Overview of Industrial Robot Group Activities

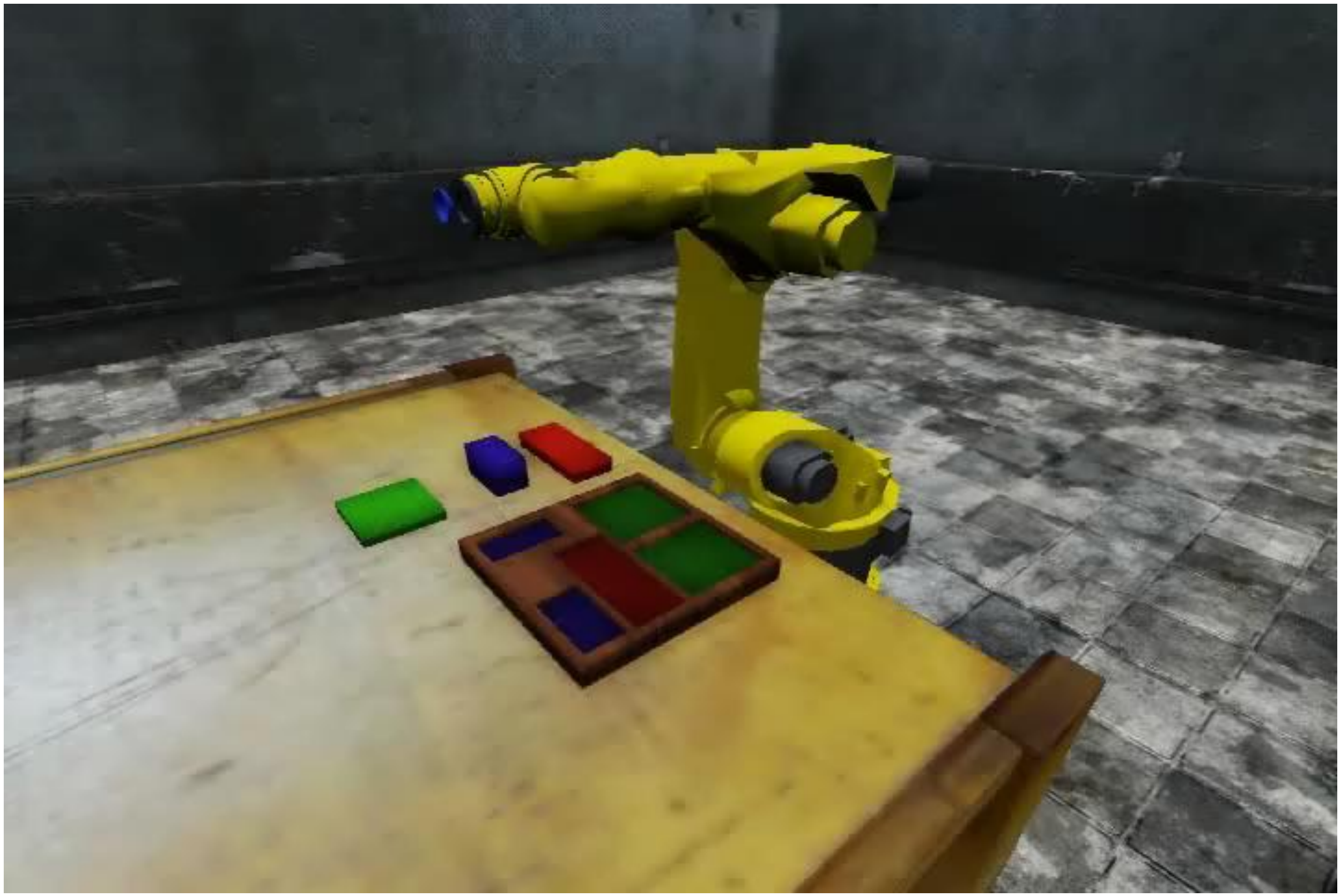
Dr. Stephen Balakirsky
National Institute of Standards and Technology
stephen@nist.gov

Dr. Thomas Kramer
Catholic University

Kit Building

- Process in which individually separate but related items are grouped, packaged, and supplied together as one unit
- Utilized in many manufacturing assembly lines
- Robotic solution requires:
  - Flexibility –Many parts with varying characteristics, part-to-part inconsistences
  - Agility – Changes in part supply locations, lack of fixturing
  - Rapid re-tasking – Ability to quickly build new varieties of kits

