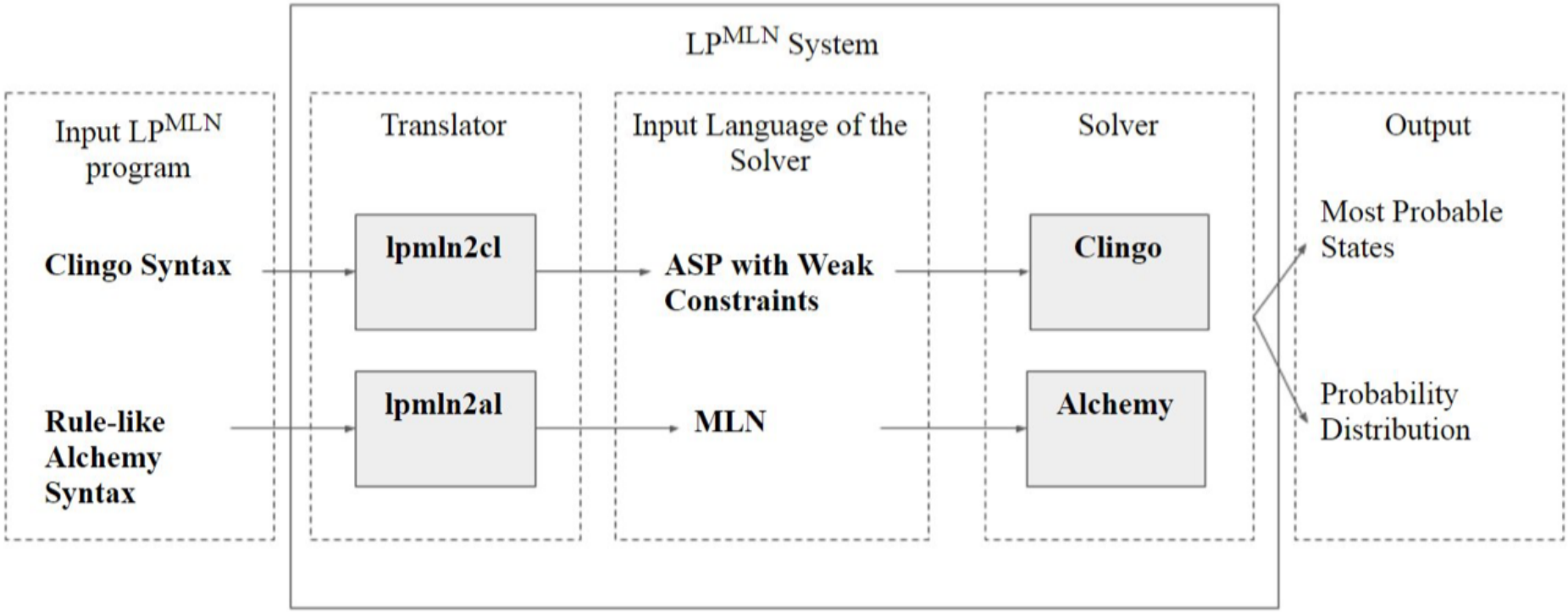


Fig. 1. LP^{MLN} System Architecture

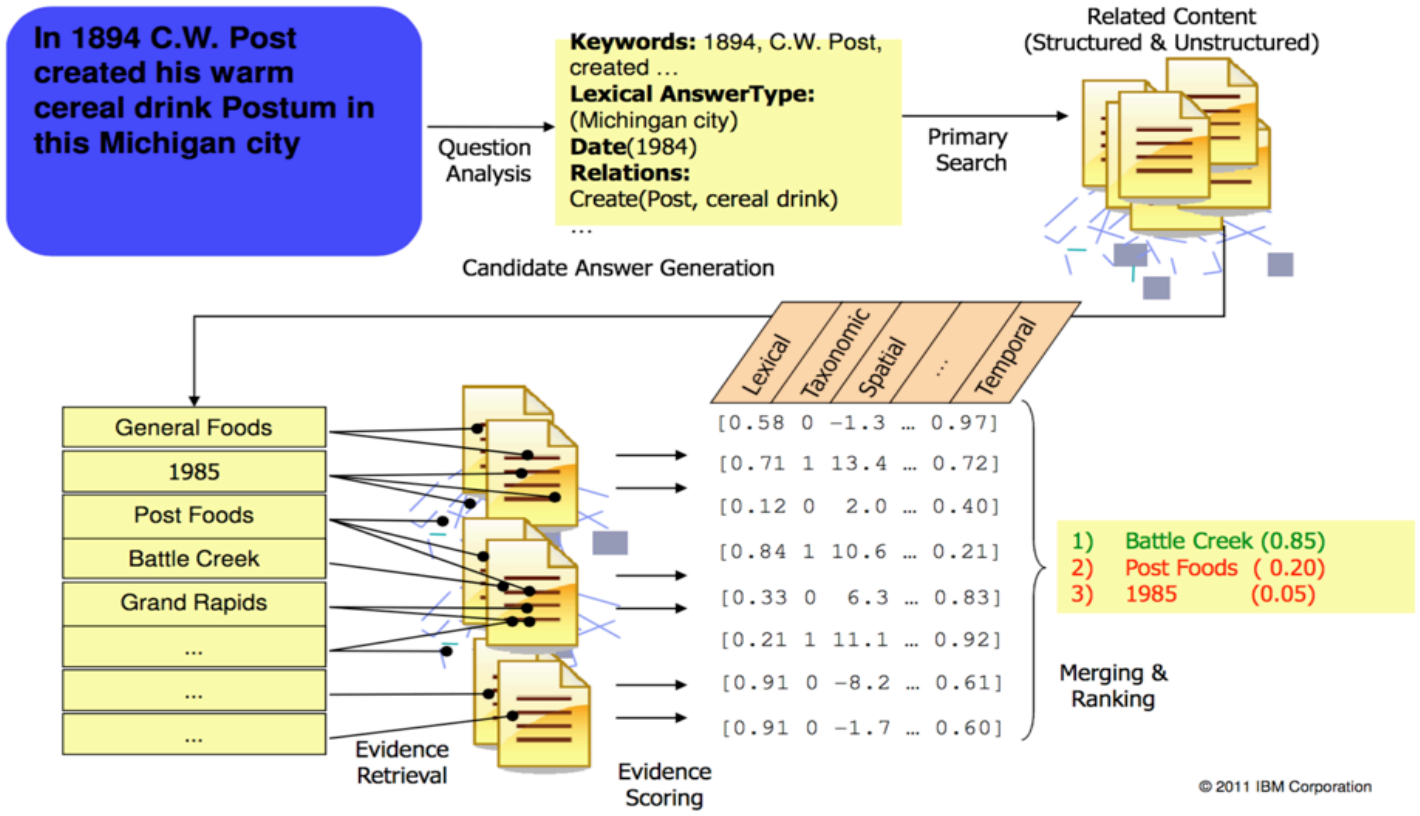
Outline

- Introduction to ASP
- Language of ASP
- Extensions of ASP by ARG@ASU
  - ASPMT
