

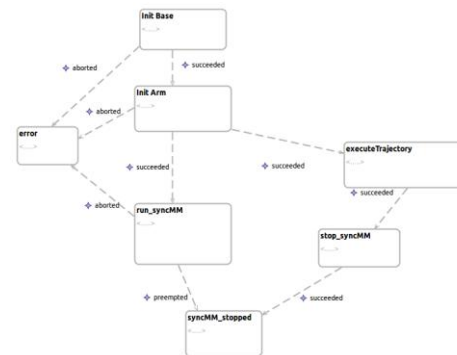
**MODEL DRIVEN ENGINEERING:
A WAY TO HANDLE COMPLEXITY, QUALITY AND
PROCESSES IN COMPONENT BASED ROBOT SYSTEMS**

Alexander Bubeck

Model driven engineering: A Way to Handle Complexity, Quality and Processes in Component Based Robot Systems

ROS INDUSTRIAL – AN ENABLER FOR INDUSTRIAL ROBOTICS? – October 17, 2012

Alexander Bubeck



Introduction

ROS in a nutshell

- Component based communication framework for robot applications
 - Distributed
 - Programming language independent
- Repository of open source robotic software
 - Covering e.g. manipulation, mobility, object perception and control
- Development tool chain for robotics
 - Build system with dependency management
 - Debugging and runtime analysis
 - Visualization and simulation tools
- Widely adopted in robotics research community



Current challenges in the ROS community

Software quality and reuse

- **High complexity**
 - Currently over 2000 ROS package and libraries
 - Broad area of development activities (from algorithms to system integration)
- **Little reuse** → lots of duplicates
 - Possible reasons:
 - Interfaces are not standardized (only some pseudo standards)
 - Code quality varies → high freedom of implementation
 - Few indicators for quality of certain ROS packages

 Model driven engineering to disseminate standards and proven concepts

Software Engineering concepts for robotics

Overview

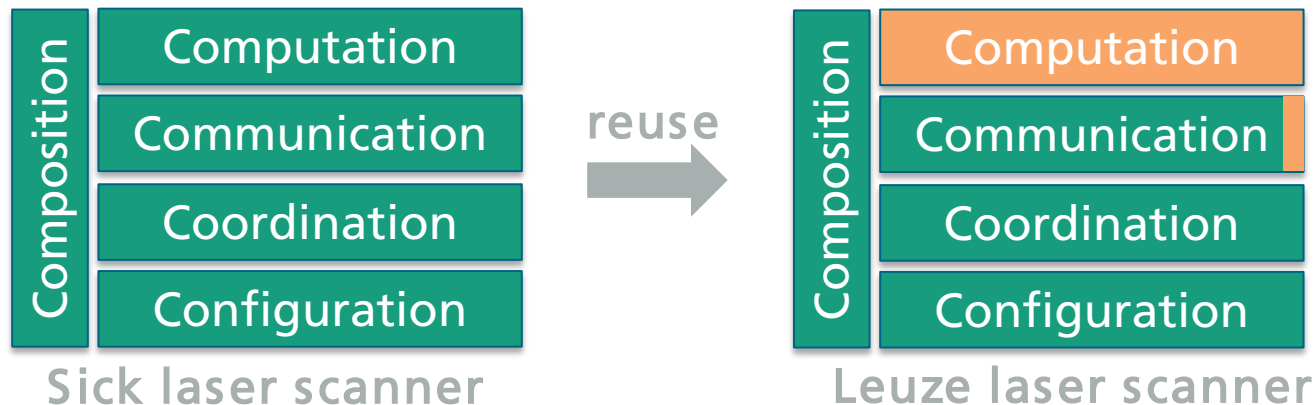


- BRICS EU project
 - Structure and formalization of the robot development process
 - Providing tools, models, and functional libraries, which help accelerating this process significantly.
- Best practice concepts in Software Engineering that can improve quality and reuse in robotics:
 - Separation of Concerns
 - Separation of User roles
 - Separation of platform-dependent Code

Software engineering in robotics

Separation of concerns

- Behavior of components correspond to specific concerns of a developer
- Concerns for software components:



- Reuse of software means exchange or modification of specific concerns

➡ Separation of concerns improves reuse

Software engineering in robotics

Separation of user roles

- During development of robotic applications multiple process steps are passed (see in e.g. ISO/IEC 12207)
- Simplified subset of this process in robotics is the following



