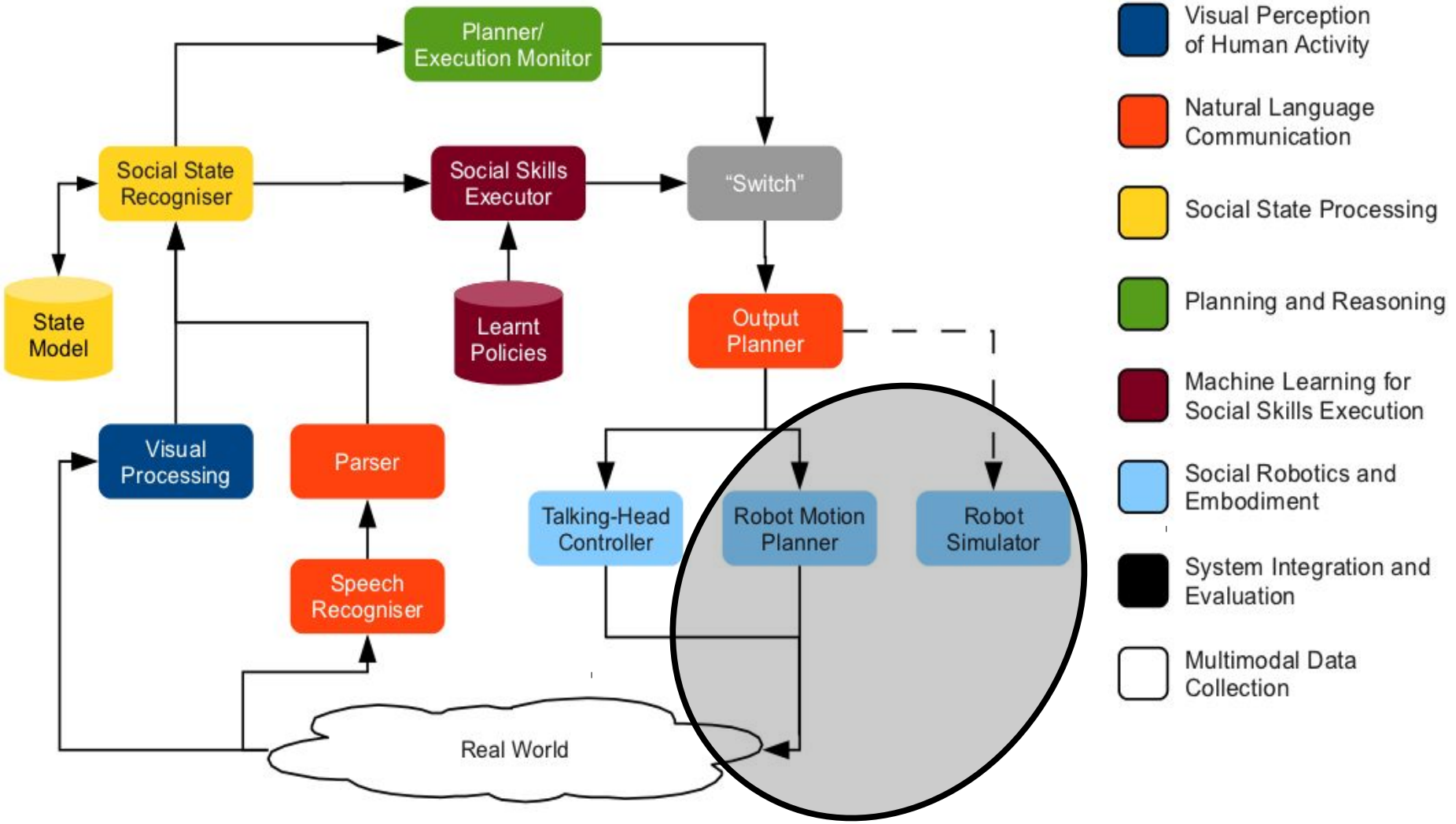
A Software Architecture for Robot Control and its Application to Social Robotics

Andre Gaschler

Tutorial ”Joint Action for Social Robotics”
International Conference on Social Robotics
Amsterdam, Netherlands
2011-11-23