Real Planning / Simulated Robot

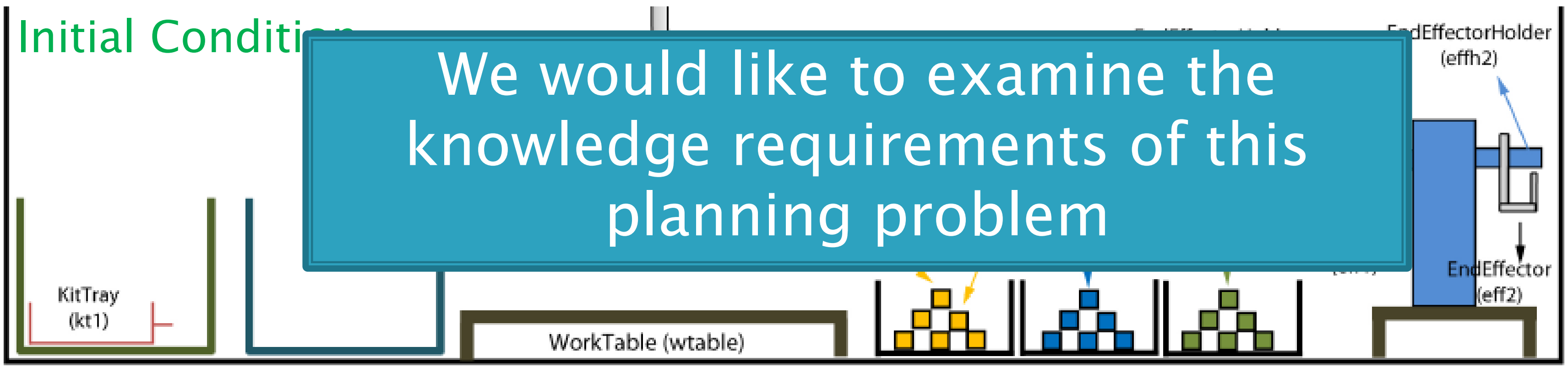


Video is 2x real time

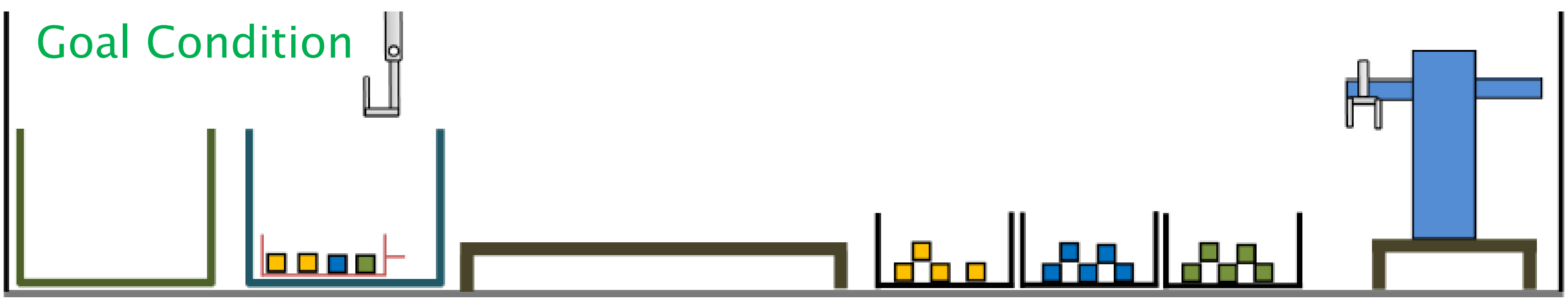
Planning Problem

$$\mathcal{P} = (\Sigma, s_0, g)$$

- Σ – State transition system (possible actions)
- S_0 – Initial conditions (objects and relationships)
- g – Goal conditions



We would like to examine the knowledge requirements of this planning problem



Underlying Knowledge Representation

- Objects
  - What objects and attributes are relevant
  - Taxonomy of objects
  - Relationships between objects
- Actions
  - What actions and attributes are relevant
  - Necessary conditions for an action to occur
  - Likely results of the action
- Dynamic world
  - Object attributes change over time
  - Actions do not always accomplish what we want them to