- Each step represents another user role that differs among others regarding:
 - Knowledge of the user
 - Scopes of the component concerns for the user

➡ Separate user roles according to the development process

Software engineering in robotics

Separation of code

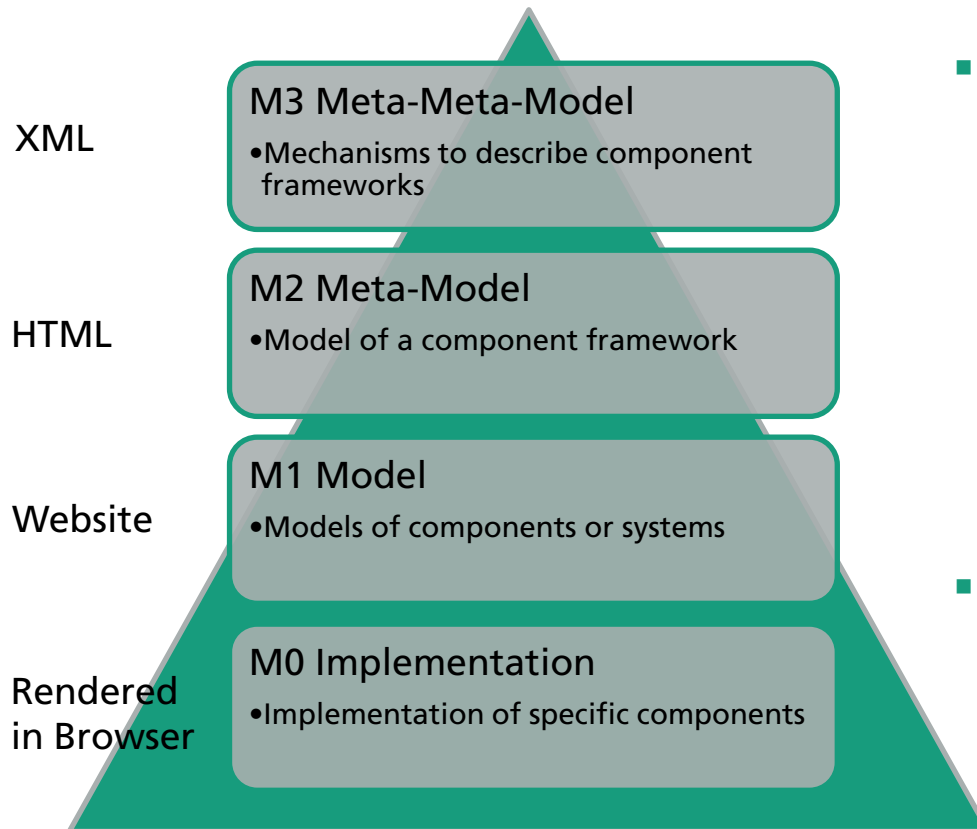
„A good way to stay flexible is to write less code.“

- Often repeated tasks during ROS development:
 - Creation of application skeleton
 - Implementation of communication and configuration
 - Documentation of communication and configuration
 - Implementation of internal coordination of ROS node
- Separation between platform-dependent code and independent code possible

