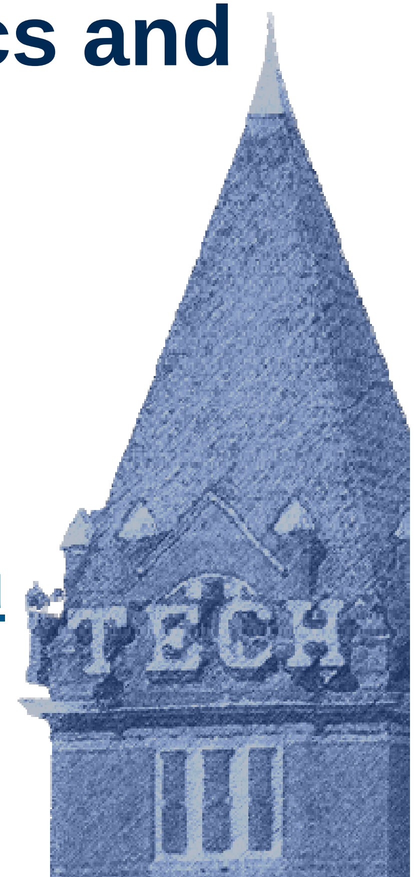
IEEE RAS Ontologies for Robotics and Automation
Industrial Subgroup

Dr. Stephen Balakirsky

Georgia Tech Research Institute

Stephen.Balakirsky@gtri.gatech.edu

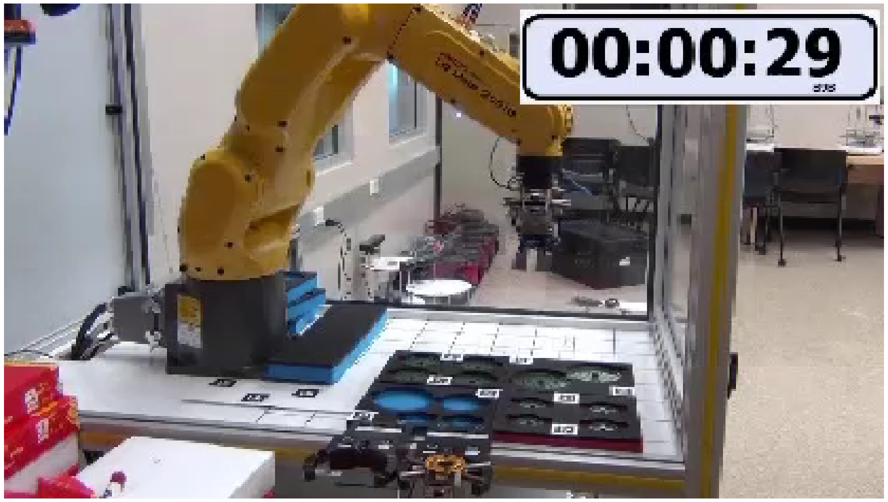
(404) 407-8547



Georgia Tech Research Institute

Our Goals

- Perform cooperative research on agile assembly
  - Robot, planning system, vision system, … agnostic
  - Direct CAD to assembly
  - Error recovery and correction
  - Fixtureless assembly
- Will allow:
  - Lot-size 1 assembly on automated lines
  - Reduced line down time due to programming
  - Less human intervention due to assembly errors
  - More competition in all aspects of robotics



Teach Pendant Programming (~20 min)

Who Are We?

- Existing standard:
  - P1872-2015 IEEE Standard for Ontologies for Robotics and Automation
    - Standardizes how artificial agents represent and communicate their knowledge about the world
    - Defines core ontology that represents the most general concepts, vocabulary, relations, and axioms
- Prosed subgroup:
  - "Industrial Robot Ontology"
  - Developing set of canonical languages for robot cell control and description of cell
- This talk will cover past, present, and future work of group

Subgroup’s First Steps: Robot Agnostic Operations

- Examining robot agnostic, low-level control
- Desire to create an ontology that allows industrial cells to be more flexible and agile
- Canonical Robot Command Language (CRCL)
  - Basis set of commands
  - Formal definition of these commands
  - Will result in ability to utilize the set of commands on different vendor’s robots with same results
  - Implemented as XML (schema and instance files)

Two Classes of Commands

Robot Agnostic

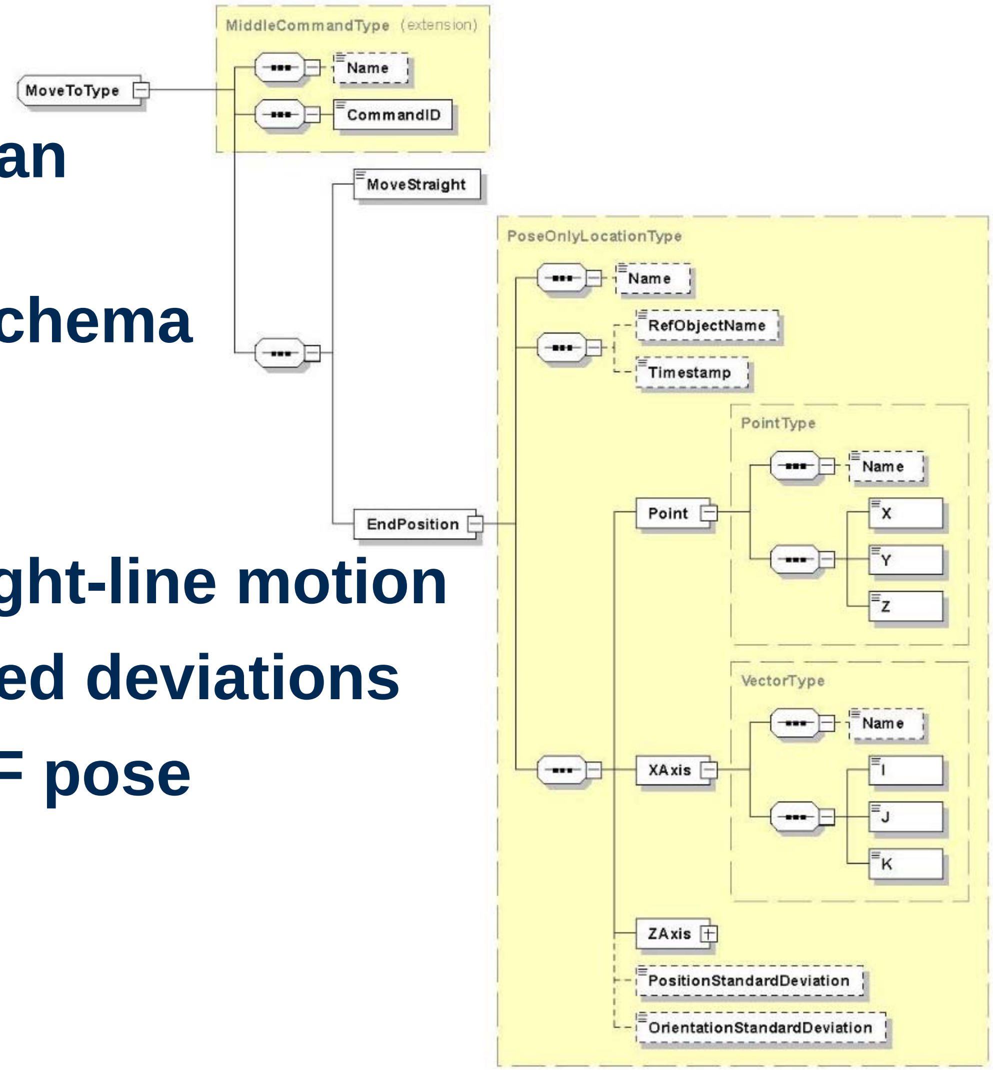
- Initialization/Termination
- Open/Close tool changer
- Dwell
- Get status
- Message (Comment)
- Linear movement
  - Move through
  - Move to
- Screw motion
- Run program
- Set (acc, speed, units, tolerance)
- Set end effector operation
- Stop motion