Scope


Planner/
Execution Monitor
Social State
Recogniser
Social Skills
Executor
"Switch"
State
Model
Learnt
Policies
Output
Planner
Visual
Processing
Parser
Talking-Head
Controller
Robot Motion
Planner
Robot
Simulator
Speech
Recogniser
Real World
Visual Perception
of Human Activity
Natural Language
Communication
Social State Processing
Planning and Reasoning
Machine Learning for
Social Skills Execution
Social Robotics and
Embodiment
System Integration and
Evaluation
Multimodal Data
Collection

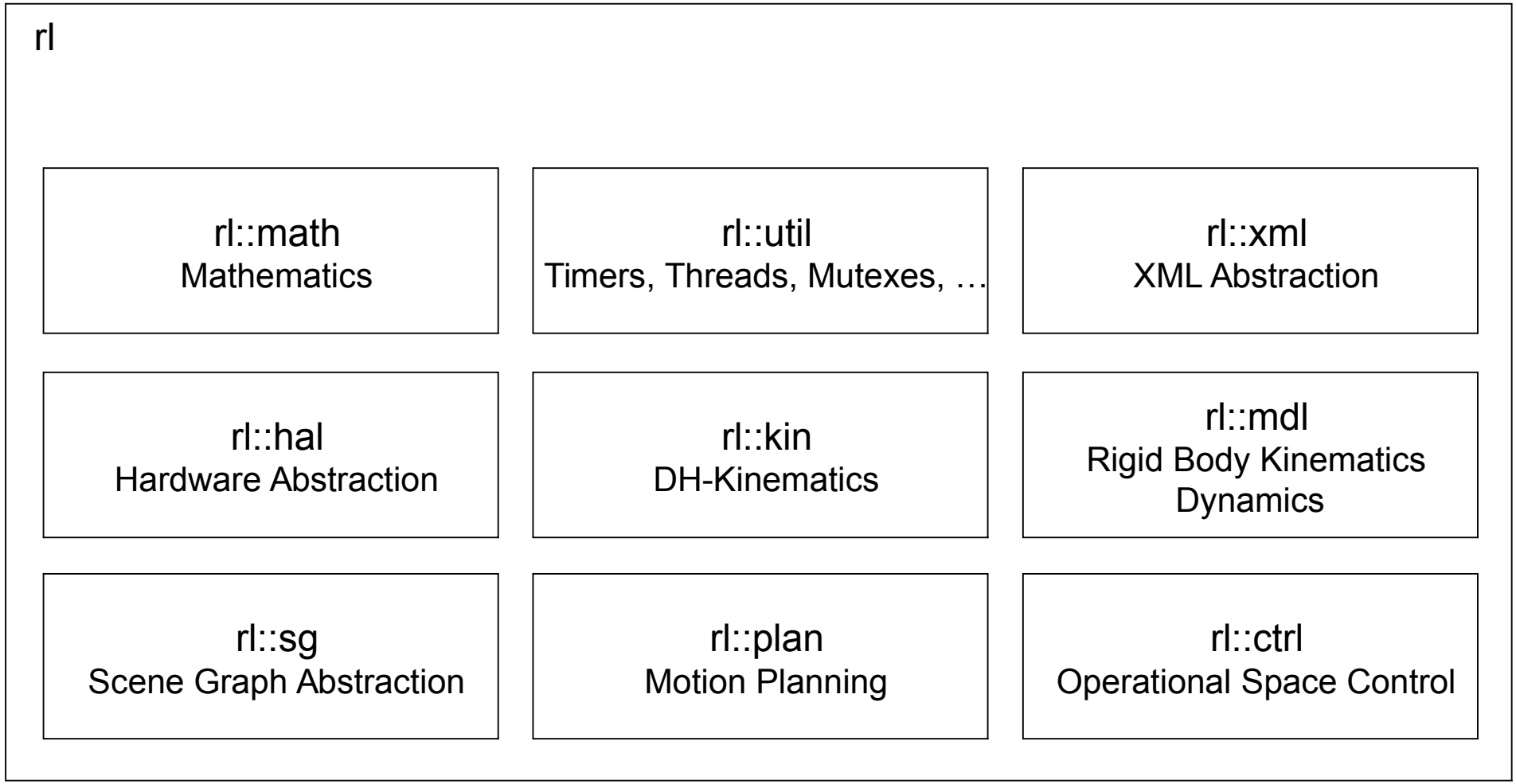
Robotics Library - Introduction

- Robotics Library
  - in short: RL
  - was developed mainly to support HRI EU projects JAST and JAHIR, more currently also JAMES
  - completely open source
  - BSD licensed, also free for commercial usage
  - http://www.roboticslibrary.org/