Objects, Relationships, and Actions

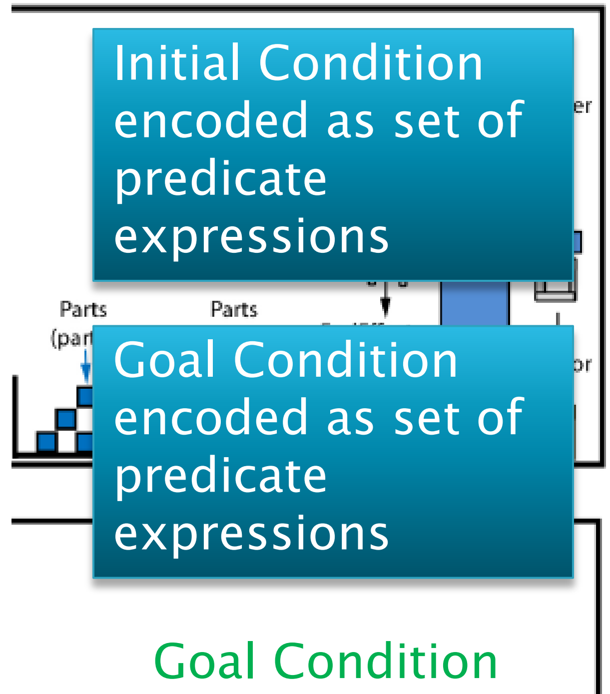
- State-Variable Representation[1] (SVR) used to formally define environment's objects, object relations (state), and actions
- Each state is represented by a tuple of values of *n* state variables $\{x_1, \ldots, x_n\}$
- Action is represented by a partial function that maps this tuple into some other tuple of values of the *n* state variables
- SVR is not a standard interchange language
  - Can encode the SVR into Planning Domain Definition Language (PDDL)

[1]Dana Nau, Malik Ghallab, and Paolo Traverso. 2004. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc.

Planning Domain Definition Language (PDDL)

Initial Condition

r-no-eff(r_1)	ktlocation(r_1,$lbwekt_1$)
lbwekt-non-empty($lbwekt_1$)	partlocation($part_{A-1}$,pt_A)
lbwek-non-full($lbwk_1$)	partlocation($part_{A-2}$,pt_A)
part-tray-non-empty(pt_A)	partlocation($part_B$,pt_B)
part-tray-non-empty(pt_B)	partlocation($part_C$,pt_C)
part-tray-non-empty(pt_C)	efftype(eff_1,$part_{A-1}$)
efflocation(eff_1,$effh_1$)	efftype(eff_1,$part_{A-2}$)
efflocation(eff_2,$effh_2$)	efftype(eff_1,$part_B$)
effhhold-eff($effh_1$,eff_1)	efftype(eff_1,$part_C$)
effhhold-eff($effh_2$,eff_2)	efftype(eff_2,kt_1)
worktable-empty($wtable$)	efftype(eff_2,$kins_1$)



Initial Condition encoded as set of predicate expressions

Goal Condition encoded as set of predicate expressions

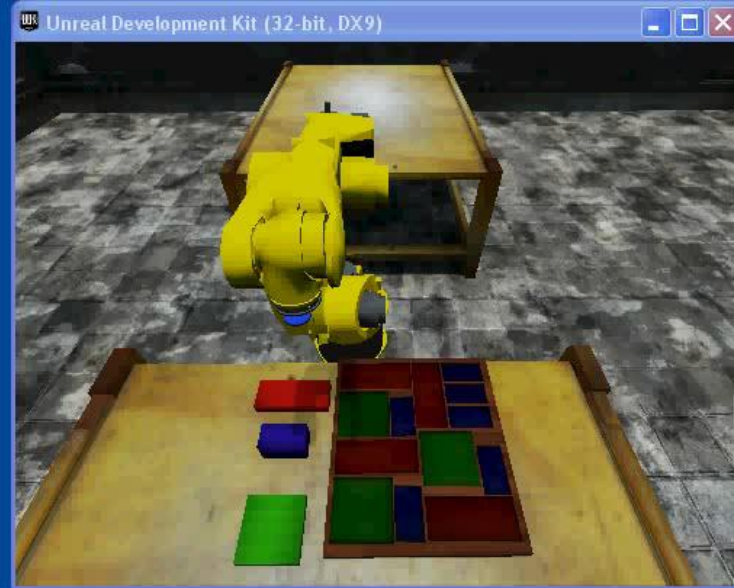
Plan and actions are necessary to move from initial predicates to goal predicates