Robot Specific

- Joint related
  - Control mode (position, force, torque)
  - Actuate joint(s)
  - Configure joint(s) report
- Set parameters (robot, end effector)

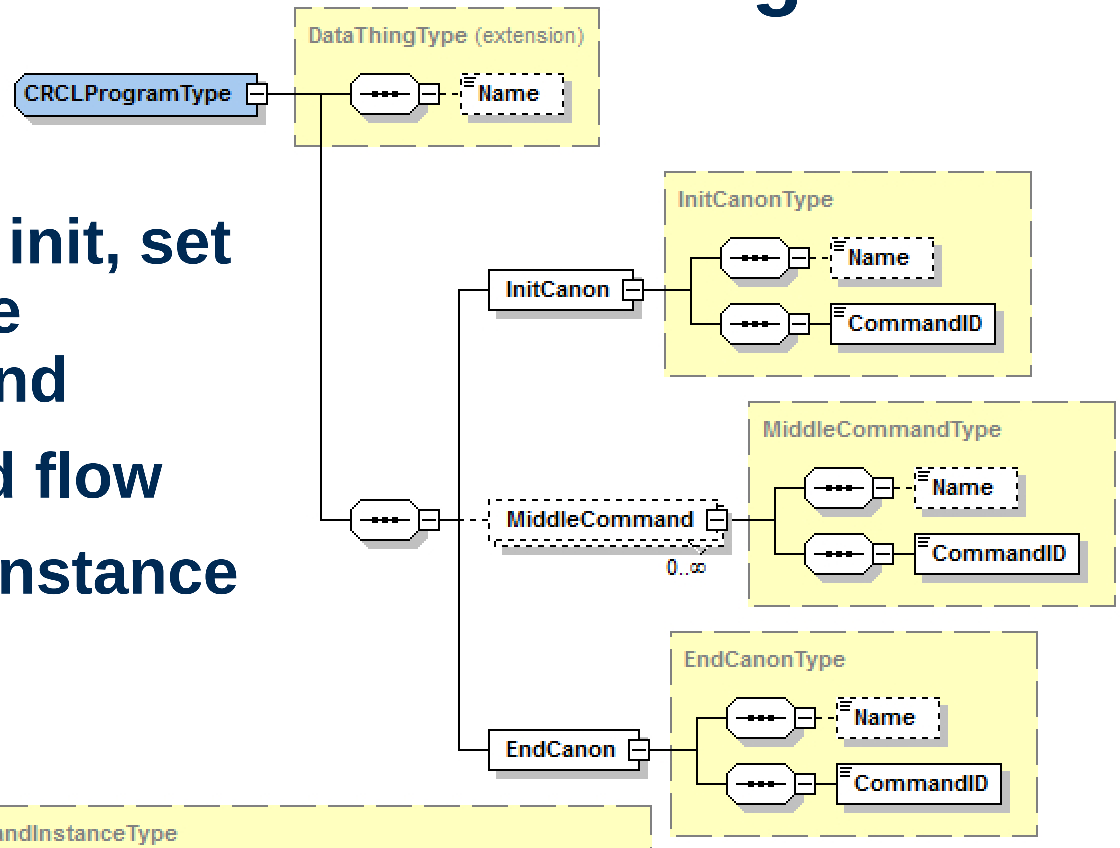
Example Command

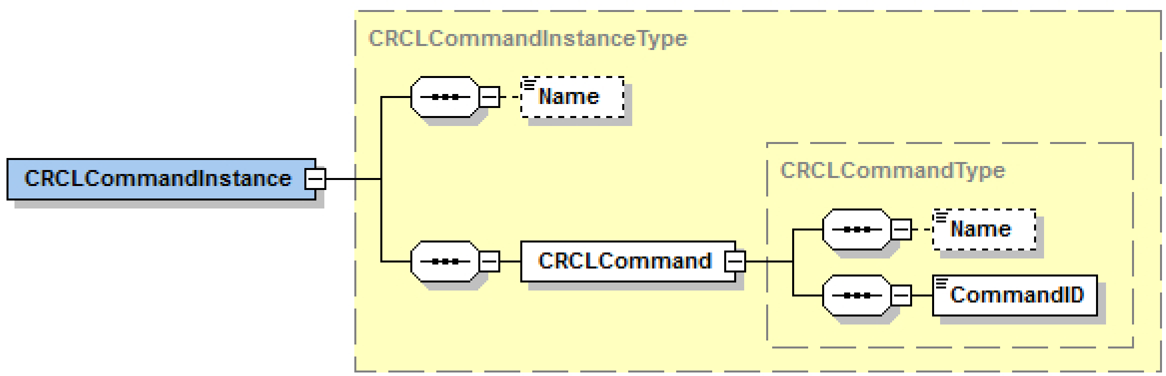
- **MoveTo allows robot motion to single Cartesian pose**
- **Composed of multiple schema elements**
- **Allows:**
  - **Requirement for straight-line motion**
  - **Specification of allowed deviations**
  - **Specification of 6-DOF pose**



Sets of Commands – A Robot Program

- Static program
  - Program includes init, set of “middle” middle commands, and end
- Interactive command flow
  - Single command instance with handshake





Sample CRCL Command


```
<CRCLCommandInstance >
 <Name>autoId8</Name>
 <CRCLCommand xsi:type="MoveToType">
  <Name>autoId7</Name>
  <CommandID>4</CommandID>
  <MoveStraight>true</MoveStraight>
  <EndPosition>
   <Name>autoId9</Name>
   <Point>
    <Name>autoId10</Name>
    <X>0.343627</X>
    <Y>-0.259000</Y>
    <Z>-0.338000</Z>
   </Point>
   <XAxis>
    <Name>autoId11</Name>
    <I>-0.947210</I>
    <J>-0.320613</J>
    <K>0.000000</K>
   </XAxis>
  <ZAxis>
   <Name>autoId12</Name>
    <I>-0.000000</I>
    <J>0.000000</J>
   <K>-1.000000</K>
   </ZAxis>
  </EndPosition>
 </CRCLCommand>
</CRCLCommandInstance>
```


- Command bound to fixed Cartesian position or fixed set of joint angles
  - Command/program only works if conditions are identical to those that existed during programming
- Low-level functionality