  - LPMLN
- Recent Applications by ARG@ASU
  - QA System
  - Smart Home

Knowledge-Based QA System

IBM WATSON

Example Question

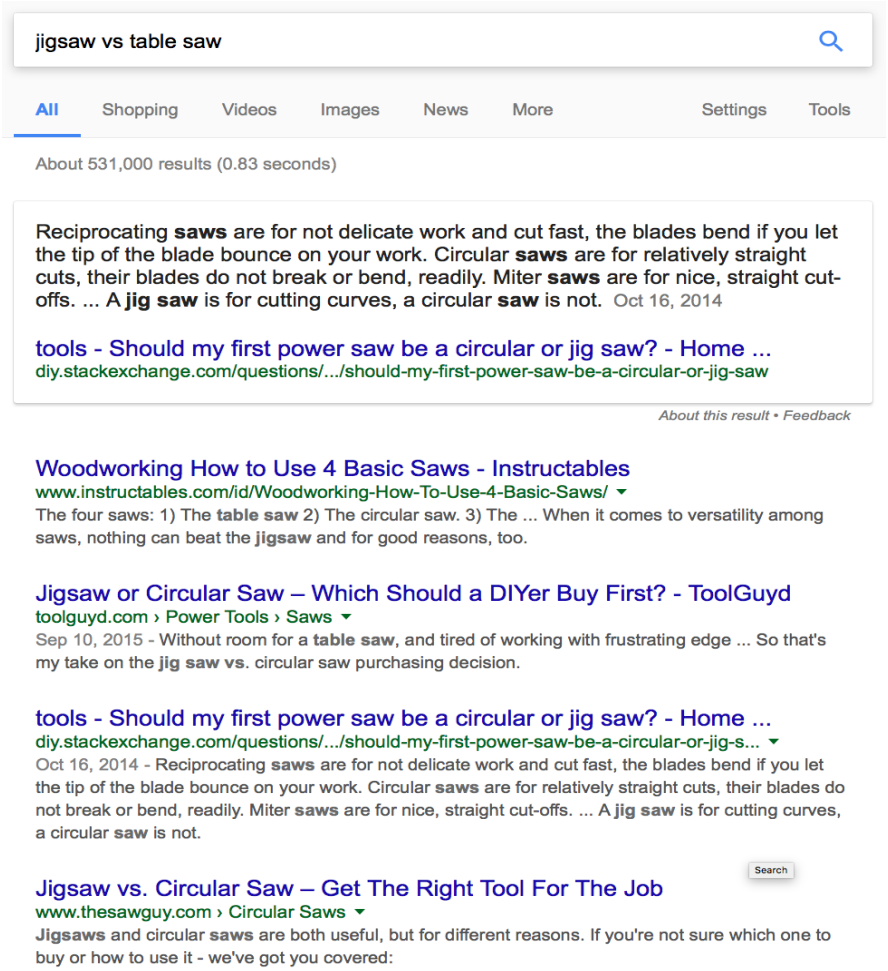


(Slides borrowed from James Fan)