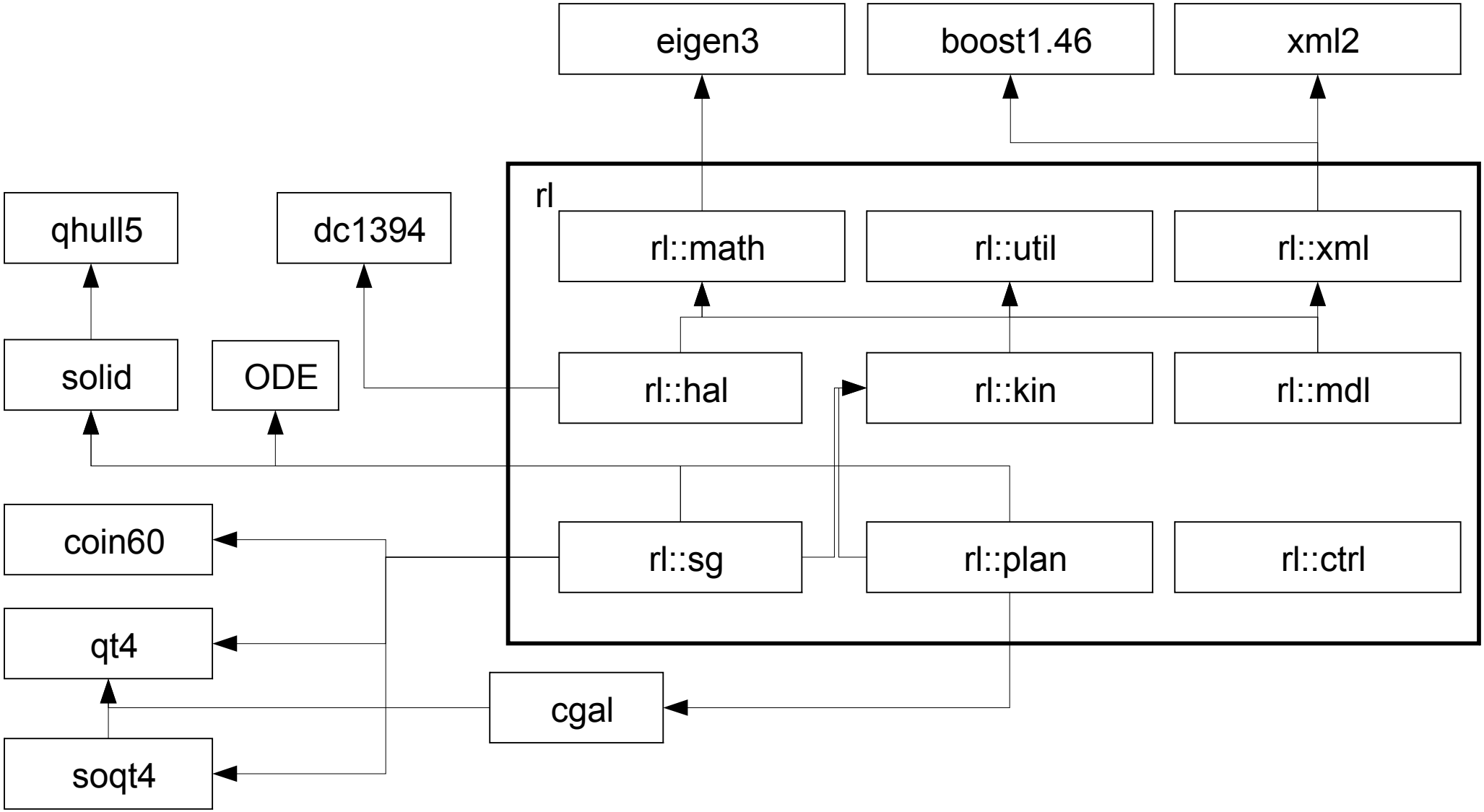
Motivation and Scope

- Goal: Provide a fundamental set of robotics functions
  - keep it simple, i.e. a library, not a middleware
  - abstract from common robotics hardware
- More focussed, complete and consistent than other robotics libraries
  - single focus on a library (compared to ROS)
  - compatible data structures from basic math over hardware abstraction to scene visualization
  - complete implementation of all available functions