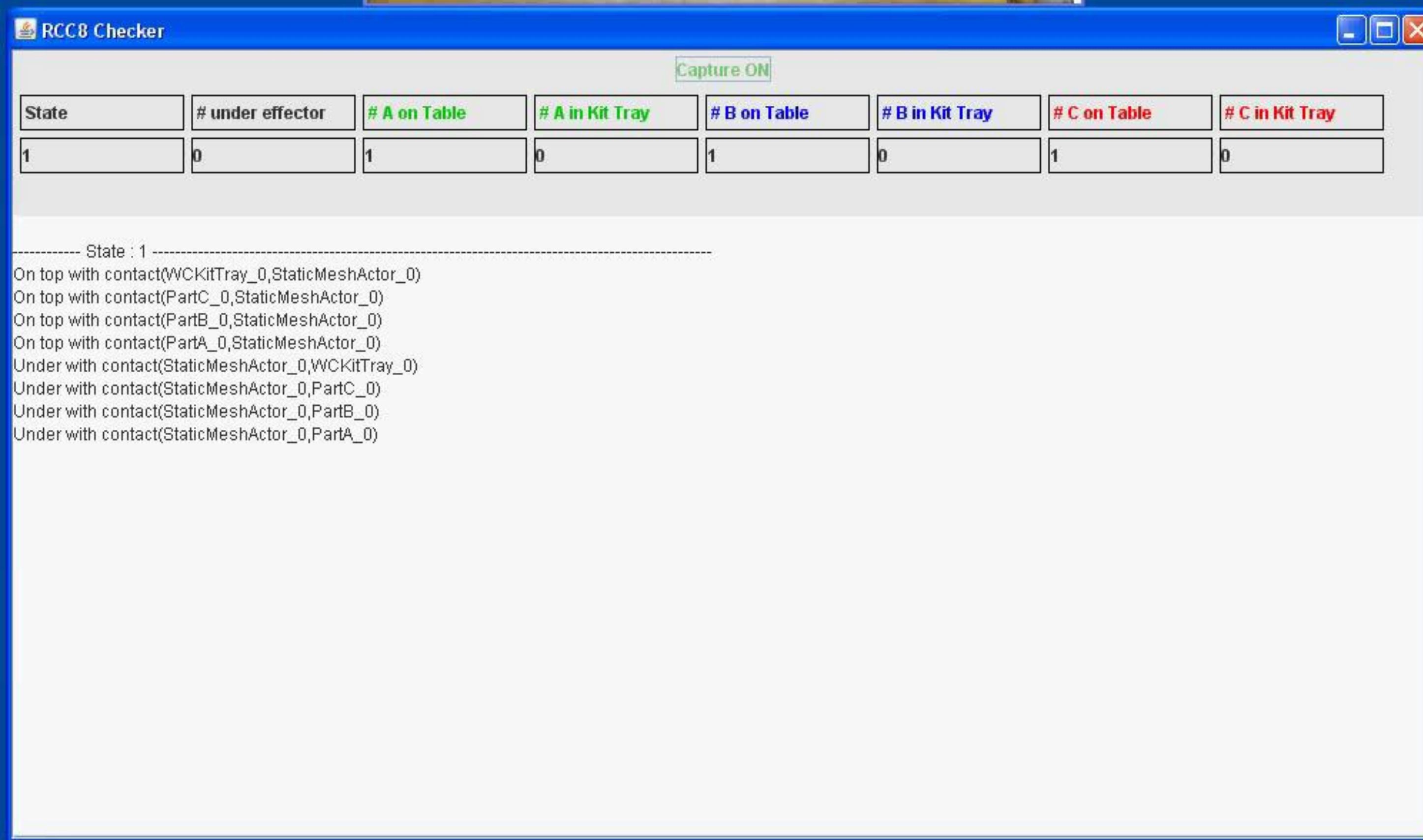
Goal Condition

partlocation($part_{A-1}$,$kins_1$)
partlocation($part_{A-2}$,$kins_1$)
partlocation($part_B$,$kins_1$)
partlocation($part_C$,$kins_1$)
kinslocation($kins_1$,$lbwk_1$)

Auto-Determination of Predicates

- Sensor processing simulated from ground truth
- Each table line is new system state (set of predicates)
- Each table column is "watched" predicate
- Bottom area shows state deltas





Planning Domain Definition Language (PDDL)

- Entire kitting domain composed of:
  - 28 predicate expressions that act as preconditions or are acted upon during effects: *rhold-empty, r-with-eff, kit-tray-location, worktable-empty, …*
  - 9 high-level operators (actions): *take-kit-tray*,

take-kit-tray(r,kt,$lbwekt$,eff,$wtable$): The *Robot* r equipped with the *EndEffector* eff picks up the *KitTray* kt from the *LargeBoxWithEmptyKitTrays* $lbwekt$.

precond	*effects*
rhold-empty(r),	¬rhold-empty(r),
lbwekt-not-empty($lbwekt$),	kit-tray-location(kt,r),
r-with-eff(r,eff),	rhold(r,kt),
kit-tray-location(kt,$lbwekt$),	¬kit-tray-location(kt,$lbwekt$)
eff-location(eff,r),	
worktable-empty($wtable$),	
efftype(eff,kt)	

