



Supported by:



Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag



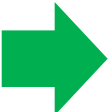
Workshop: Hybrid Production Systems

Introduction

Martin Naumann



Fraunhofer IPA
Nobelstr. 12
70569 Stuttgart
Germany
Tel.: +49 711 970-1291
E-Mail: martin.naumann@ipa.fraunhofer.de

- Continuation of the dialogue started in ERF2014
 - Presentation of first results from FiaD, LIAA, ROBO-Partner and ReApp
 - Exploration of possible synergy effects
-  Initiate activities with regard to developing a
Reference Architecture for a Hybrid Production System

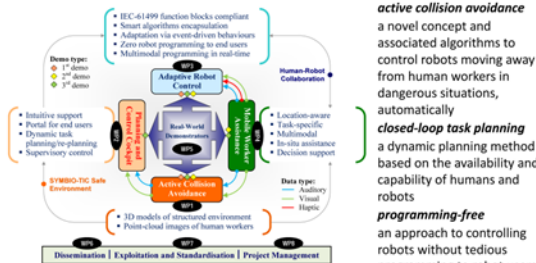
Welcome of 3 new FoF projects in the cluster

- Domain 5: Human-centred Manufacturing
 - Area 2: Human Robot Interaction



Lower costs, increased safety, better working conditions and higher profitability

SYMBIO-TIC is aimed at a novel hybrid assembly/packaging ecosystem in dynamic factory environment based on human-robot collaboration. The system will be context-aware in task planning and execution, safe to human workers in a shared fenceless working space, flexible and adaptive to dynamic changes, and cost effective.



Objectives

- To develop an active collision avoidance subsystem to safeguard human workers
- To generate adaptive task plans appropriate to both robots and human workers
- To adapt to dynamic changes with intuitive and multimodal programming
- To provide human workers with in-situ assistance on what-to-do and how-to-do
- To demonstrate and validate the project concept and solutions.

Demonstrators

- A food-packaging demonstrator by robotomation GmbH in Germany (a system integrator);
- An aeronautic component assembly demonstrator by Acturi in Spain (an end-user); and
- An automotive engine assembly demonstrator by Volvo Car Corporation in Sweden (an end-user).

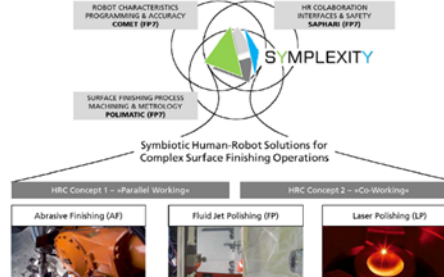
SYMBIO-TIC has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 637107
For more information visit: www.symbio-tic.eu



Symbiotic Human-Robot Solutions for Complex Surface Finishing Operations
SYMPLEXITY

Brilliance through Human-Robot-Collaboration

In almost every sector of industrial manufacturing polishing techniques are applied. Nonetheless most processes are performed manually as the required tasks partially demand high cognitive effort as well as motoric flexibility. SYMPLEXITY is closing the gap between the automated respectively manual polishing of simple respectively complex geometries by creating a safe environment of collaboration between the robot and the human worker.



Objectives

- Accurate and cognitive industrial robot systems enabling safe human-robot collaboration for surface finishing operations
- Easy to use interfaces for planning, control and re-planning of shared finishing tasks
- Collaboration oriented process technology for abrasive finishing, laser and fluid jet polishing
- Integrated and autonomous sensing system for objective identification of surface properties
- Introduction of developed collaborative finishing solutions into manufacturing industry

Evaluation

Due to the broad range of applied manufacturing technologies in the SYMPLEXITY project, different demonstrators from a variety of end users with different areas of application are processed, this covers parts from the

- medical sector
- automotive industry
- aeronautic sector
- tool and die making industry

SYMPLEXITY has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 637080
For more information visit: www.symplexity.eu



Highly customizable robotic solutions for effective and safe human robot collaboration in manufacturing applications
FOURBYTHREE

Can you imagine humans and robots working together in a fenceless industrial environment?

FourByThree is an alliance made up of 15 European partners, coordinated by IK4-TEKNIKER. The project started in December 2014 and will last for 36 months.



"We had some developments on series elastic actuators and thought they could be used for building a robotic manipulator for safe human-robot collaboration. We are bringing decades of research together into one single product."

José de Gea
Fernández (DFKI)

FourByThree aims to develop a set of tools, hardware and software, to enable the creation of robots that have **four** main features:

- Modularity
- Efficiency
- Usability
- Safety.

Three main actors are involved:

- Humans
- Robots
- The environment

The results will be validated in four industrial pilot studies and different use cases:

- assembly and disassembly of small parts,
- collaborative welding,
- assembly of big parts.