Components of RL



Dependencies of RL

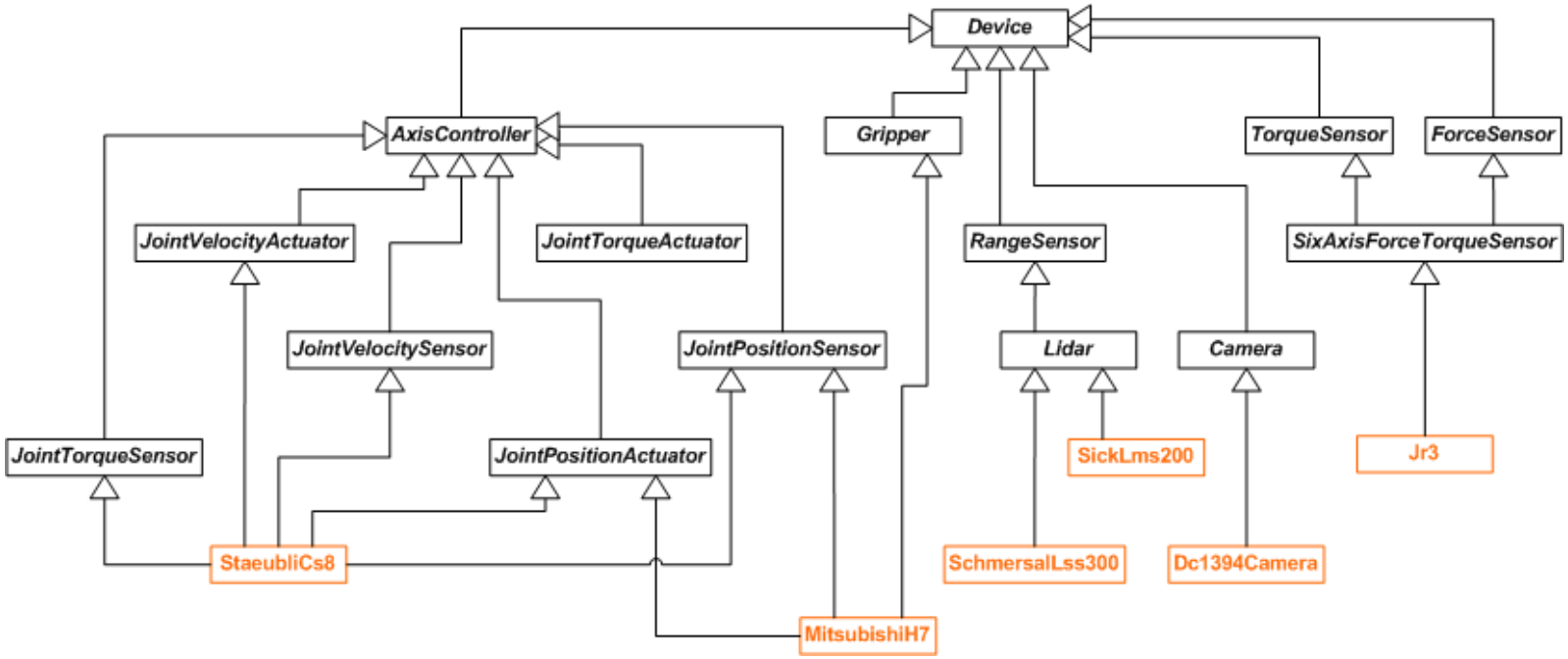


Basic Components of RL

- rl::math
  - Matrix and vector types based on Eigen3
  - Transformation and Rotation
  - Quaternions
  - Polynomials
  - Kalman filter
- rl::util
  - Timers
  - Threads
  - Mutexes and semaphores
- rl::xml
  - XML abstraction

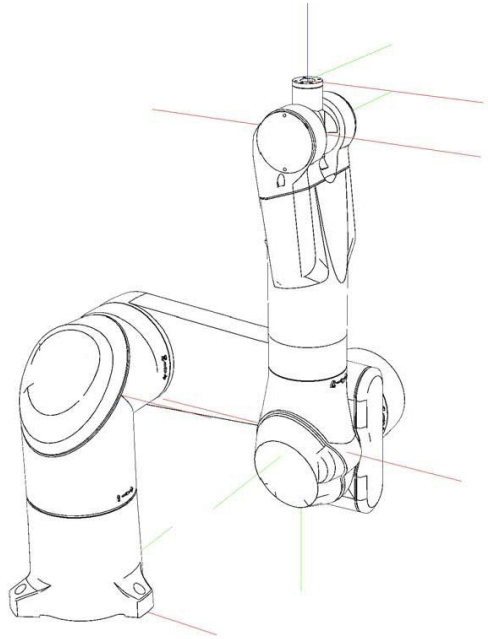
Hardware Abstraction Layer

- rl::hal
  - Socket and serial communication
  - Common robot controllers
  - Common laser range sensors and cameras