CRCL In Operation

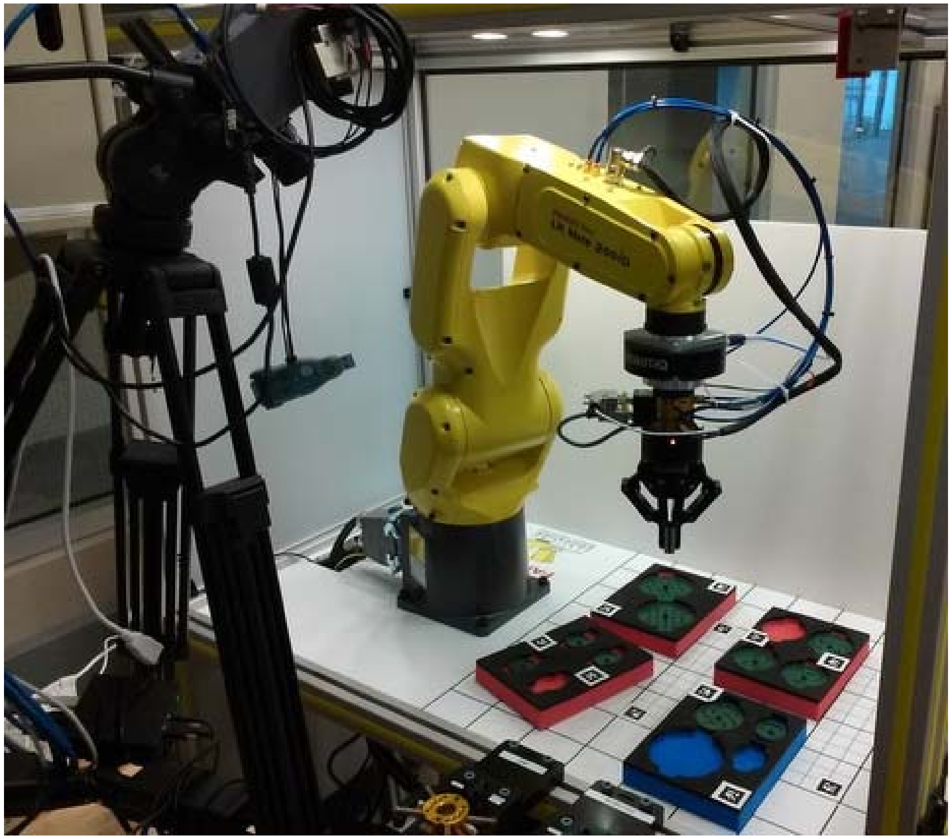
- Open loop control under CRCL
- Benefits
  - Robot agnostic
  - Simple, standard command set
- Issues
  - Parts must be fixtured or set in same location on each run
  - Unable to detect or respond to failures (lacks agility)
  - Robot may understand command, but can it execute?

CRCL Next Steps

- Part of ROS Industrial (https://github.com/ros-industrial/crcl)
- Gather comments on utility, missing components, opportunities for improvement
  - First part of industrial ontology to be proposed as standard

Current Explorations: Knowledge Driven Robotics

- **Allow programming of robot to be based on behavior composition rather than low-level programming**
- **Needs:**
  - **Domain independent behavior-based planning system interface**
  - **Vendor independent robot control language**
  - **Vendor independent sensor control language**
  - **Encoded domain knowledge**



Robot cell contains planner, arm, vision system, and assembly components

Planning Domain Definition Language (PDDL)

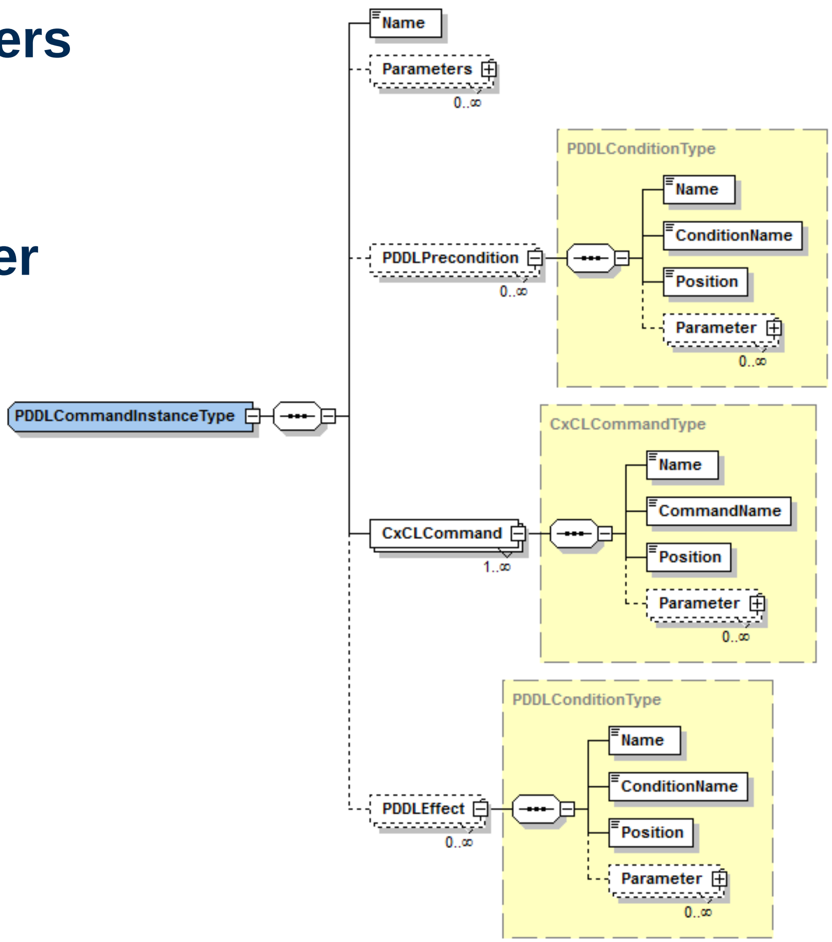
PDDL Output

(LookForPart robot_1 sku_part_Vex36 part_Vex36_tray)
(SetGrasp robot_1 sku_part_Vex36 part_gripper)
(TakePart robot_1 part_gripper)
(LookForSlot robot_1 sku_part_Vex36 kit_s2m1l1)
(PlacePart robot_1)

- **First developed in 1998 by Drew McDermott et. al**
  - **Widely used by AI planning community**
  - **Domain independent**
- **Desire to directly encode PDDL into ontology**
  - **Modify ontology not robot program for new commands**
  - **If CRCL safe, all PDDL should also be safe**
- **Exploring what predicates and PDDL commands are necessary for operation**

PDDLCommand

- Translation from PDDL and parameters to CxCL and parameters
  - More intuitive than CxCL
  - Supports late binding of parameter values
  - Includes multiple command languages
    - CRCL (robot motion commands)
    - CVCL (vision commands)
    - CMCL (pose math commands)
- Includes effects of actions to allow automatic action verification
- Include preconditions of actions for action validity checking



TakePart Command

TakePart

Parameters (2)

	Name	PDDLParameter...	Position
1	TakePartRobot	Robot	1
2	TakePartEndeffector	EndEffector	2

CxCLCommand (6)

	Name	CommandName	Position	Parameter
1	TakePartAddGrasp	CVCLAddPoseType	1	Parameter (3)
2	TakePartAddSafeOffset	CVCLAddPoseType	2	Parameter (3)
3	TakePartMoveToSafe	CRCLMoveToType	3	Parameter (2)
4	TakePartMoveToPart	CRCLMoveToType	4	Parameter (2)
5	TakePartSetEndEffector	CRCLSetEndEffectorType	5	Parameter (2)
6	TakePartMoveToSafe2	CRCLMoveToType	6	Parameter (2)

Parameter (3) of TakePartAddGrasp:

	Name	ParameterKind	Value	Position
1	TakePartAddGraspPart	Constant	FoundPart	1
2	TakePartAddGraspGrasp	Constant	FoundGrasp	2
3	TakePartAddGraspStoreLocation	Constant	MoveToLoc	3