Implementation of PDDL Actions

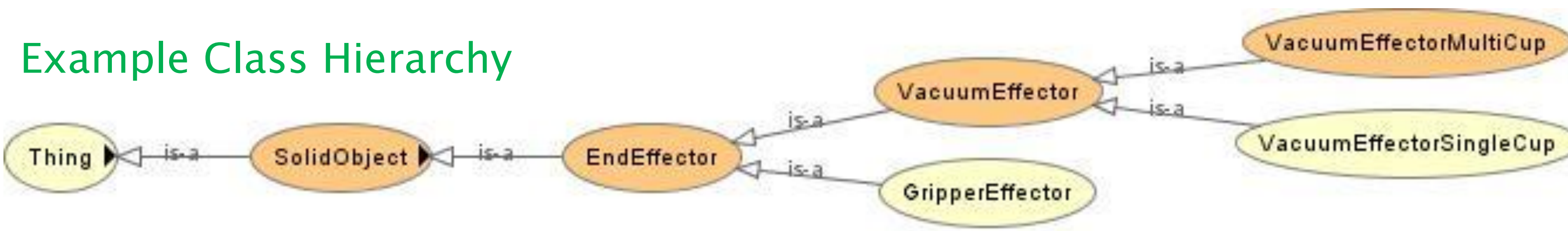
- Interface to many simulated and real robots
- Provide lowest common denominator actions for robot execution
  - Move
  - Open/Close Gripper
  - …
- PDDL actions implemented as scripted sequence of these actions (called the Canonical Robot Command Language)

SVR – PDDL Limitations

- Planning language not designed for knowledge representation
  - Lacks a taxonomy of objects
  - No representation of relationships or constraints between objects

- An ontology provides for all of the above through entities, data properties, and object properties
  - Web Ontology Language (OWL) can be used to represent objects, relationships, and constraints
  - OWL Services (OWL-S) and the Semantic Web Rule Language (SWRL) can be used to represent actions, preconditions, and effects
- Tools exist to move from OWL and OWL-S/SWRL to PDDL representations

Example Class Hierarchy



Issues With Dynamics

- PDDL/OWL representation is a set of static files with no real-time information
  - Can auto-generate MySQL database from OWL instance file and maintain database with sensor processing (access functions also auto-generated)
  - Can automatically re-generate OWL instance file to allow for continued reasoning

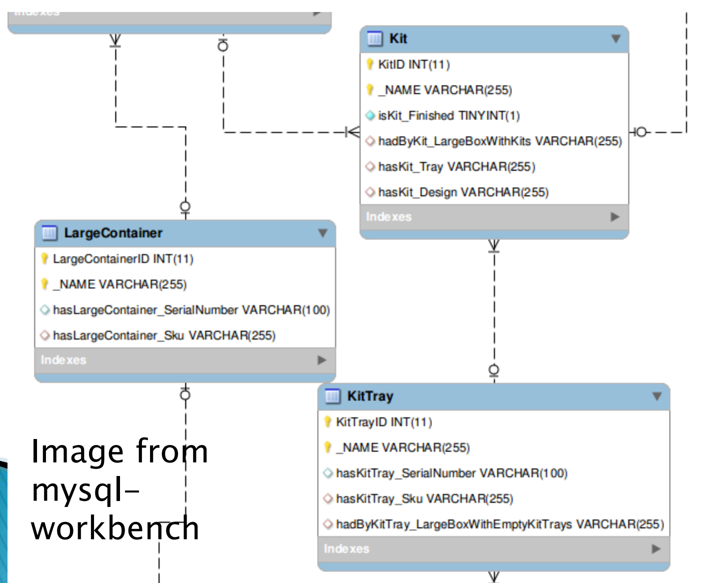
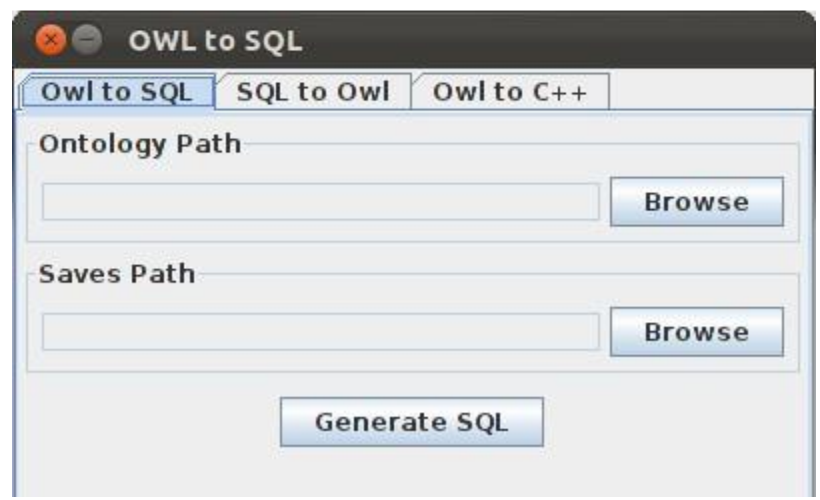