Denavit-Hartenberg Kinematics

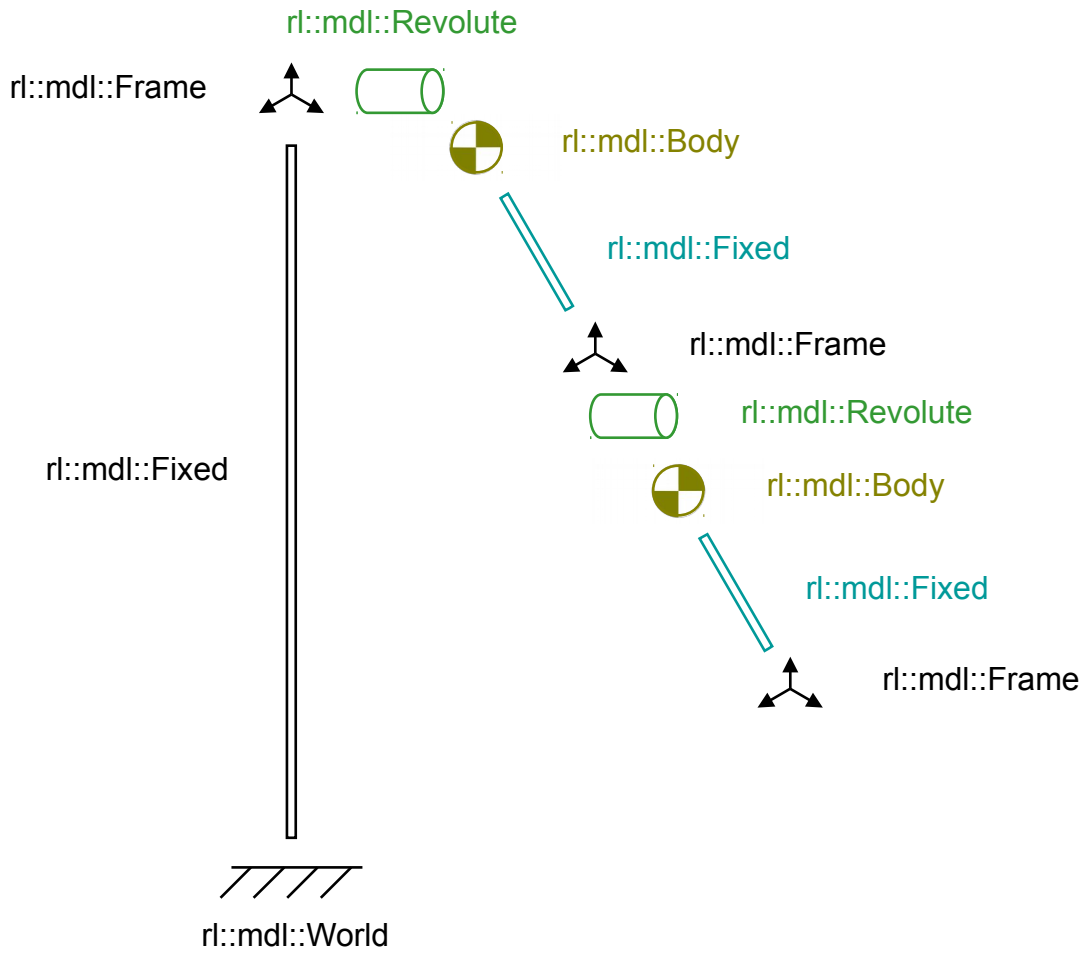
- rl::kin
  - Forward kinematics
  - Algebraic inverse kinematics for most 6 DoF robots
    - Stäubli, Mitsubishi, Kuka...
    - Handedness of solution is preserved
  - Analytical inverse kinematics
  - Jacobian and inverse jacobian
  - Manipulability measure



d	*θ*	*a*	*α*
0.375	0	0	-90
0	-90	0.4	0
0.02	-90	0	-90
0.45	0	0	90
0	180	0	90
0.07	0	0	0