PDDLEffect

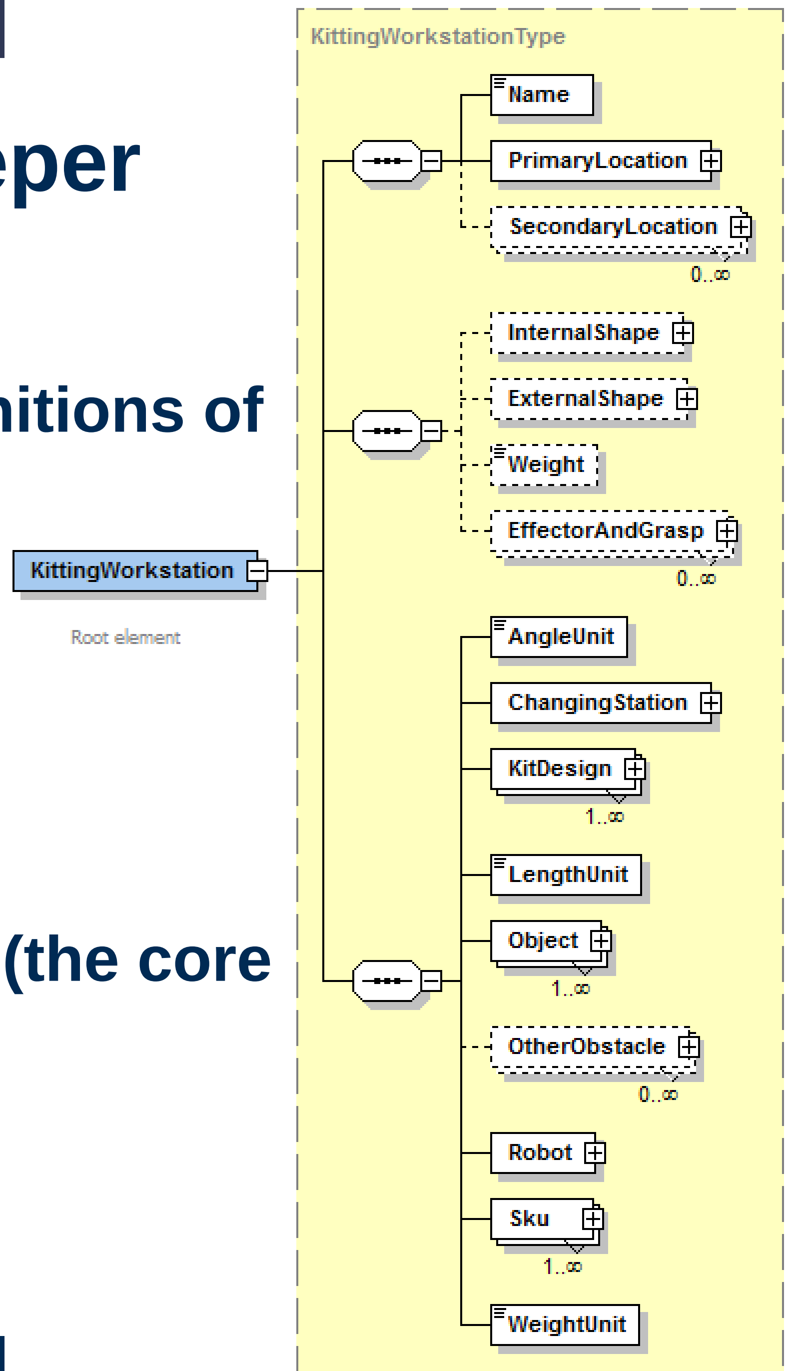
Name	TakePartEffectSetParent
ConditionName	CECLSetParentType
Position	1

Parameter (2)

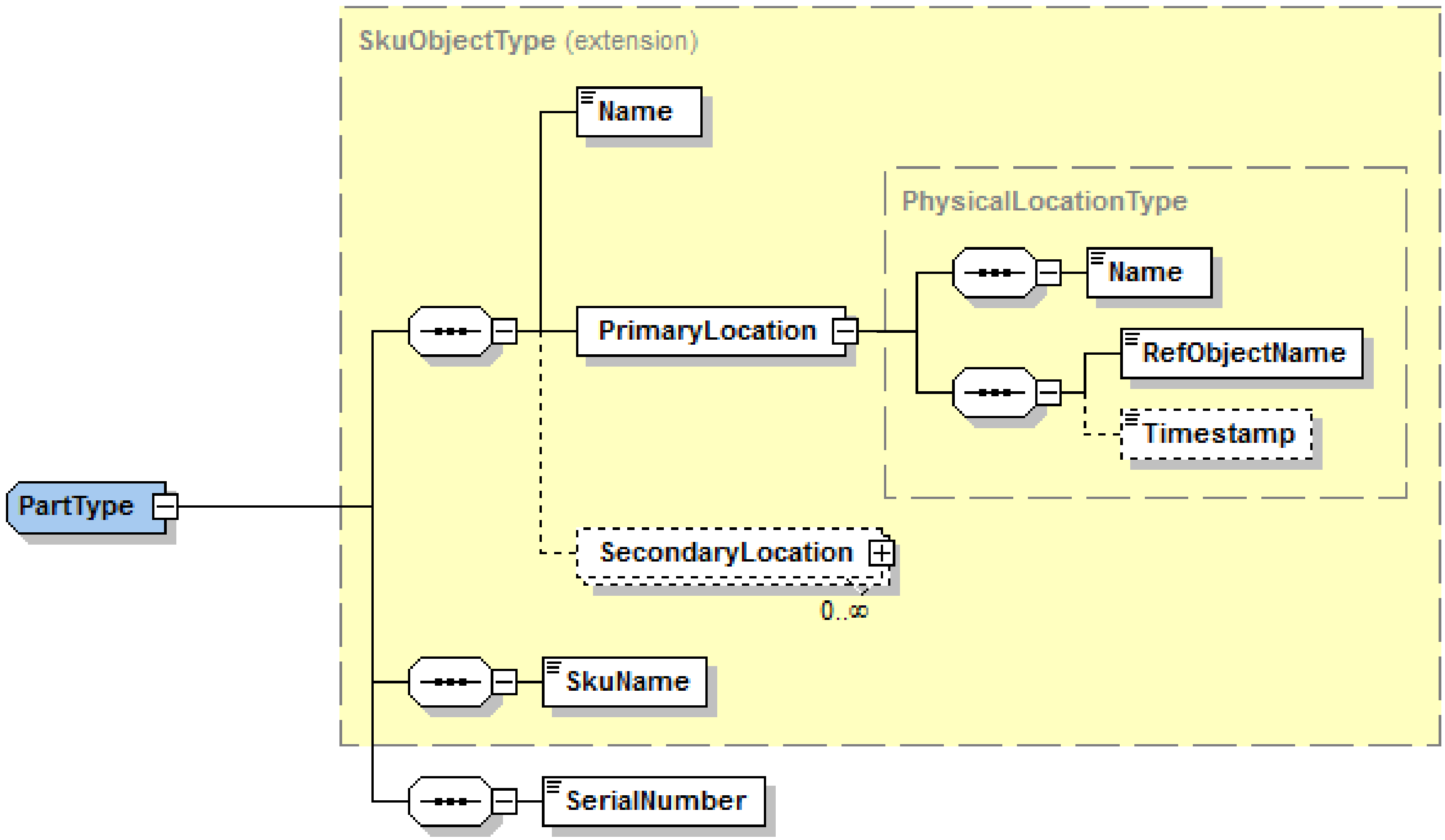
	Name	ParameterKind	Value	Position
1	TakePartEffectSetParentPart	Constant	FoundPart	1
2	TakePartEffectSetParentParent	PDDLParameter	2	2