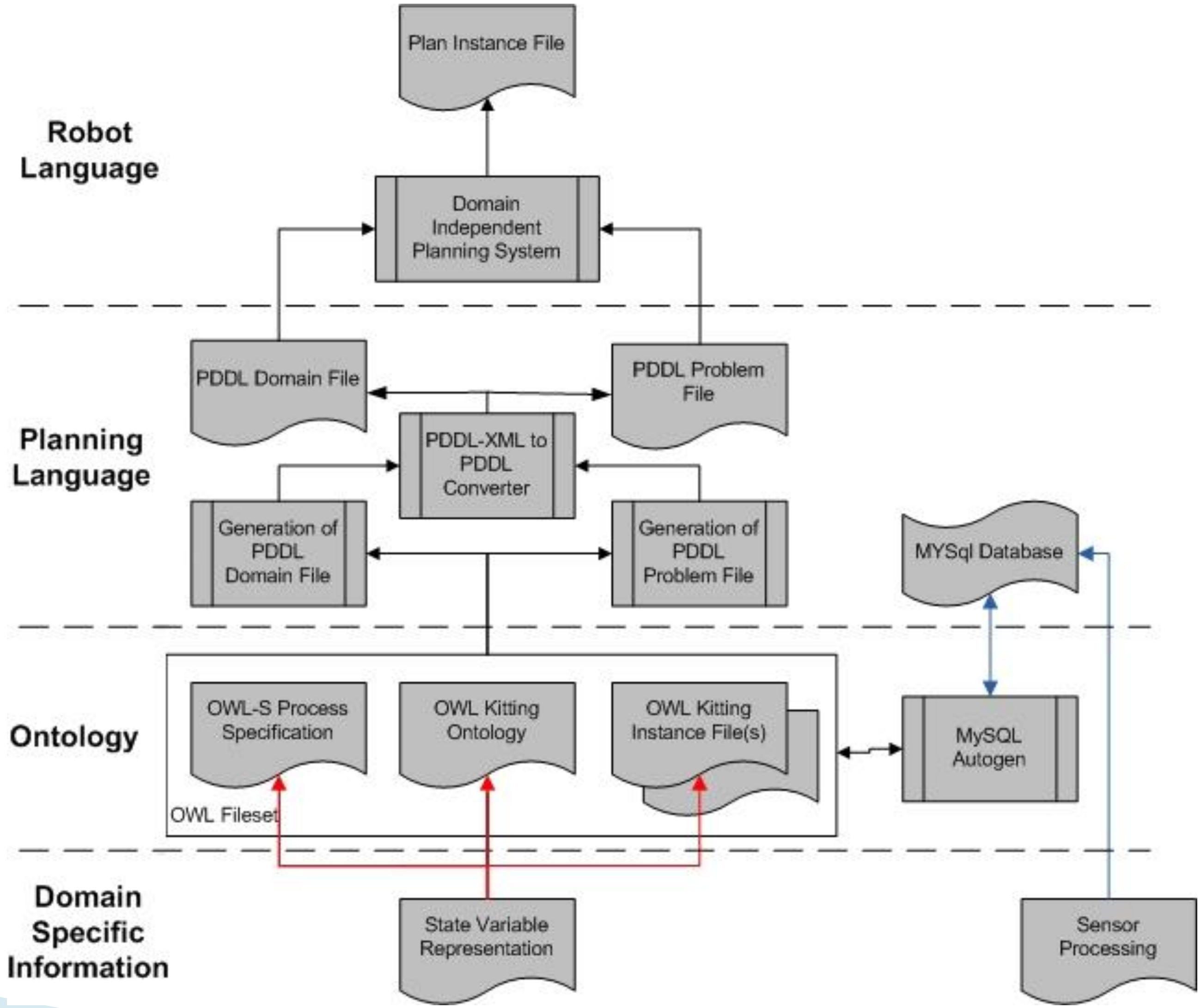
Image from mysql-workbench



Knowledge Generation

	System Execution
Auto-gen	MySQL database, PDDL plan file
Auto-gen	PDDL Domain, Problem files
Hand created	OWL-S, SWRL Rule files
Hand created	OWL Model, Initial Condition, Goal Condition files