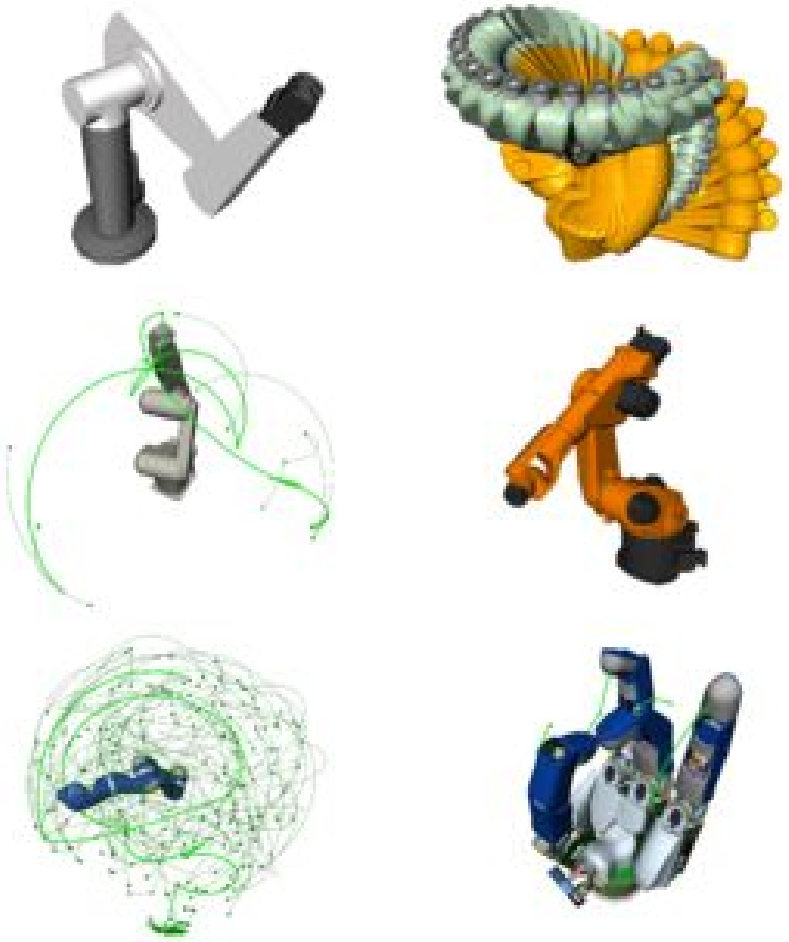
Rigid Body Dynamics

- rl::mdl
  - Kinematics and dynamics of generalized models of revolute and prismatic joints
  - Recursive Newton-Euler