Commands Imply Deeper Knowledge

- “Kiting” ontology contains definitions of items such as:
  - Locations
  - Designs
  - Parts, trays, kits, …
  - Grasping
- Based on concepts from CORA (the core robot ontology)

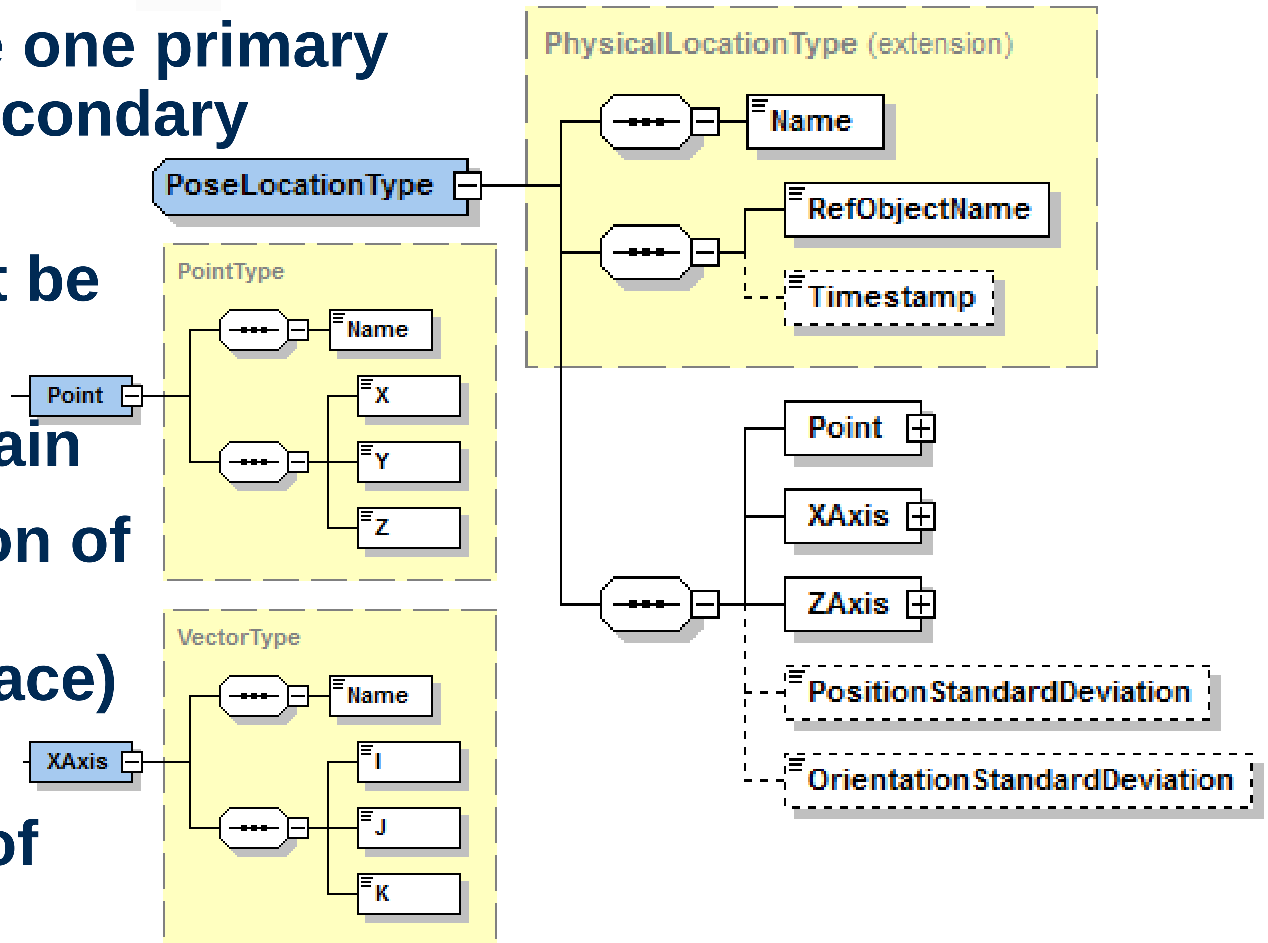


Structure of a "Part"

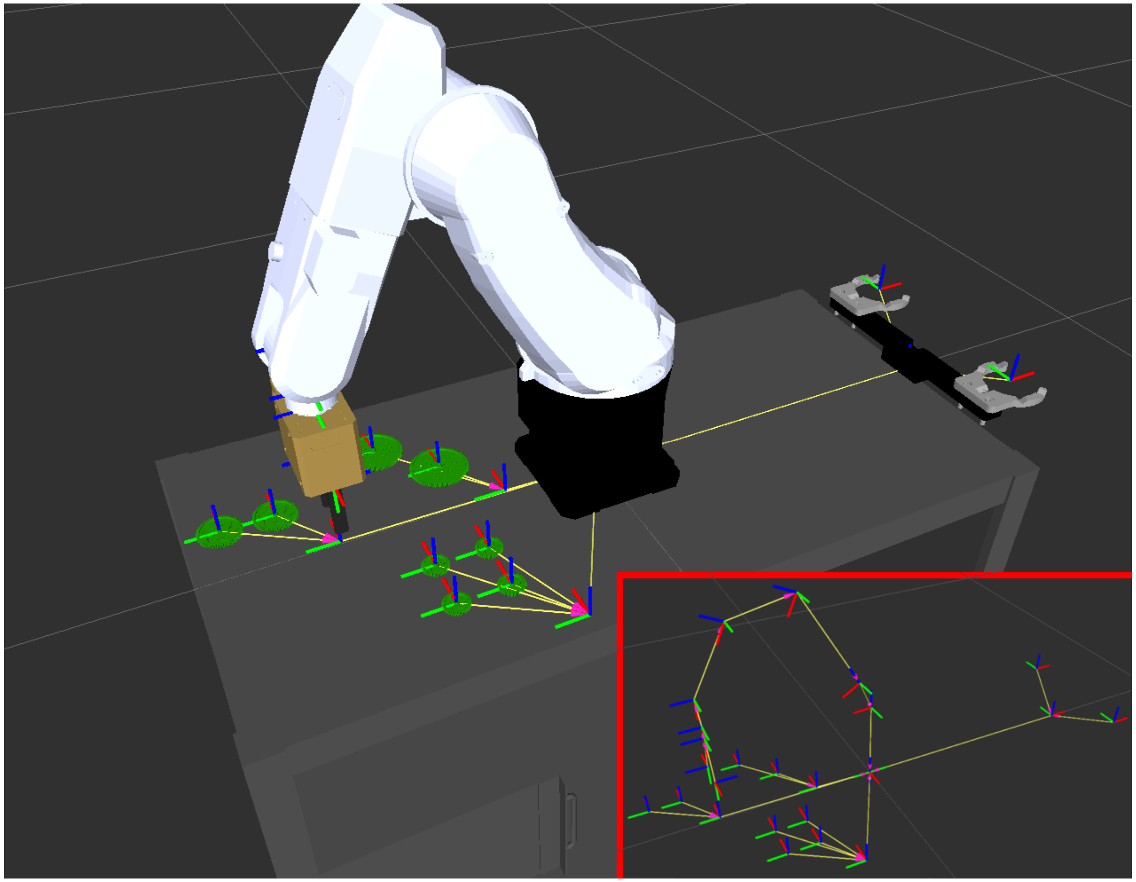


Recognize the location of the parts trays, kit trays, and their contents

- Parts may have one primary location and secondary locations
- Locations must be consistent
- Locations contain
  - Point (location of new frame in reference space)
  - Unit vectors (orientation of new frame)