Knowledge Architecture



Implementation Issues: OWL/OWL-S

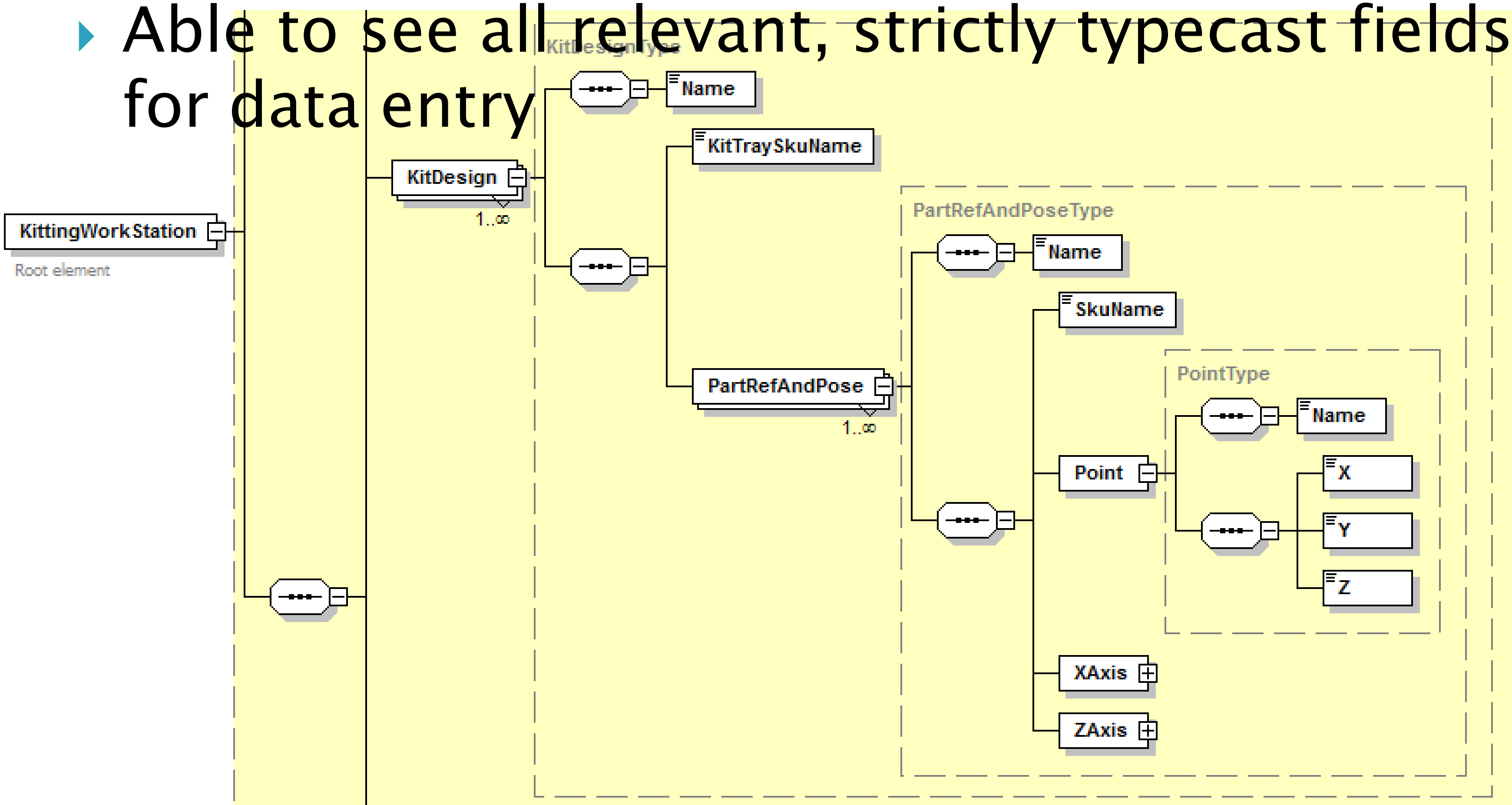
- Naïve user, but…
  - SWRL Rules
    - Very verbose data entry for rule, pre-conditions, and effects
    - Difficult/impossible to represent negation
  - OWL Instance Modification
    - Open world assumption allows spelling errors
    - Required to typecast every data entry
    - Need to follow complex trail of tables and pointers
    - Removal of instance requires careful clean-up to remove “hanging” entries
- Is there another way?
  - Investigated use of XML and PDDL with automatic generation of OWL files

```
<!-- http://www.nist.gov/el/ontologies/IPMAS-Activities10.owl#KitTrayInLargeBoxWithEmptyKitTrays -->

<owl:NamedIndividual rdf:about="&ontologies;IPMAS-Activities10.owl#KitTrayInLargeBoxWithEmptyKitTrays">
    <rdf:type rdf:resource="&expr;SWRL-Condition"/>
    <rdfs:label>hasSolidObject_PhysicalLocation(TakeKT_KitTray, ?rli) RelativeLocationIn(?rli) &amp; hasPhysicalLocation_RefObject(?rli,TakeKT_LargeBoxWithEmptyKitTrays) &amp;  </rdfs:label>
    <rdfs:comment>A KitTray must have a PhysicalLocation which is a RelativeLocationIn (?rli) that must point to a hasPhysicalLocation_RefObject which is a LargeBoxWithEmptyKitTrays.</rdfs:comment>
    <expr:expressionObject>
        <rdf:Description/>
    </expr:expressionObject>
</owl:NamedIndividual>
```

