A Mosaic of Models as the Driver for Domain-Specific Modeling in the Robotics Industry

Advanced Research & Technology Office
MathWorks

Anastasia Mavrommati, PhD
amavromm@mathworks.com
stacymav.github.io
Robotics and DSL in Research Conferences

What about Robotics in the manufacturing industry?
The Automatic Servant of Philon

~300 BC

Source: Kotsanas Museum of Ancient Greek Technology, Athens, Greece
www.kotsanas.com
UNIMATE robot (serving liquor)

~300 BC  
1954 AD

Source: https://robots.ieee.org/robots/unimate/
When did the need for programming languages in Robotics appear?

First industrial robot: 1954

First robot programming language in academia: 1973

First robot programming language in industry: 1979

Teach the robot by guiding it through motion

- Physical guide through
- Using the teaching pendant

First desideratum: Move robot!

Move to location PART. Then, move to location BOX.
First solution: Textual programming languages

Variable Assembly Language

ROBOT USERS

1979
First solution: Textual programming languages

Teaching Pendant Language

Robotic Control Language

40 years!

1979

2019

✓ Programmed on the robot computer

✓ Robot operation stopped

✓ Similar semantics (assembler-like)

✓ More GPL-like syntax

ROBOT USERS

PART

BOX

1. ?APPRO PART, 50°
2. ?APPRO PART
3. ?CLOSEI®
4. ?DEPART 150°
5. ?APPROS BOX, 200°
6. ?APPROS BOX
7. ?OPENI®
8. ?DEPART 75°
9. ?APPROS BOX

MOVES PART®

MOVE BOX®

L P[1:PART] 100%

J P[2:BOX] 50%
Target groups for Robotics applications

- ROBOT USERS
- SOFTWARE DEVELOPERS
- ROBOT BUILDERS
Target groups for Robotics applications

ROBOT USERS

SOFTWARE DEVELOPERS

ROBOTICS RESEARCHERS

ROBOT BUILDERS
Target groups for Robotics applications

- ROBOT USERS
- SOFTWARE DEVELOPERS
- ROBOTICS RESEARCHERS
- ROBOT BUILDERS
New desideratum: Move robot at the office!

Move to location PART. Then, move to location BOX.
Solution: Off-line programming languages

✓ No access to robot required
✓ Similar to General Purpose Languages with abstractions for Robotics
Solution: Off-line programming languages

<table>
<thead>
<tr>
<th>ROBOT COMPANY</th>
<th>PENDANT TEACHING</th>
<th>OFF-LINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FANUC</td>
<td>TP</td>
<td>KAREL</td>
</tr>
<tr>
<td>KUKA</td>
<td>KUKA Robot Language (KRL)</td>
<td>KRL</td>
</tr>
<tr>
<td>ABB</td>
<td>RAPID</td>
<td>RAPID</td>
</tr>
<tr>
<td>Universal Robots</td>
<td>URScript</td>
<td>URScript</td>
</tr>
<tr>
<td>Staubli</td>
<td>VAL3</td>
<td>VAL3</td>
</tr>
<tr>
<td>YASKAWA</td>
<td>INFORM</td>
<td>MotoCom C++ Library</td>
</tr>
<tr>
<td>DENSO</td>
<td>PacScript</td>
<td>PacScript</td>
</tr>
</tbody>
</table>

More than 30 different robot programming languages
All languages are specific to robot manufacturers.
Robotics developers need non-robot specific tools

SOFTWARE DEVELOPERS

...circa 2007

ROS

Publisher-subscriber communication middleware

Not a modeling language but an enabling platform for domain-specific applications

Source: https://robohub.org/ros-101-intro-to-the-robot-operating-system/
Development of ROS

...circa 2007

From languages specific to *robots*, to languages specific to *Robotics*. 

...circa 2012
Outline

Where are we and how did we get here?

