Model Based Robot Software Development

Jakob Schwendner
DFKI Bremen & Universität Bremen
Robotics Innovation Center
www.dfki.de/robotics
robotics@dfki.de
Motivation

- Robot complexity and task complexity is growing
- Costly development
- How to handle the complexity?
  - Modules
  - Models

Modules allow reuse – Models say how
Robot Development

• Process should be
  ▪ **Structured** – development process
  ▪ **Reproducible** – design & behaviour
  ▪ **Scalable** – in system and development complexity
  ▪ **Dynamic** – on-line reconfigurability

• Three different views
  ▪ Behaviour
  ▪ Hardware
  ▪ Software
Robot Development - Behaviour

- Process should be
  - Structured
  - Reproducible
  - Scalable
  - Dynamic

- Three different views
  - Behaviour
  - Hardware
  - Software
Robot Development - Hardware

- Process should be
  - Structured
  - Reproducible
  - Scalable
  - Dynamic

- Three different views
  - Behaviour
  - Hardware
  - Software
Robot Development - Software

- Process should be
  - Structured
  - Reproducible
  - Scalable
  - Dynamic

- Three different views
  - Behaviour
  - Hardware
  - Software
DROCK Project

Goal: Design and implementation of a framework and tools for robot programming.

- Objectives:
  - Modelling of robotic structure
  - Mapping of tasks onto execution units
  - Execution and monitoring with dynamic reconfiguration
  - Tools to support the development
  - Evaluation of toolset in demo scenario and user study
Challenges

- Modelling
  - Consistent models
  - Practical focus on robotic problems
- Mapping
  - Requires robot specific heuristics
  - Deployment on heterogeneous units (CPU, uC, FPGA)
- Execution
  - Monitoring in heterogeneous environment
  - Decision on reconfiguration
  - Distributed supervision
Tools

Modelling Tools → Behaviour Models → Execution Framework
Modelling Tools → Software Models
Modelling Tools → Hardware Models

Behaviour
Software
Hardware

Design Time → Run Time
State of the Art

- ROS (Robot Operating System)
  - Implicit component model
  - Low-resistance workflow – ad-hoc development
  - Topic based communication
- Orocos RTT
  - Implicit component model
  - Point to point connections
  - Real-time capable
- GenoM
  - Explicit component model
  - Capable of generating Behaviour Interaction Priority (BIP)
- Rock
  - Explicit component and network model
  - Orocos RTT as Middleware
  - Dynamic run-time reconfiguration
Robot Construction Kit (ROCK)

- www.rock-robotics.org
- rock.opendfki.de
- github

- Development started in 2008
- Running on more than 12 Robots at DFKI/RIC
- In use by ESA ESTEC, Space Applications, Marum, Geomar, etc.
- 199+ library packages (internal + external)
- 136+ component packages
- 80 messages / month on rock-dev@dfki.de
- 30 active members on mailing list in last 6 months
Industrial Applications

- Industrial robotics will change
- Complex tasks – human shared environments
- Model based approach
  - Robust
    - Generic instantiation
    - Generic failure handling
  - Verification
    - Component level testing
    - Correct by construction
    - Model inference
- Industrial Use
  - Space
  - Off-shore
  - Human shared

NASA / JPL

Oceaneering

BMW AG