A permanent Living Lab will allow a continuous evaluation of the results achieved

FOURBYTHREE has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 637095
For more information visit: www.fourbythree.eu



14:00-14:10: Introduction

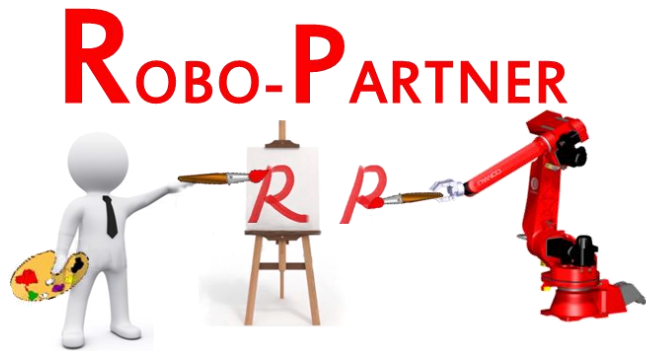
14:10-15:00: Project Presentations

- » Human robot interaction for programming of assembly tasks – ROBO-PARTNER (Dr. George Michalos, LMS-University of Patras)
- » Factory-in-a-day: wrap-up after year 1 (Prof. Gorden Cheng, TU München)
- » Hybrid Workplace Design - LIAA (Ramez Awad, Fraunhofer IPA)
- » Uniform capability description and reasoning for resources on the shop floor - ReApp (Dr. Stefan Zander, FZI)

15:00-15:25: Discussion of architecture and agreement of next steps

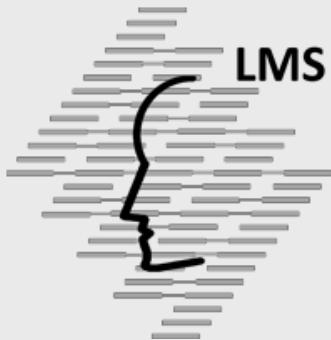
15:25-15:30: Conclusion

- **Discussion Points:**
 - ReApp Ontology
 - Reference Architecture
 - Other topics?



ROBO-PARTNER

Human robot interaction for programming of assembly tasks



Dr. George Michalos

Laboratory for Manufacturing Systems and Automation

Director: Professor G. Chryssolouris

University of Patras Tel.: +30 2610 997 262

www.lms.mech.upatras.gr

E-Mail: michalos@lms.mech.upatras.gr

- Introduction
 - Challenges and state of the art
- Robot Programming Framework
 - ✓ Human robot interaction mechanisms
 - ✓ Hierarchical decomposition of assembly operations
 - ✓ Human commands mapping to robot commands
 - ✓ ICT Integration architecture
- Application example

Introduction: Challenges

Flexible robotic equipment



Enable the execution of numerous manual assembly tasks



also..

Give the perception of human-like behavior



But

Programming methods still require robotics skills



and

Complex programming of assembly tasks



Therefore

Simpler methods for programming a robot are needed

**New technology
- sensors for
interacting with
human can help
to reduce the
complexity of
programming**



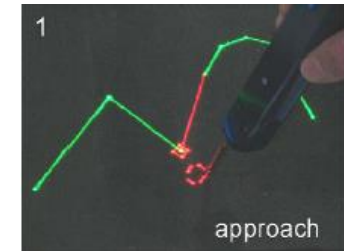
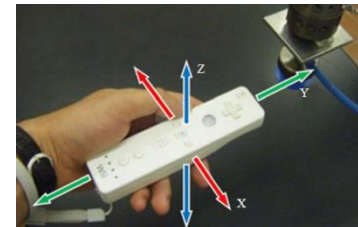
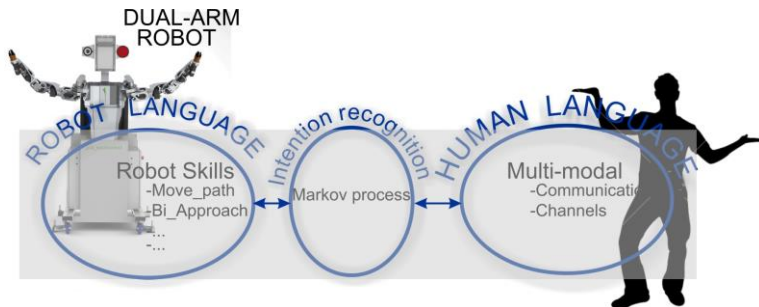
- Currently there are about 2.3 million industrial manufacturing SMEs in the EU (99% of all companies)
- Surveys by the European Business Test Panel (EBTP):
 - across 90 European companies
 - poor acceptance of robots within SMEs
 - 61% of SMEs use 1 to 10 robots / 32% between 11 to 50
- Challenges in adopting new technologies
 - Lack of flexible equipment for highly dexterous operations
 - Absence of user friendly programming techniques and interfaces
 - Difficulty to maintain and operate by non-expert users.