QA System for DIY [Wang, L_ & Kim]

- Many of the questions cannot be answered by simple database/knowledge base search
  - Answering requires complex reasoning, context-awareness (DIY project, user's skill level, time constraints, etc.), as well as ability to explain

Using Google



Band Saw vs Table Saw – Which Comes First? | - Toolerant
www.toolerant.com/band-saw-vs-table-saw-which-comes-first/
Apr 7, 2013 - Working with a hand-held **jigsaw** or circular saw can satisfy your basic needs, but .... A band saw is loud, but a **table saw** is much louder.

table saw or jigsaw - Woodworking Talk - Woodworkers Forum
www.woodworkingtalk.com/f12/table-saw-jigsaw-5045/
Jul 10, 2008 - I have been looking for a **table saw** but really cant afford one. The other option is a **jigsaw** maybe. Have thought of using a **jigsaw** and then ...

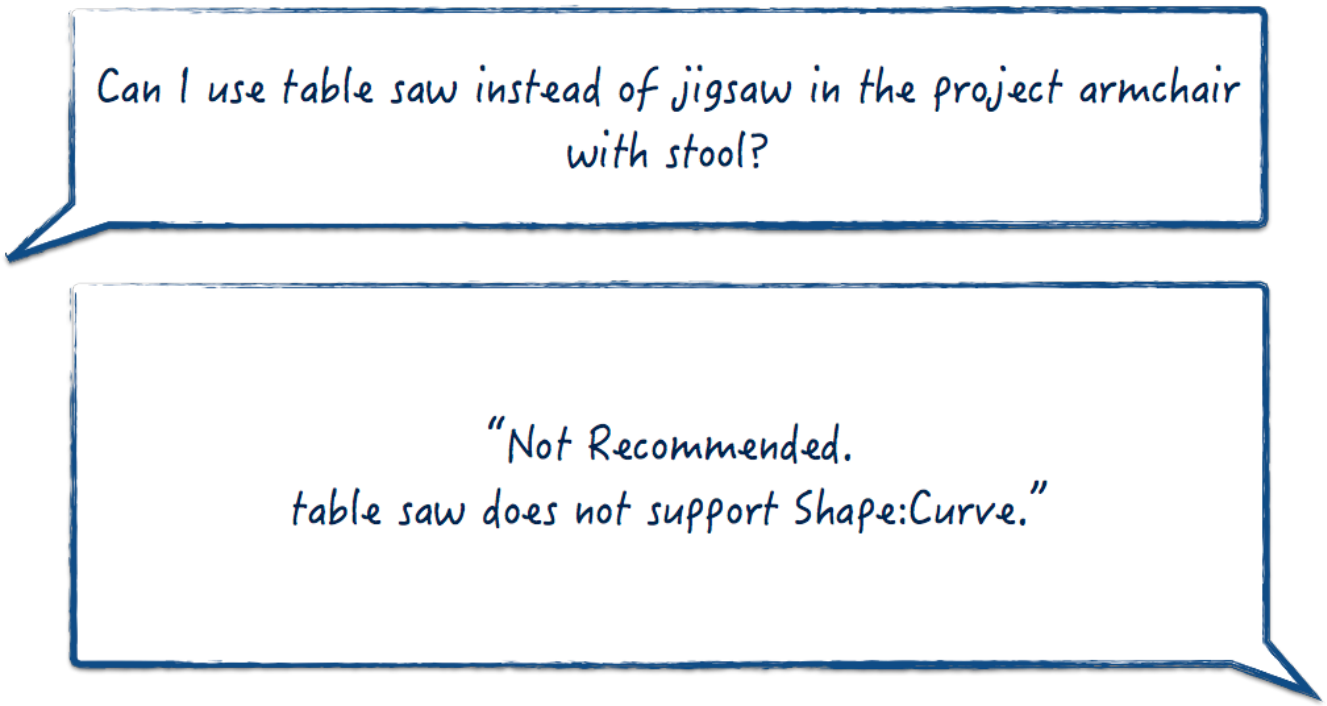
If You Can Only Buy One Saw: Circular or Jigsaw? - The Garage ...
www.garagejournal.com › ... › The Tools › General Tool Discussion
Jan 16, 2013 - 20 posts - 17 authors
Building a small project like a train **table** well the **jig saw** will make all ... do not like cordless **saws** as I feel the cost **vs** performance **vs** battery life ...

Bandsaw vs Jig Saw or Scroll Saw ??? - The Garage Journal Board
www.garagejournal.com › ... › The Tools › General Tool Discussion
Dec 4, 2012 - 9 posts - 8 authors
Jig saw can be used on wood, plastic, metal, depending on blade. ... be used to resaw thicker pieces of wood much safer than on a **table saw**.

Life Without A Tablesaw? - The Wood Whisperer
www.thewoodwhisperer.com/articles/life-without-a-tablesaw/
Mar 17, 2012 - Whether you use a miter gauge or a cross-cut sled, the **tablesaw** is incredibly ... If you don't mind a rougher cut, you could also use a **jigsaw**. ..... My biggest concern; like Marc's is over kickback **vs** a cut, and as such I have ...

