ReApp: Ontology-Driven Tools for Robot Application Development

Björn Kahl
Fraunhofer IPA
E-Mail: bjoern.kahl@ipa.fraunhofer.de
ReApp in a nutshell

Robot (software) development is hard

Let’s make it easier
(again)

• By semantically annotating and standardizing components & interfaces
• Allowing graphically assisted composition and configuration of solutions
Technologie-Stack

Applications

Development tools, kinematics, perception, visualization, etc

Networking, motion control, engineering data, etc

Hardware

© Fraunhofer IPA
ReApp Architecture

Cloud-layer

Hardware
Software

App-Repository

Test Framework

Virtual Deployment Environment

Virtual Deployment Environment

Development (model) layer

HW Component Developer: Create HW Wrappers
SW Component Developers: Create SW Components and Skills
System Integrators: Create Skills and Solutions
End Users: Create/Adapt Solutions

Find, Create, Annotate, Test and Release Software Components
Compose Skills (from Components)
Create and Deploy Solutions

ReApp Engineering Workbench

Systems layer (Execution Environment)

ROS Industrial

© Fraunhofer IPA
ReApp Domain Ontology

Base Ontology

Component Classification
- e.g: Sensor with Dimension 2D and Output Format PointCloud → Laser Scanner

Inference of new Knowledge
- Inference of Capabilities, Attributes, ...

Hardware Ontology
- Has Hardware Type

Software Ontology
- Has Software Type

Capability Ontology
- Has Capability
- Has Attribute
- Has ROS Interfaces

Multi-Criteria-Search
- Capabilities
- Attributes
- ROS-Interfaces
ReApp attributes and parts library
Adding type infers capabilities etc.
ReApp Workbench (Development Environment) Composition of Apps to Solutions

Skill Editor

App Explorer (store)

App Explorer (local)

ECC Editor (Control flow)

Property View
Questions?