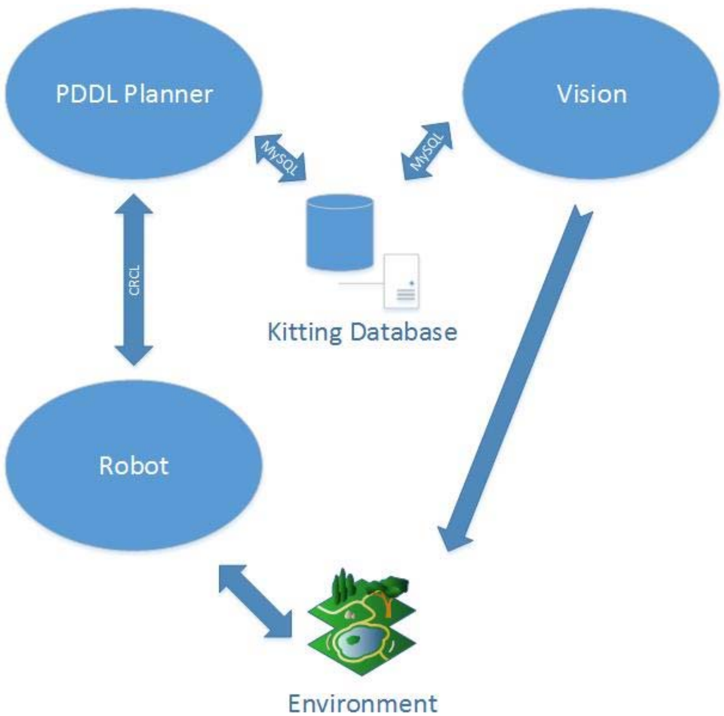


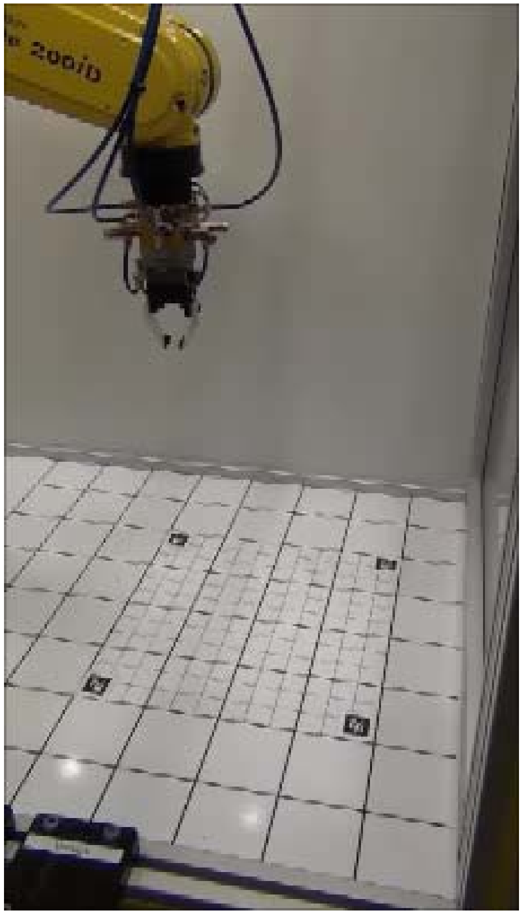
Position Hierarchy



Closed Loop Control

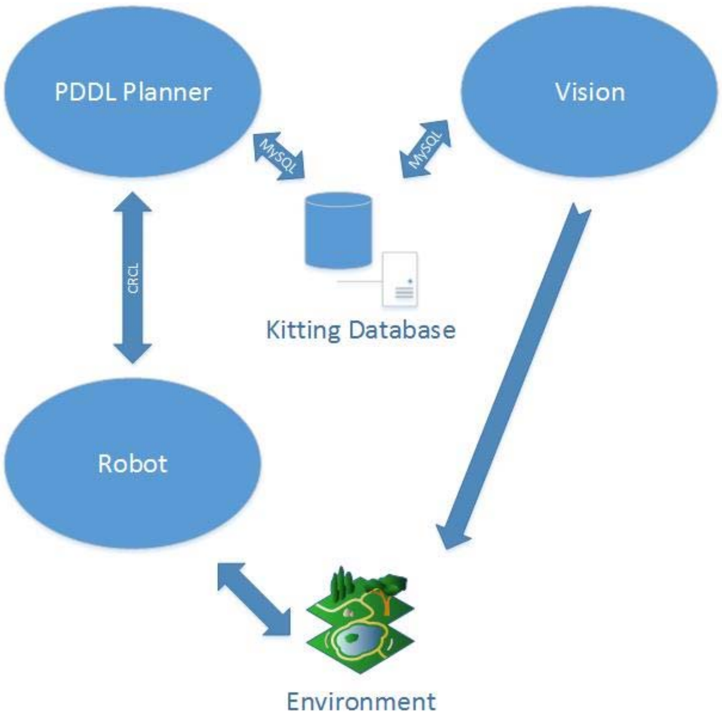
- Combination of CRCL and Kitting Database allow for closed loop control of system
- Closed over entire PDDL program
- Single write of date by vision