Best Power Saw for a DIY Beginner? | Apartment Therapy
www.apartmenttherapy.com/best-power-saw-111072
Mar 11, 2010 - I want something that's easy to use and versatile. Is there such thing as an "all- purpose power **saw**"? Should I buy a circular **saw** or a **jigsaw**?

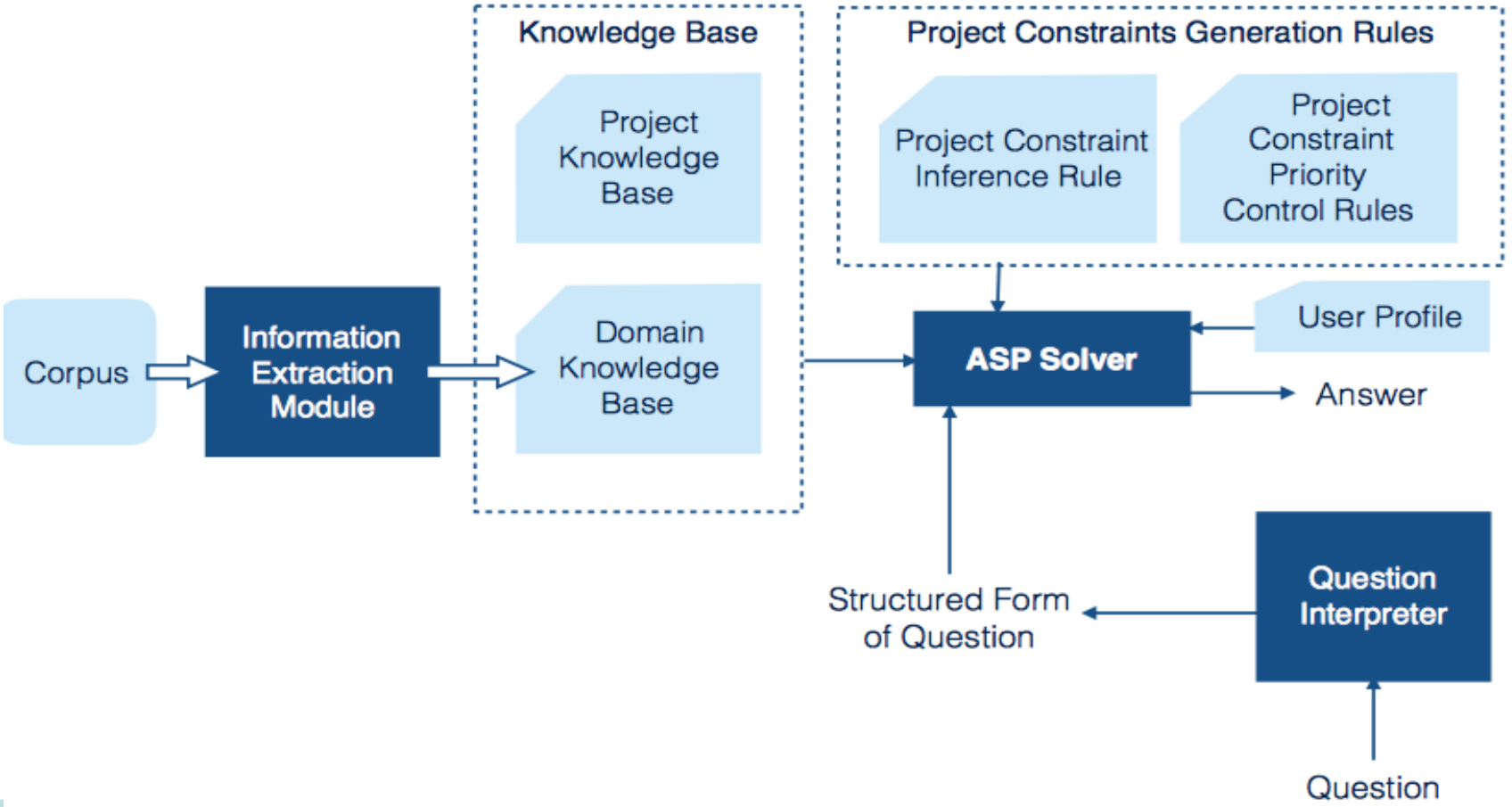
Our System


Can I use table saw instead of jigsaw in the project armchair with stool?
"Not Recommended.
table saw does not support Shape:Curve."