 Auto-generation of code for standardization and better quality

Model driven engineering

Overview

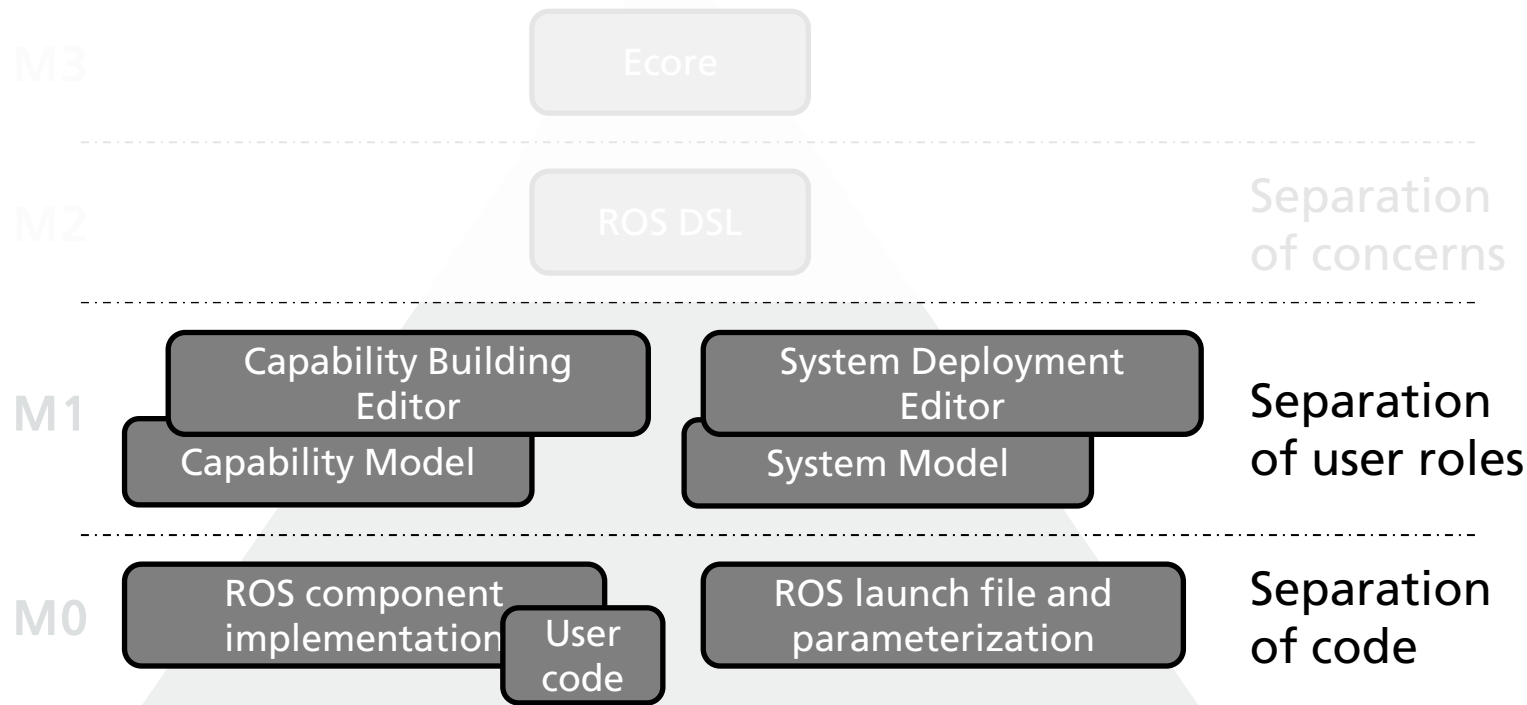


- Development and use of domain models to represent abstract knowledge to
 - Encapsulate complexity
 - Differentiate user roles
 - Enforce architectures
 - Support reuse
- Object management group defines multi-layer architecture

Model driven engineering

Application to ROS using Eclipse

- Eclipse Modeling Project provides a tool chain for creating domain specific languages (DSL's) and model editors:



BRIDE – An Eclipse based MDE-toolchain for ROS

Overview

The BRICS IDE (BRIDE) is an Eclipse distribution including:

- Graphical model editors for creating ROS packages and ROS systems



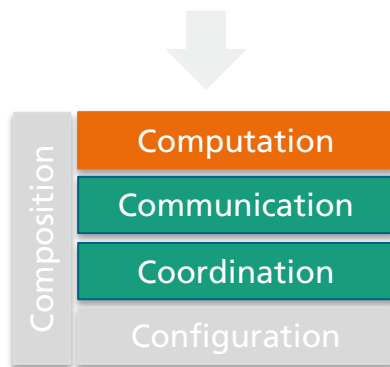
- Automatic generation of
 - C++ or Python ROS nodes
 - Launch files configuring ROS topics, services and parameters
- Separation of ROS independent user code and ROS skeletons
- Integration with Eclipse development environments for C++ and python (CDT and pydev)