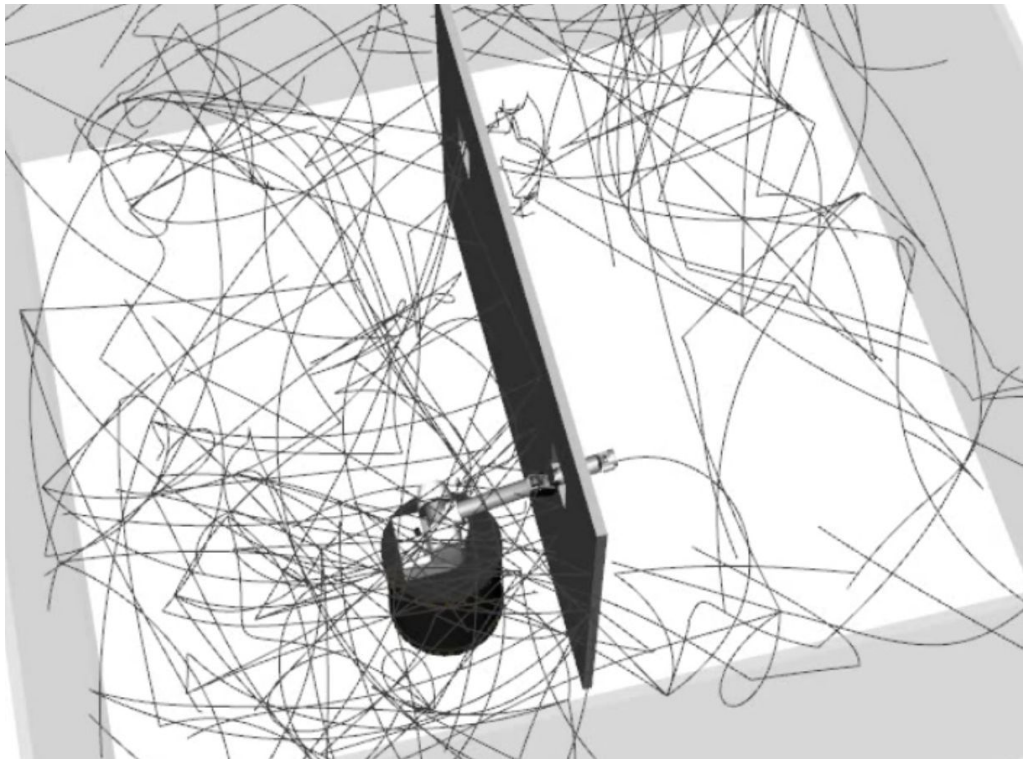
Scene Graph

- rl::sg
  - makes use of the Open Inventor over OpenGL with the help of Coin3D
  - models are described in the VRML format