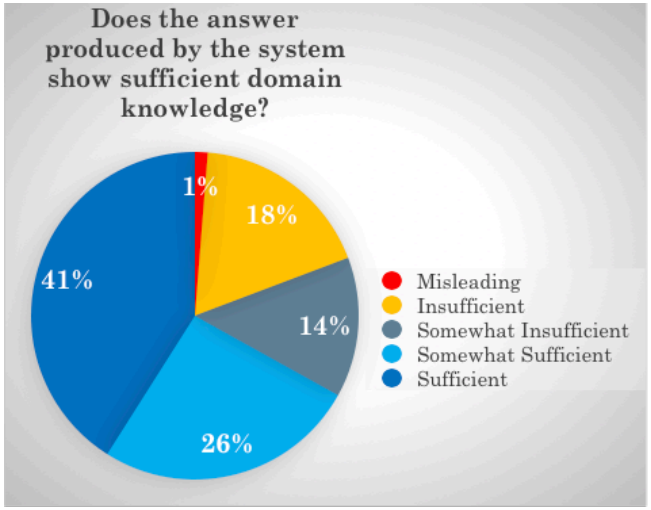
ASP Based Approach

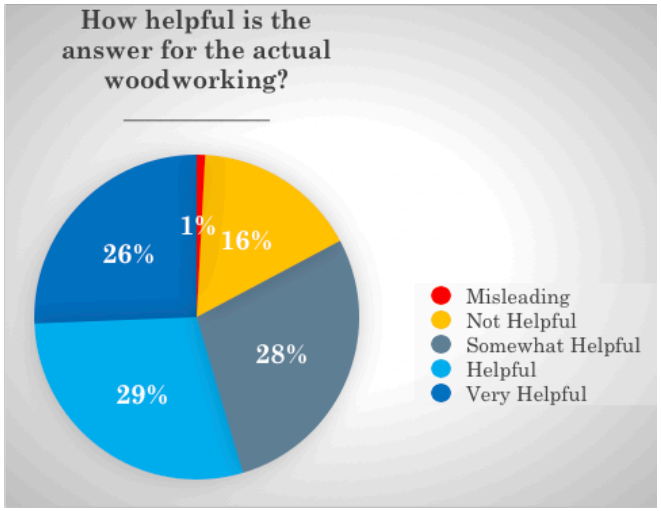
- Find constraints from the context that the replacement would violate
  - properties of objects,
  - kind of DIY project
  - user's age, skill level, etc.
  - time limit, budget, etc.
- Conclude "Not Recommended" if such constraint can be found
- Conclude "possible" otherwise

System Architecture



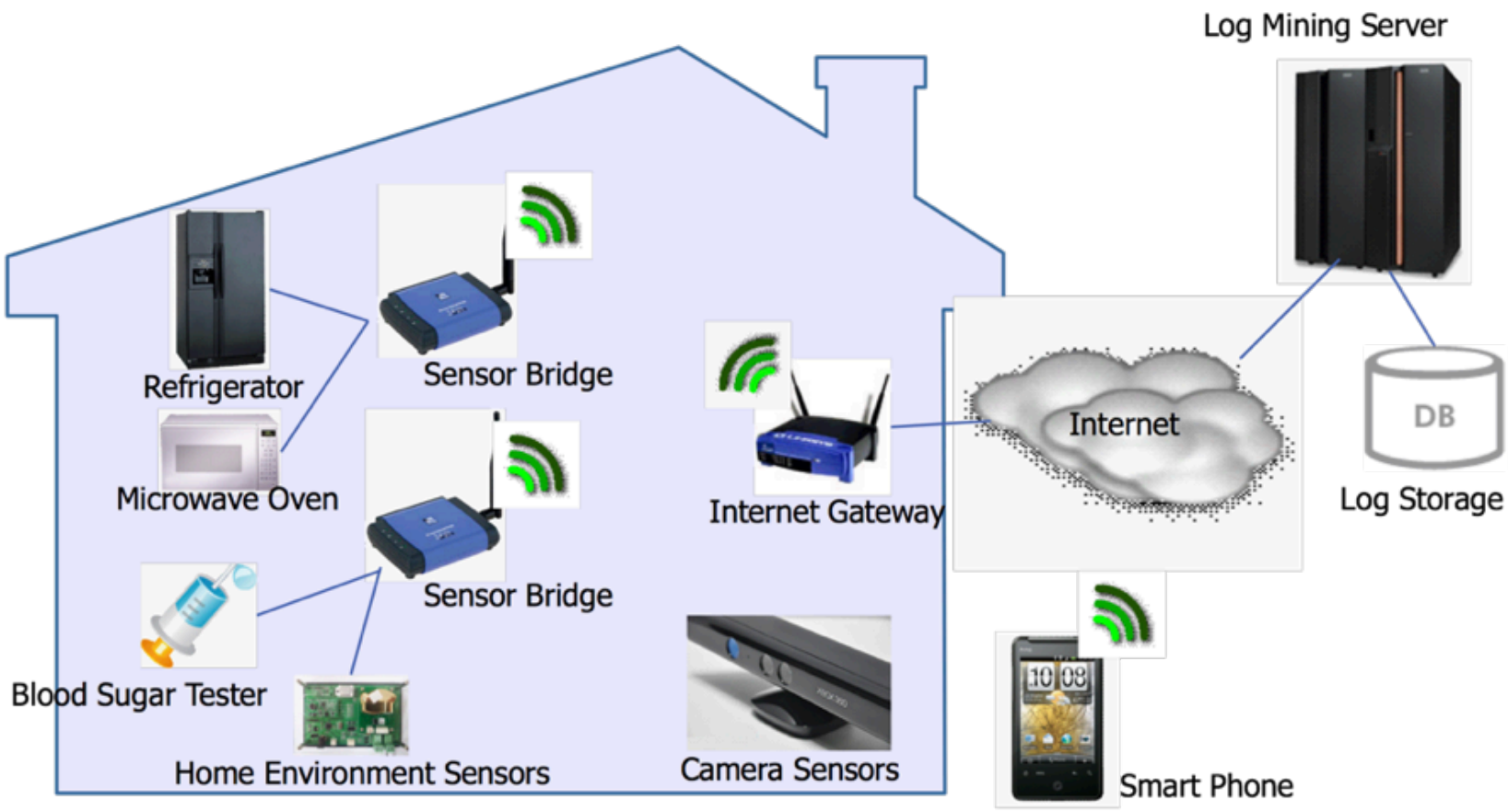
- Our result shows that our system provides a much better user experience.
- Using Google, 40% of the participants failed to find an answer within three minutes and that the average number of the websites visited by the participants was three.
- This indicates that answering a non-factoid DIY question using online search requires quite much time and efforts.





