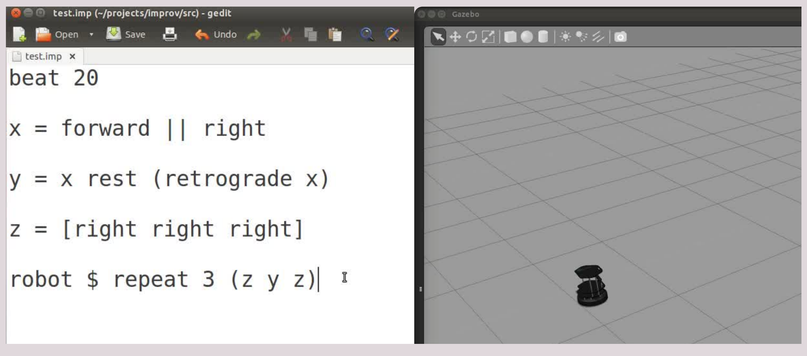
Improv: Live Coding for Robot Motion Design

Alexandra Q. Nilles[1], Chase Gladish[1], Mattox Beckman[1], Amy LaViers[2]
[1]Department of Computer Science, University of Illinois, Urbana-Champaign
[2]Mechanical Science and Engineering Department, University of Illinois, Urbana-Champaign

Motivation

- Most languages for creating robot motion are very powerful, but cannot describe movement concisely, and can be intimidating for robotics newcomers
- Choreographers and movement studies experts have developed abstractions for describing movement - can we improve robot programming with this expertise?
- "Live coding" has become a popular way to creatively, performatively generate music and visuals - what about embodied motion?
- Tools for prototyping robot movement could be useful for performance, education, researchers, and industrial automation.

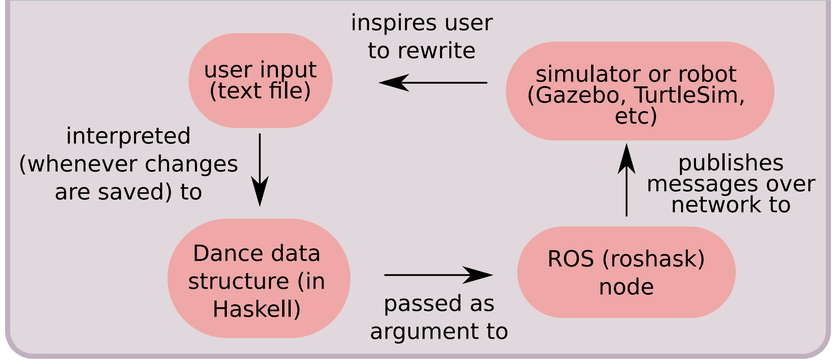
Example Interface



Design Principles

- minimize "representational distance" between intended movement and code
- rapid movement prototyping
- workspace with few attentional switches

Information Flow



Expressive Building Blocks

Combining Movements

move forward for one beat, turn right for one beat, move forward for one beat

```
forward right forward
```

move forward, right, and forward, all in one beat

```
[forward right forward]
```

move in a curve, forward and right

```
forward || right
```

Transforming Movements

move forward four times

```
repeat 4 forward
```

do movement `x`, reflected across saggital plane

```
reflect YZ x
```

reverse "forward right left"

```
reverse (forward right left)
```

retrograde "forward right left"

```
retrograde (forward right left)
```

Comparing Representations

ROS Program in Python

```
if __name__ == '__main__':
  pub = rospy.Publisher(
    'turtle1/cmd_vel',Twist)
  rospy.init_node('publisher_node')
  loop_rate = rospy.Rate(5)
  while not rospy.is_shutdown():
    vel=Twist()
    vel.linear.x = 1.0
    vel.angular.z = 1.0
    pub.publish(vel)
    loop_rate.sleep()
```

Equivalent Program in Improv

```
turtle1 $ forward || left
```

Future Work

- More articulated robots: how to represent in platform-invariant way?
- Robot-robot and robot-environment interaction: `approach`, `landmarks`, conditional expressions
- User studies:
  - How does percieved usability depend on programming experience?
  - Does the live interface enable creativity? What is the effect of delay?

Acknowledgments

Funding: This work is partially funded by NSF grant #1328018 and DARPA grant #D16AP00001.