A mosaic of models for Robot System Design
Layers of technologies for Robotic System Design

One modeling language for all?
Layers of technologies for Robotic System Design

<table>
<thead>
<tr>
<th>Speech Recognition</th>
<th>Gesture Recognition</th>
<th>Graphical User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Planning</td>
<td>Task Scheduling</td>
<td>Task Description</td>
</tr>
<tr>
<td>Deep Learning</td>
<td>Reinforcement Learning</td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td>Depth</td>
<td>LiDAR</td>
</tr>
<tr>
<td>Collision modeling</td>
<td>Vision modeling</td>
<td></td>
</tr>
<tr>
<td>Motion Planning</td>
<td>Trajectory Optimization</td>
<td></td>
</tr>
<tr>
<td>Kinematics</td>
<td>Dynamics</td>
<td></td>
</tr>
<tr>
<td>Motor Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actuation</td>
<td>Sensing</td>
<td></td>
</tr>
<tr>
<td>Human-Machine Interface Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intelligence Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perception Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>World Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motion Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigid Body Modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Modeling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Or a pandaisia* of languages?

* Pronounced *p-aa-n-dh-eh-s-EE-aa*, loosely translated as “feast”
Layers of language abstractions for Robotic System Design

Example technology: *Simulate robot motion*
Layers of language abstractions for Robotic System Design

Languages inherent to robots

Languages adapted to Robotics

Languages applied to Robotics

<table>
<thead>
<tr>
<th>Language</th>
<th>Type</th>
<th>Domain</th>
<th>Target Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot interface</td>
<td>DSL</td>
<td>KINOVA robots</td>
<td>Users</td>
</tr>
<tr>
<td>Robotics System Toolbox</td>
<td>DSL</td>
<td>Robotics</td>
<td>Developers</td>
</tr>
<tr>
<td>MATLAB</td>
<td>DSL</td>
<td>Math/Engineering</td>
<td>Researchers</td>
</tr>
<tr>
<td>C/C++</td>
<td>GPL</td>
<td></td>
<td>Builders</td>
</tr>
</tbody>
</table>
Example: Modeling robot kinematics

“Model the kinematic structure of a robot arm with 1 rigid link and 1 rotational joint”
Example: Languages for robot kinematics

GAZEBO / ROS

<robot name="myrobot">
  <link name="base">
  </link>
  <link name="my_only_link">
    <visual>
      <geometry>
        <box size="0.6 0.2 0.1"/>
      </geometry>
    </visual>
  </link>
  <joint name="my_only_joint" type="revolute">
    <parent link="base"/>
    <child link="my_only_link"/>
  </joint>
</robot>

Robotics System Toolbox

robot = rigidBodyTree;
addBody(robot, base, 'base');

my_only_link = rigidBody('my_only_link');
my_only_joint = rigidBodyJoint('my_only_joint', 'revolute');
my_only_link.Joint = my_only_joint;
addBody(robot, my_only_link, base);

✓ Textual
✓ Declarative

✓ Textual
✓ Imperative
More languages for robot kinematics

GAZEBO / ROS
Robotics System Toolbox

✓ Graphical
✓ Declarative

Simscape Multibody
More languages for robot kinematics

GAZEBO / ROS
Robotics System Toolbox
Simscape Multibody

Unreal Engine

Use third party 3D computer graphics software*
- Graphical
- Declarative

Source: https://developer.nvidia.com/isaac-sdk

* Source: Christopher Yonge – YouTube Channel
Outline

Where are we and how did we get here?

A mosaic of models for Robot System Design

Robot System Design for pick-and-place applications
Pick and place workflow

1. Build Environment
2. Move to Home Position
3. Identify Parts and Determine Where To Place Them
   - Detect Parts
   - Classify parts
4. Execute Pick-and-Place Workflow
   - Select part to be sorted
   - Pick up the object
   - Place the object
Pick and place workflow

What formalisms do we need to implement this workflow?
A pandaisia of modeling formalisms!
A pandaisia of modeling formalisms!

- A single developer must be able to switch among modeling formalisms with intuitive semantics.
- There must be mechanisms in place to integrate all modeling formalisms into one system.