BRIDE – An Eclipse based MDE-toolchain for ROS

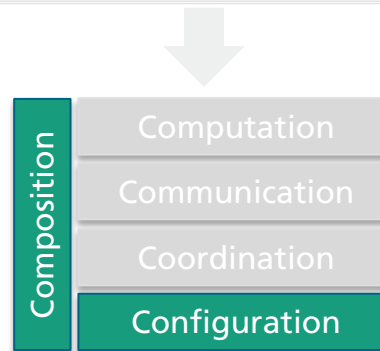
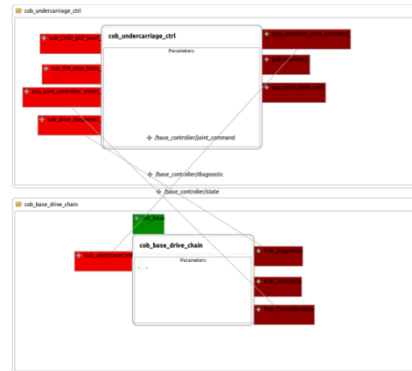
Work flow of model driven engineering

- Model driven engineering work flow in practice

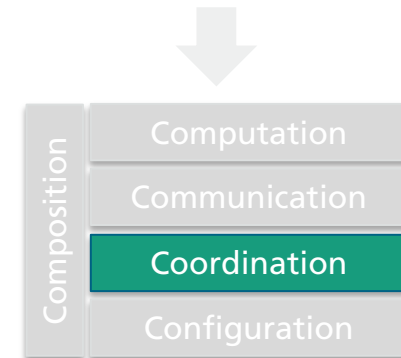
Capability building



System deployment



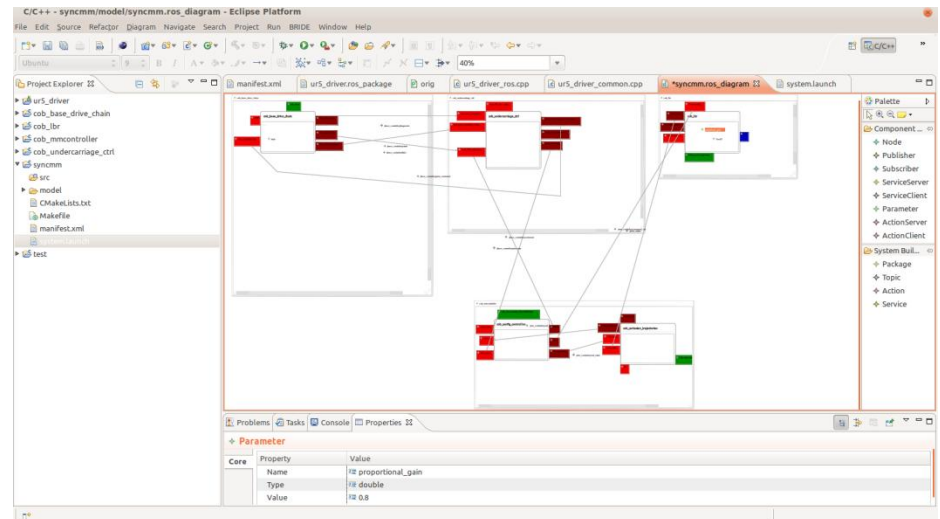
Application building



BRIDE – An Eclipse based MDE-toolchain for ROS

Evaluation

- Evaluations of BRIDE showed that between ~10% and ~50% of component code can be auto-generated based on the models
- Components implemented in BRIDE can be transferred to other frameworks (e.g. OROCOS)
- Transparent system deployment with explicit composition and configuration



Summary

Model driven engineering in robotics

- Separation of concerns, user roles and code can improve reusability and quality of software components
- BRIDE, a model driven engineering tool chain, disseminates these Software Engineering techniques in the ROS community
- Model driven approach can be transferred to other domains
 - Support of workflow of specific manufacturing processes
 - Teach-In processes for service and industrial robots

BRIDE for ROS:
<http://github.com/abubeck/BRIDE>

Contact:
Alexander Bubeck
alexander.bubeck@ipa.fraunhofer.de
Tel: +49 711 970 1314

Acknowledgement to the project FP7-ICT-
231940-BRICS (Best Practice in Robotics)
www.best-of-robotics.org

ROS INDUSTRIAL – AN ENABLER FOR INDUSTRIAL ROBOTICS?

DEVELOPMENT OF INDUSTRIAL APPLICATIONS WITH ROS – EXPERIENCES