IoT Reasoning

Number of Connected Device Will Increase

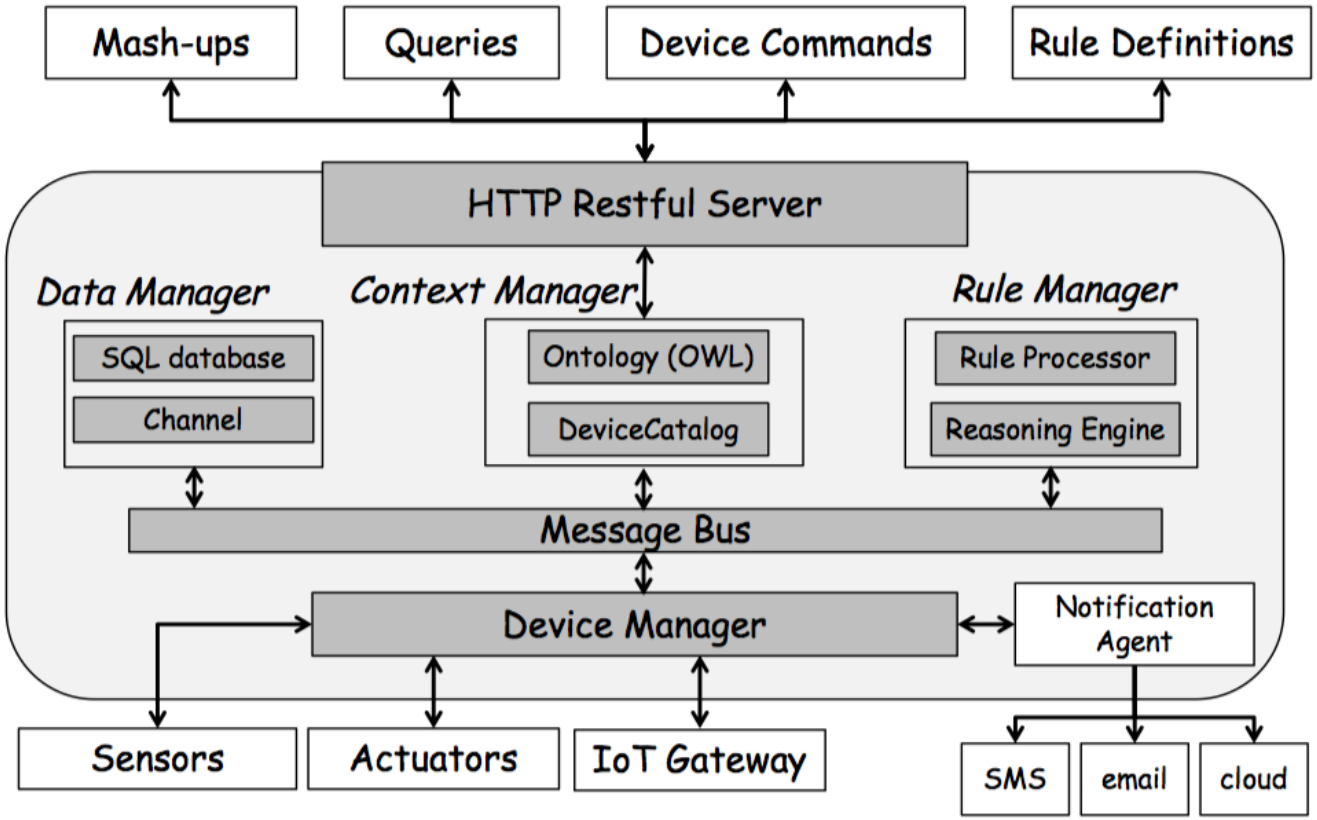


Smart Sensors/Devices Everywhere

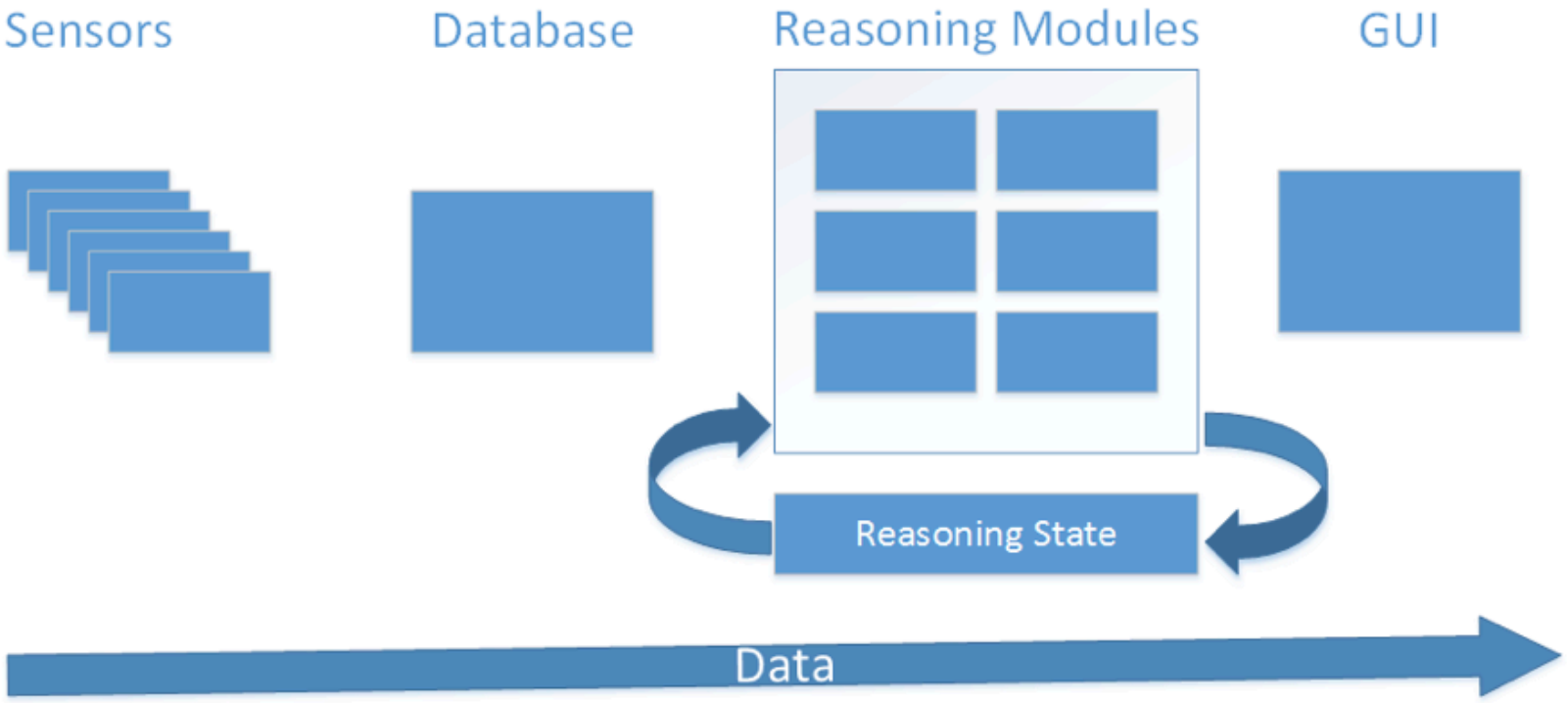
- But does it provide smart service?
- Where does it break?