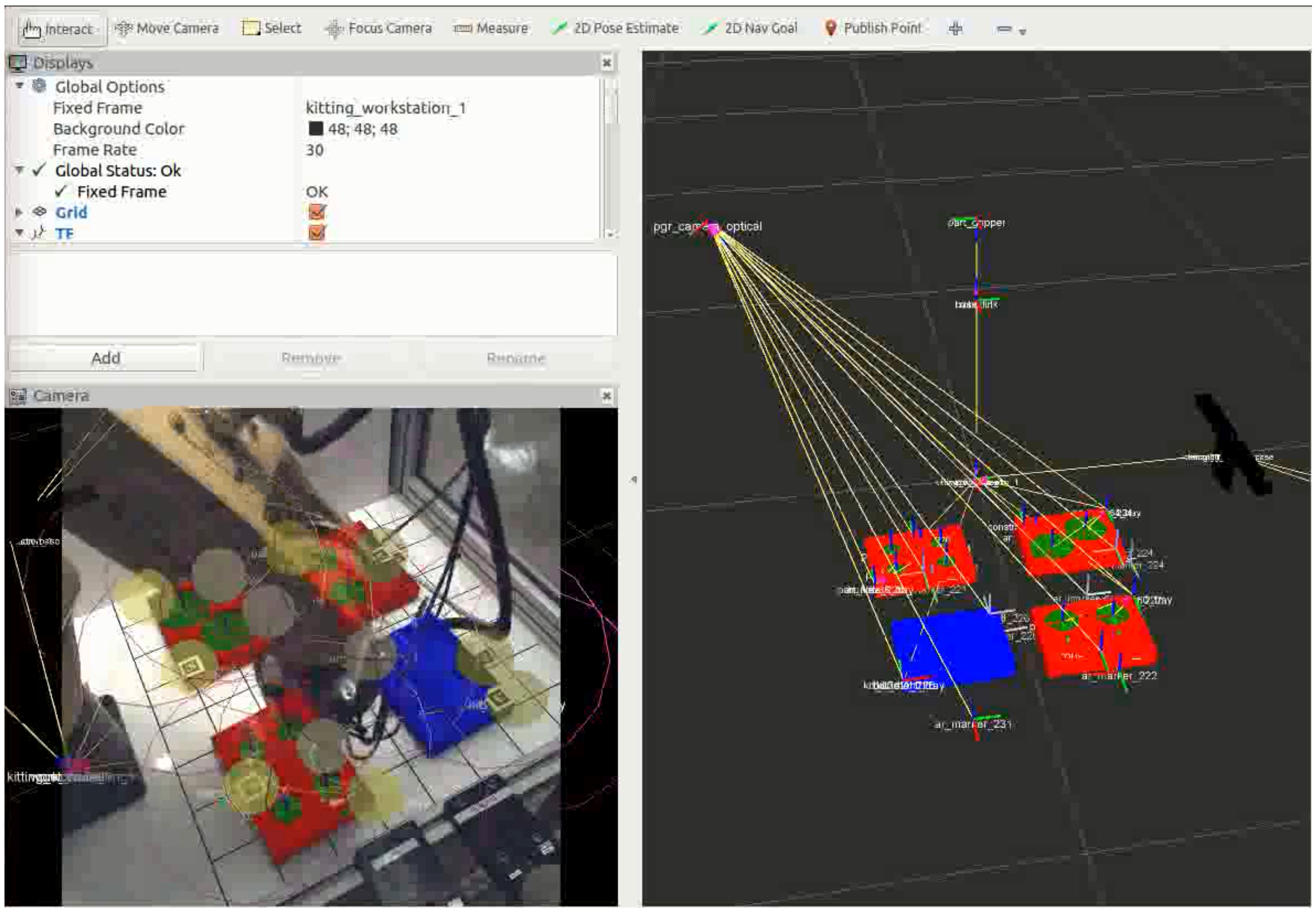


Tightening The Loop

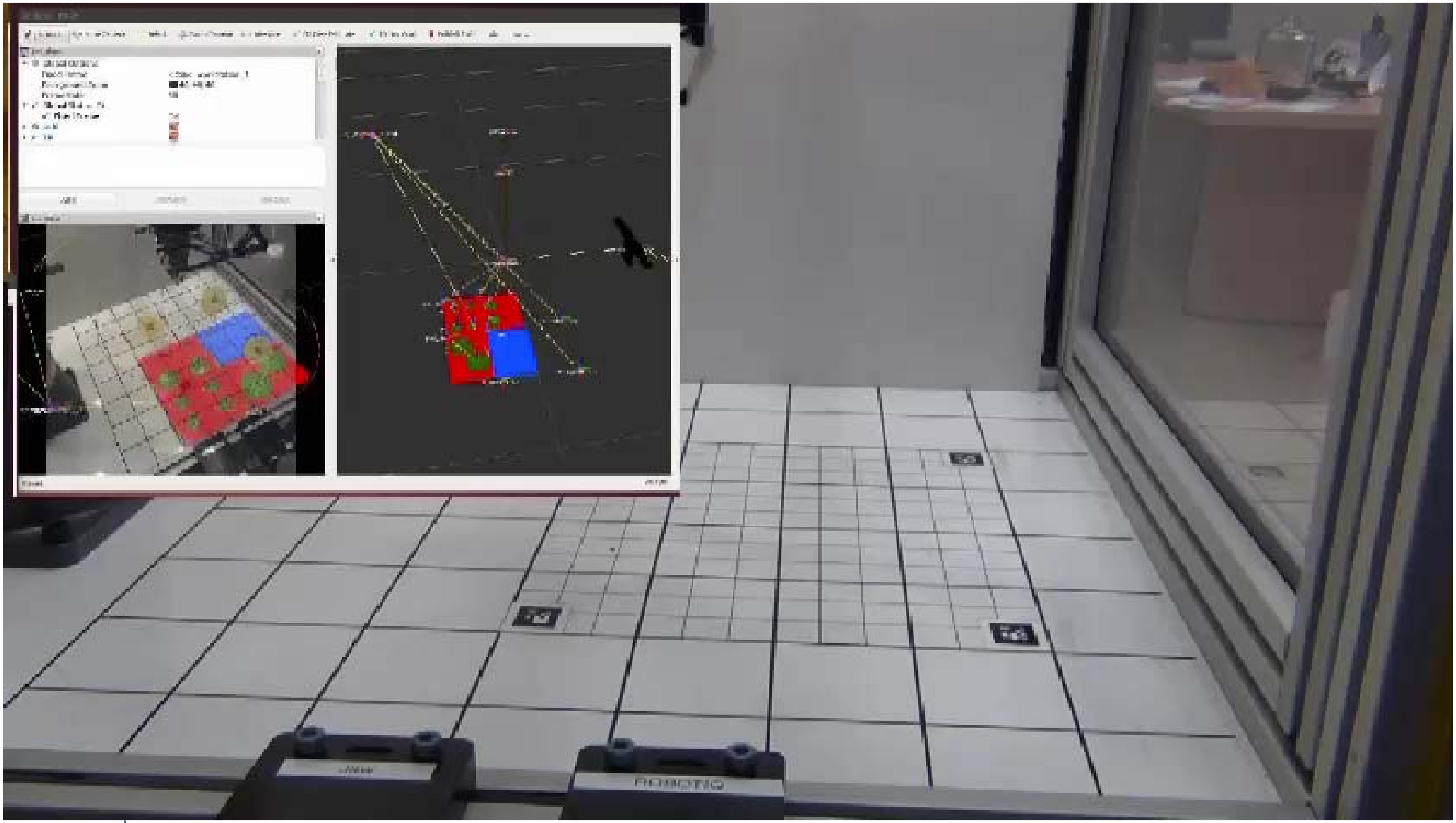
- Combination of CRCL, CVCL, and Kitting Database allow for tight closed loop control of system
- Closed on a command-by-command basis
  - Vision system updates database after each operation