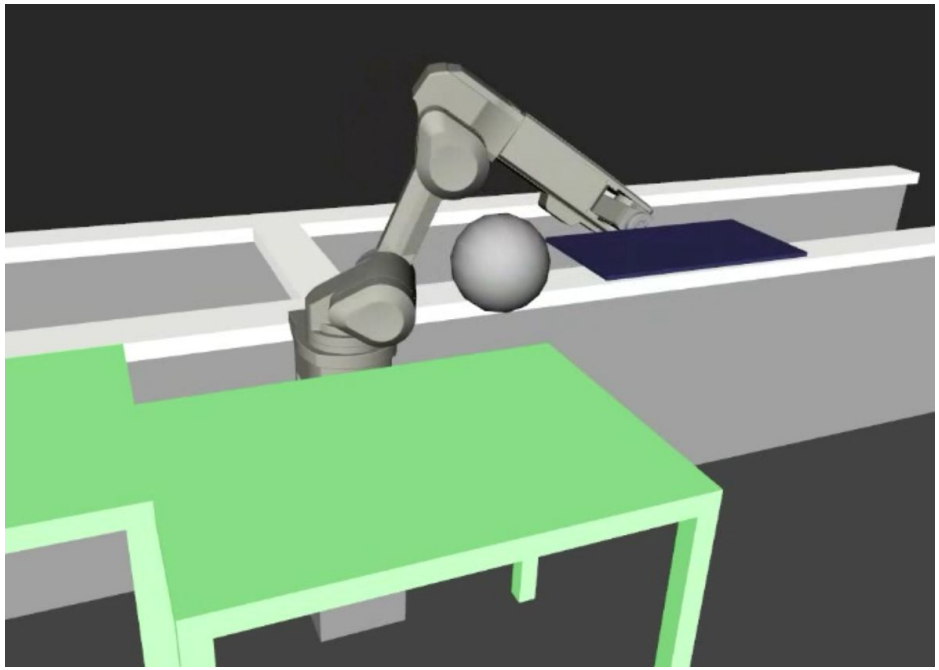
Motion Planning

- rl::plan
  - Rapidly-exploring random trees (RRTs)
  - Probabilistic roadmaps



Operational Space Control

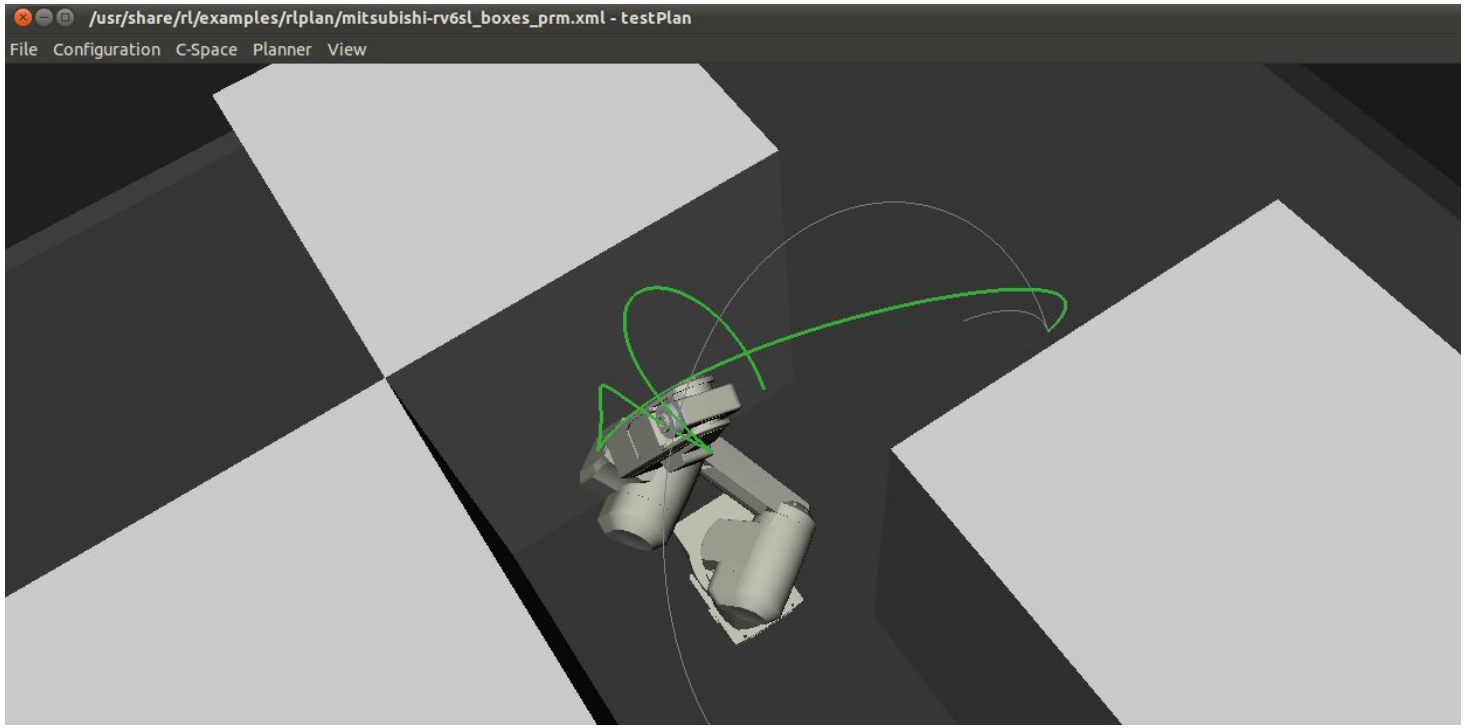
- rl::ctrl
  - Actions consisting of prioritized tasks
  - Tasks
    - Position
    - Collision Avoidance
    - Posture
  - Work in progress



$$T_{\text{Orientation}} \triangleleft T_{\text{Avoidance}} \triangleleft T_{\text{Position}} \triangleleft T_{\text{Posture}}$$

Demo

Robot Motion Planning with difficult Obstacle Avoidance



Getting the Robotics Library

- For Ubuntu 10.04 and newer (or Debian)
  - Add our repository to your trusted repositories
    ```
    sudo apt-add-repository ppa:roblib/ppa
    sudo apt-get update
    ```
  - Install and try out examples
    ```
    sudo apt-get install librl-examples
    /usr/share/rl/examples/examplePlan1.sh
    ```
  - Install the developer libraries
    ```
    sudo apt-get install librlkin-dev librlsg-dev
    ```
- For Windows
  - We can provide you with binaries, esp. for dependencies
  - You can also (re-)compile from source

Learn more

- Visit the Robotics Library website

  http://www.roboticslibrary.org
- Have a look into the source code

  http://www.roboticslibrary.org/download
- Ask questions to the mailing list

  roblib-developers (at) lists.sourceforge.net
- Ask me

  Andre Gaschler: gaschler (at) fortiss.org
- or the main developer

  Dr. Markus Rickert: rickert (at) fortiss.org

References

- Markus Rickert. Efficient Motion Planning for Intuitive Task Execution in Modular Manipulation Systems. Dissertation, Technische Universität München, Munich, Germany, 2011.
- Markus Rickert, Alois Knoll. Robotics Library. ICRA Workshop on Open Source Software in Robotics, 2009.
- Claus Lenz, Markus Rickert, Giorgio Panin, and Alois Knoll. Constraint task-based control in industrial settings. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3058-3063, St. Louis, MO, USA, October 2009.
- N. N. http://www.roboticslibrary.org, accessed 01 Apr 2014.