  - Context missing
  - Integration of knowledge missing
  - No planning / scheduling

x

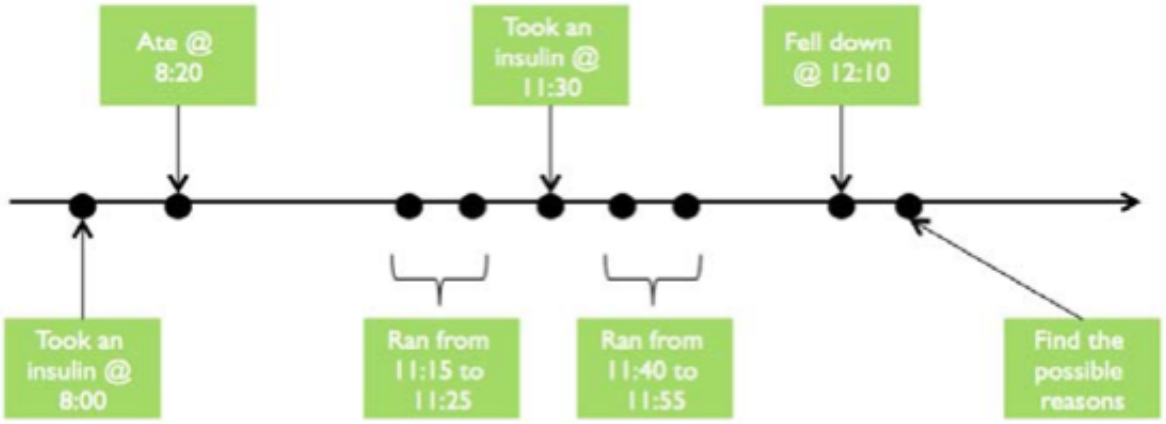
KRIPS (Knowledge-Rich Networked Intelligence for Personal Assistants [Gardell, Yang, YH Lee & L_]