XML Representation

- Schema view clearly shows elements, element structure and properties, and classes
- Able to see all relevant, strictly typecast fields for data entry



More Tools

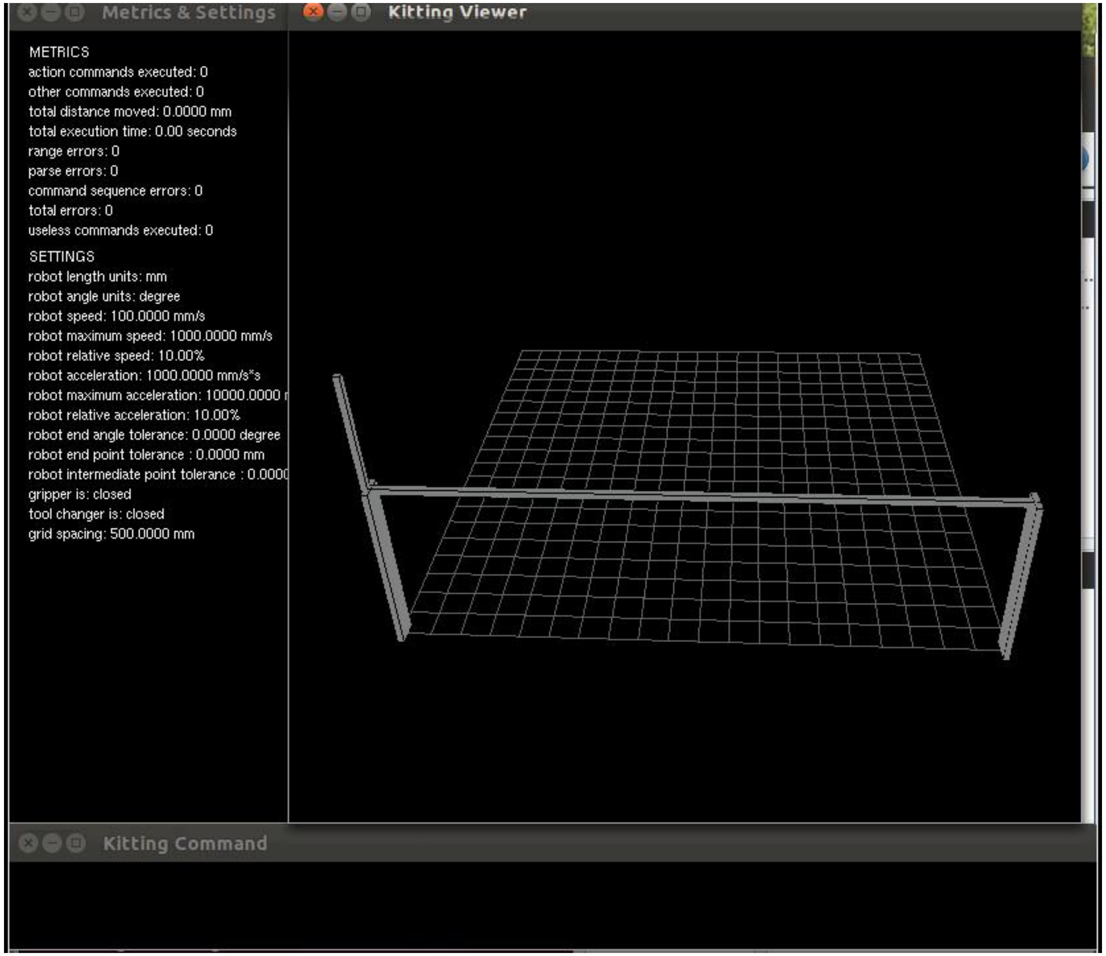
- We have created XML->OWL conversion tools
  - Can automatically generate an OWL “instance” file from an XML file
  - Work currently underway to generate OWL “model” file from XML schema
- XML provides
  - Closed world assumption
  - Type checking
  - Consistency checking

System Alternatives?

	System Execution	System Execution
Auto-gen	MySQL database, PDDL plan file	MySQL database, PDDL plan file
Auto-gen	OWL Model, Initial-, Goal-Condition files, PDDL Problem file	PDDL Domain, Problem files
Hand created	PDDL Domain file, code for predicate generation	OWL-S, SWRL Rule files
Hand created	XML Schema, Initial Condition, Goal Condition files	OWL Model, Initial Condition, Goal Condition files

Future Work, Metrics

- Time to build
- Distance robot traveled
- Number of actions
- Useless commands
- Errors
- …



Future Work

- Current work assumes perfect actions; need techniques to gracefully cope with errors
- Migrate techniques onto real hardware with real image processing
- Extend work to apply to general assembly operations
- Develop metrics for kit building