

RoLL (Robot Learning Language)

Towards High-performance Plan-based Robot Controllers that Learn



Alexandra Kirsch, Michael Beetz

kirsch@in.tum.de

<http://www9.in.tum.de/people/kirsch/>

For planning in the domain of autonomous robots, abstraction of state and actions is indispensable. This abstraction however comes at the cost of suboptimal execution, because relevant information is ignored. A solution is to maintain abstractions for planning, but to fill in precise information on the level of execution. To do so, the control program needs models of its own behavior. These models, as well as the control routines themselves, should be learned automatically. Unfortunately, the performance of learned routines often drags substantially behind those of programmed ones, at least for complex, interacting, and dynamically changing tasks. In our opinion this is not due to a poor performance of learning algorithms, but to the insufficient integration of learning into control languages. We develop the robot control and plan language RoLL, which provides mechanisms for representing state variables, goals and actions, and integrating learning into the controller.

Aim

Development of high-performance robot controllers that

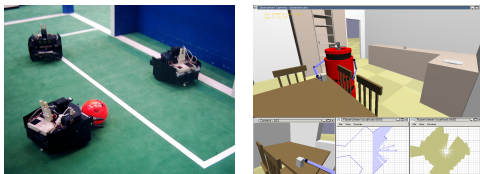
- adapt themselves to new situations,
- use models of their abilities and the environment,
- acquire sets of heterogeneous skills at different levels.

Approach

- model-based representation of the physical world
- explicit representation of abstract modalities like state variables, procedures, and goals
- synergetic interaction of programming and learning

Application

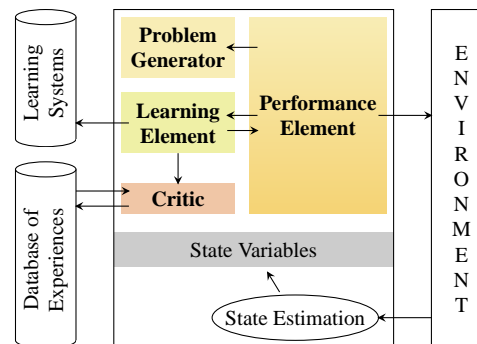
- autonomous robot soccer
- simulated household robot



RoLL's Learning Tools

RoLL provides a parameterizable execution model for plans that contains functional components for learning. Parameters are defined by RoLL specifications.

RoLL Execution Model



RoLL Specifications

1. get raw experiences

```
define raw experiences bezier-raw-exp
while achieving goal exp-collection-goal
using problem generator bezier-generator
select *.value, *.goal-value
from position-sv, episode.target
in interval [episode.start, episode.end]
sample period 100ms
```

2. transform experiences to facilitate learning

```
define abstract experiences bezier-exp
using raw experiences bezier-raw-exp
select bezier-param-0, bezier-param-1
from clustered-filtered-experiences
where minimum(deltaTime)
```

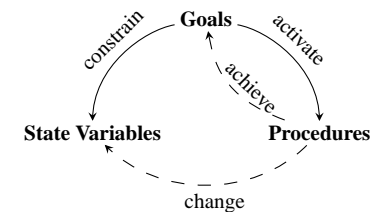
3. perform learning

```
define learning problem for routine navigation
experiences bezier-exp
learning system neural net
learning bias hidden units=3, function=backprop
```

4. embed learned routine into controller

RoLL Performance Elements

RoLL uses state variables, goals and procedures for specifying competent robot behavior.



State Variables

- represent physical quantities
- have physical dimensions, units, and ranges
- can be *observable*, *derived*, and *controllable*

```
def-observable-sv position
slots ~x unit=m range=[-6m, 6m]
~y unit=m range=[-4m, 4m]
history=(length:3, sampling:0)
```

Goals

- are constraints over controllable state variables
- arbitrate for the best routine in current situation
- monitor procedure execution

```
declareGoal reach-pose (position, precision, weights)
specification  $\exists t : t > t_0$ 
 $\wedge \text{dist}(\text{pos}(t), \text{pos}) < \text{precision.dist}$ 
available-routines = reach-position
objective =  $\frac{\text{weights.time}}{t-t_0} + \frac{\text{weights.dist}}{\text{dist}(\text{pos}(t), \text{pos})}$ 
```

Procedures

- any kind of procedure (functions, control routines)
- can be learned or programmed
- provide (learned) models of their behavior

```
declare routine navigation
superclass low-level-routine
fulfills goal robot-at-position
status to-be-learned
```

Progress

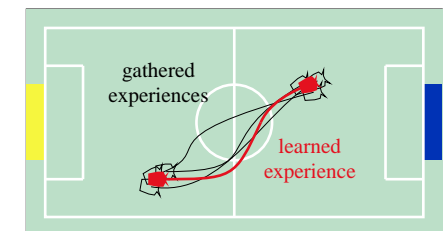
Our language is implemented on top of the robot control and plan language RPL.

Model-based constructs

- rational reconstruction of the autonomous robot soccer controller
- control of soccer robots in world championship 2004
- control of simulated kitchen robot

Learning

- learning problems for soccer robots
- produced navigation routines that reach the level of programmed ones
- used in practical courses
- coupled to MySQL database system
- uses WEKA data mining tool



References

- [1] M. Beetz, A. Kirsch, and A. Müller. RPL-LEARN: Extending an autonomous robot control language to perform experience-based learning. In *3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS)*, 2004.
- [2] A. Müller, A. Kirsch, and M. Beetz. Object-oriented model-based extensions of robot control languages. In *27th German Conference on Artificial Intelligence*, 2004.