What are some formalisms that enable these requirements?
A pandaisia of modeling formalisms!
Task Modeling

The task modeling language:

✓ Allows the developer to visually match their hand-drawn flowchart to syntactic states

Use Stateflow as high-level task-planner
Use Stateflow as high-level task-planner

Task Modeling

The task modeling language:

✓ Allows the developer to visually match their hand-drawn flowchart to syntactic states

✓ renders the task complexity manageable
A pandaisia of modeling formalisms!
GAZEBO is a simulation platform essential for vision and contact modeling.

Simulink is a simulation platform essential for systems modeling.

Co-simulation allows systems integration for Robotics with combination of most intuitive languages.

Simulation results are
- Synchronous
- Reproducible
Environment modeling: Co-simulation necessities

Data structure transformation
- From GAZEBO to Simulink

Composite ports
- width
- height
- data
- data_type

Multi-formalism semantics
- Integrate with MATLAB scripts for data processing
A pandaisia of modeling formalisms!
Intelligence Modeling: System Object

AI algorithms can be embedded to execution systems to be used in the real world.
Reinforcement Learning Example: Co-simulation and System Object

**Reinforcement Learning Agent as System Object**

- **Inputs**
  - Observations
  - Rewards
- **Output**
  - Action
Outline

Where are we and how did we get here?

A mosaic of models for Robot System Design

Robot System Design for pick-and-place applications

Summary
From languages specific to *robots*, to languages **specific to Robotics**.

Robotics models span **a large range of modeling formalisms**.

“Robotics is a science of integration rather than a fundamental science, and the **integration** becomes ever more complex.”

Herman Bruyninckx

Senior Software Engineer - Model-Driven Engineering

Job Summary

You will help develop our metamodelling platform, which currently backs significant pieces of our Simulink and Stateflow graphical languages. Our metamodelling infrastructure is fairly sophisticated and is seeing increased in-house usage. We need to support our clients with advanced run-time services such as inter-model synchronization, and database backing. We have a vision of a modeled backbone for our entire product line. Come help make it happen!

Responsibilities

You will work on code generation, develop modeling tools (graphical and textual), and transform ad-hoc data structures into modeled ones. You will be working right at the intersection of theory and practice. You will need to evangelize for data modeling generally and put it into place specifically. You will be responsible for having large ideas and carrying them through to shipment.

Minimum Qualifications

- A bachelor’s degree and 7 years of professional work experience (or a master’s degree and 5 years of professional work experience, or a PhD degree) is required.
Senior Software Engineer - Model-Driven Engineering

Job Summary

Additional Qualifications

- Excellent C++, or Java, or JavaScript skills
- Experience with MOF/UML a big plus
- Experience with large codebases and build systems
- Experience with databases a plus
- Interest or experience in JavaScript a plus
- Strong abstract thinking coupled with a drive to produce software
- Masters, or PhD, in Computer Science, Electrical/Computer Engineering, Mathematics, Physics, or related engineering field

Minimum Qualifications

- A bachelor’s degree and 7 years of professional work experience (or a master’s degree and 5 years of professional work experience, or a PhD degree) is required
Can I order a “Robotics-specific modeling language”?

Pick…

1. …. a target group
   - Robot Users
   - Software Developers
   - Robotics Researchers
   - Robot Builders

2. …. a technology
   - Speech Recognition
   - Task Scheduling
   - Deep Learning
   - Collision modeling
   - Motion Planning
   - Dynamics
   - Actuation
   - Gesture Recognition
   - Task Description
   - Reinforcement Learning
   - Vision modeling
   - Trajectory Optimization
   - Kinematics
   - Sensing

3. …. an abstraction
   - Languages inherent to robots
   - Languages adapted to Robotics
   - Languages applied to Robotics

4. …. a target
   - ROS node
   - Embedded MCU
A mosaic of models and languages
A mosaic of models and languages!
Thank you!