Flow


Sensors
Database
Reasoning Modules
GUI
Reasoning State
Data

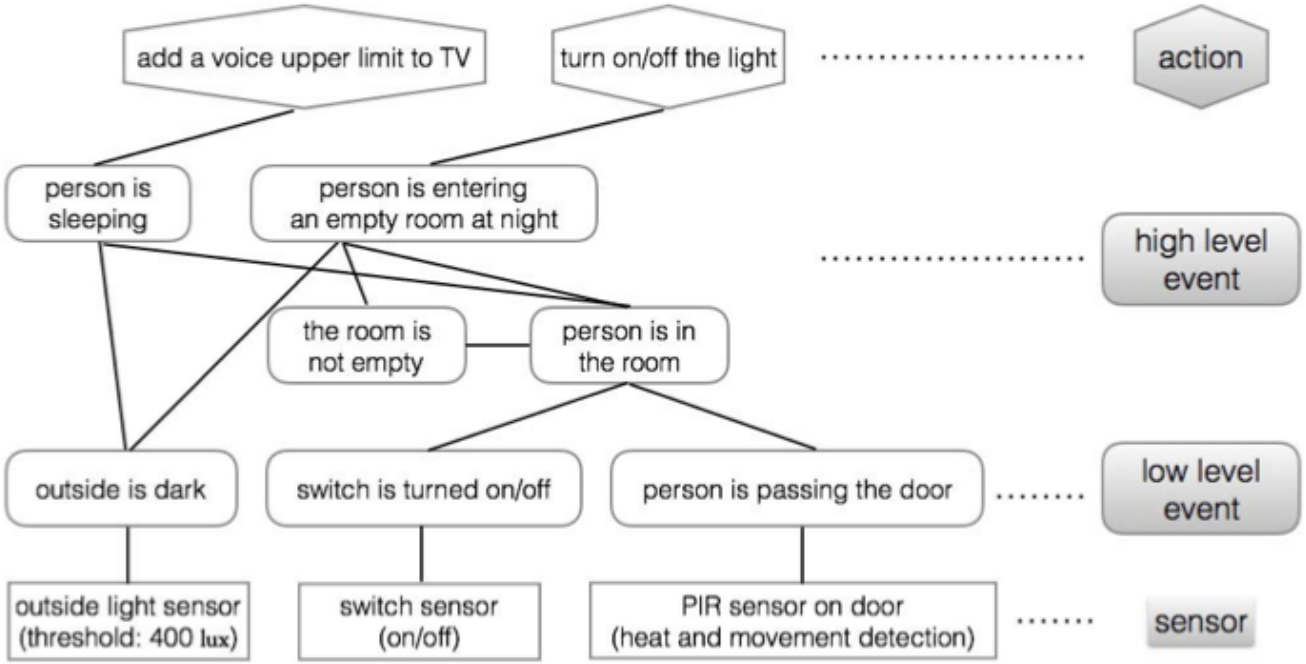
Context-Awareness



w_2 : $Ate\_after\_last\_insulin\_intake(t) \leftarrow$
$Most\_recent\_action\_record(Took\_insulin, Time(H, M, S), t),$
$Most\_recent\_action\_record(Ate, Time(h1, m1, s1), t), @diff(h1, m1, h, m) > 0.$

w_3 : $Possible\_reason\_of\_falling(Insulin\_shock, t) \leftarrow$
$Movement(Fall, Time(h, m, s), t), Low\_glucose\_level(t),$
$not\ Ate\_after\_last\_insulin\_intake(t).$

High Level Event Recognition



Personal Assistant

- Scheduling system, that creates a weekly schedule for the user, adaptive to user's preferences.
- The scheduling system uses knowledge about the user, their schedule, external information like knowledge on weather, domain specific knowledge required for a particular event etc., to provide a schedule that is smarter and more satisfactory to users.
- Unlike most commercially available digital assistants, that simply follow user commands, requires users to make complicated decision and elaborate the plan to the machine, this personal assistant performs optimized decision making to formulate ideal schedules while satisfying as many user preferences as possible without requiring the user to make actual decision of when to schedule an event.

Summary

- ASP is a declarative programming paradigm combining
  - a rich yet simple modeling language
  - with high-performance solving capacities
- ASP is useful for knowledge-intensive tasks and combinatorial search problems
- ASP can handle various types of reasoning: Boolean decision, nonmonotonic reasoning, reasoning about hybrid transitions, probabilistic reasoning
- New application domains are actively pursued