What Video Will Show

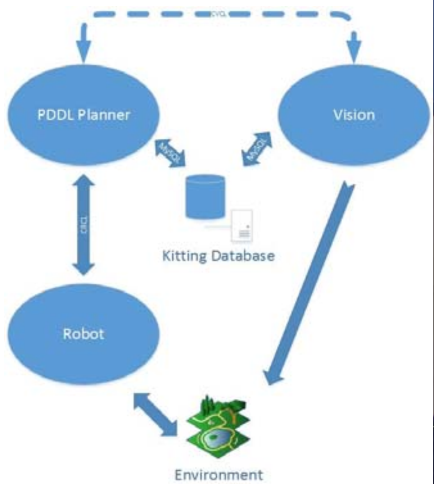


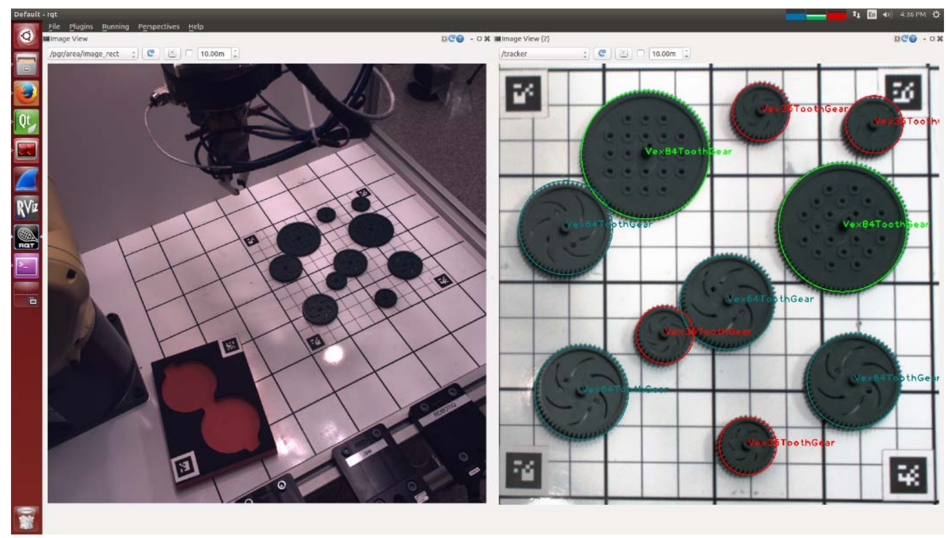
Closed Loop Operation



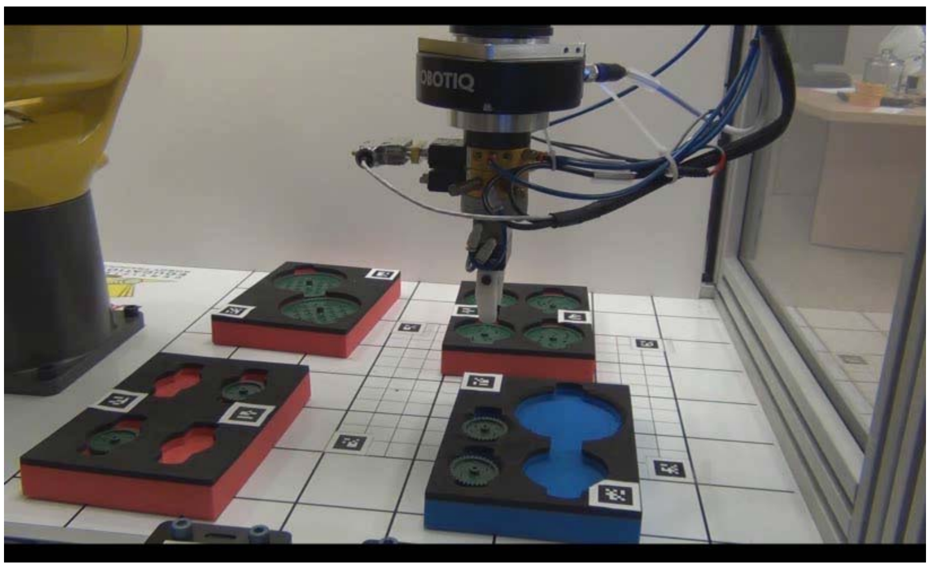
Vision/Planning Handshake, CVCL

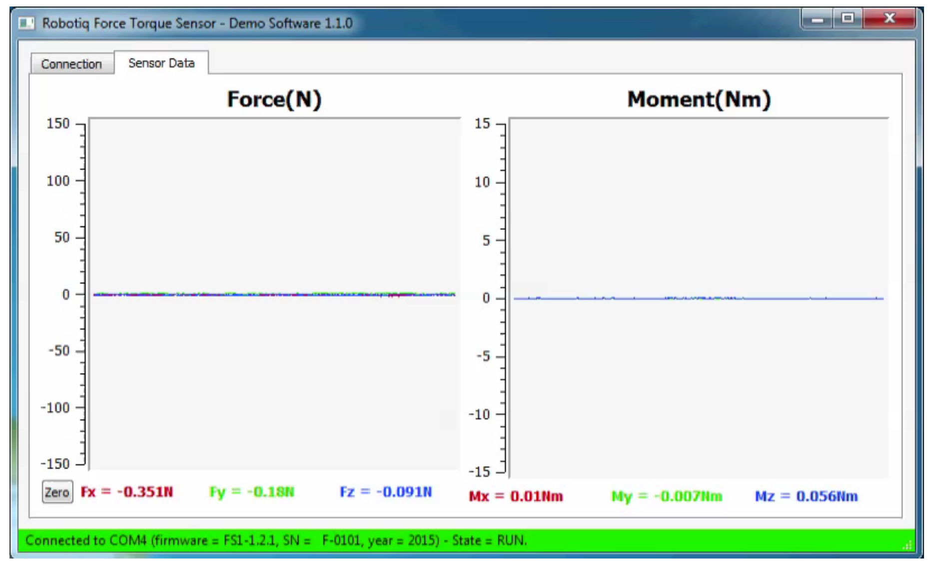
- Allows planner to request vision updates
  - Region of interest
  - Specific models
- Allows vision system to tell planner when vision is stable





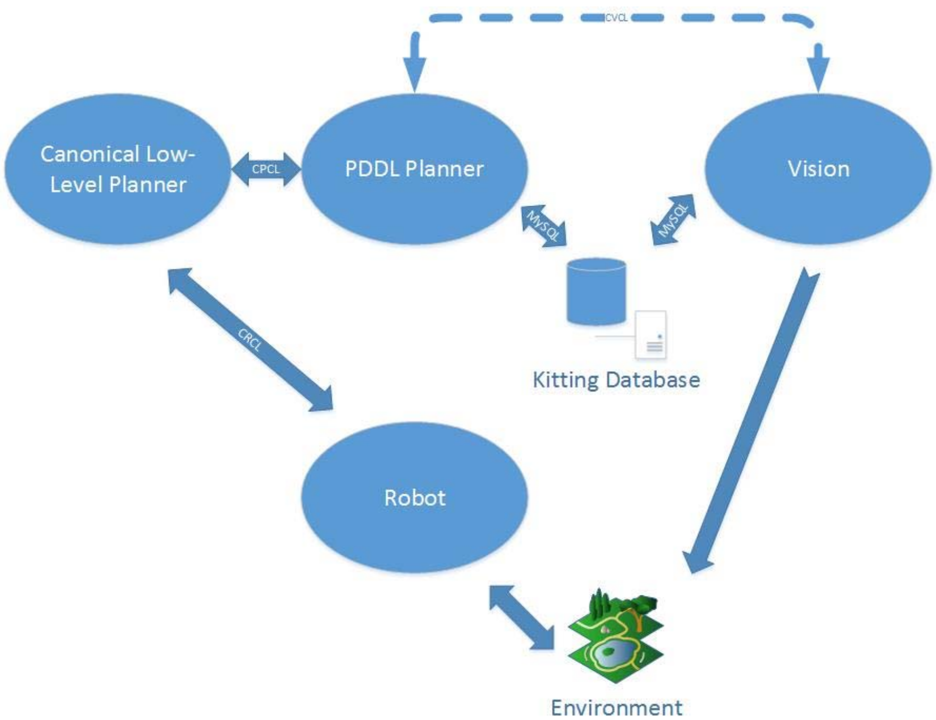
Extension for Intelligent Controller





Family of Canonical Languages

- Canonical Robot Command Language – Control of robots
- Canonical Vision Command Language – Control of vision systems
- Canonical Planning Control Language – Control of planning systems



How To Get Involved

- Participate in monthly IEEE ORA telecons
- Mailing list: iora@lists.gatech.edu
- Contact Steve or Craig for more information (contact info on next slide)

Contact Info

Stephen Balakirsky
Senior Research Scientist
GTRI
Atlanta, GA
Stephen.Balakirsky@gtri.gatech.edu

Andrew Price
Graduate Student
GTRI
Atlanta, GA
arprice@gatech.edu

Craig Schlenoff
Group Leader
NIST
Gaithersburg, MD
Craig.Schlenoff@nist.gov

http://www.nist.gov/el/isd/crcl.cfm