Introduction: Review of state of the art (1)



Methods for interactive programming

- Programming/ Learning by demonstration (PbD/LbD) approaches and instructive systems, aim at **demonstrating human-like behaviors** for programming and teaching robots, through commands such as voice, vision, haptics *[Biggs G. and Macdonald B., 2003] [Smith et al 2012][Zöllner et. al. 2004][Koenig et al 2010]*
- A few implementations in an industrial-like setup *[Reinhart et. al. 2008]*
- Human Robot Interaction architectures for programming *[Makris et. al. 2014]*



Robot programming languages

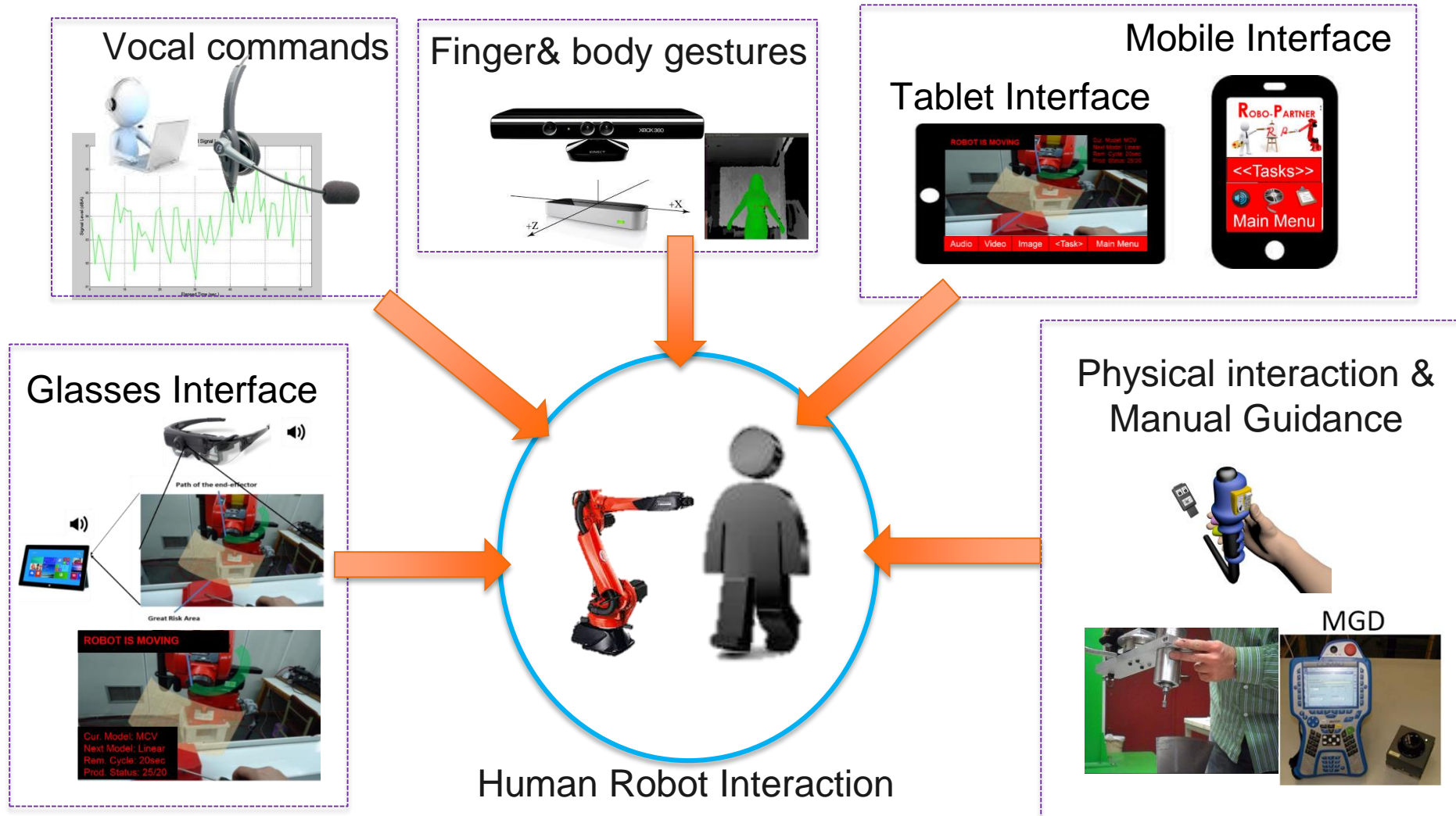
- motion-oriented, '**how to be done**'
- task-oriented, '**what**' to be done rather than 'how'

Task-oriented programming

is connected with the development of **interfaces** that will **help humans** to transfer high level commands to a sequence of robot motions and actions
[Mosemann and Wahl 2001][Thomas and Wahl, 2001][Landzettel et. al. 2001][Iba 2004]



Human-robot interaction mechanisms

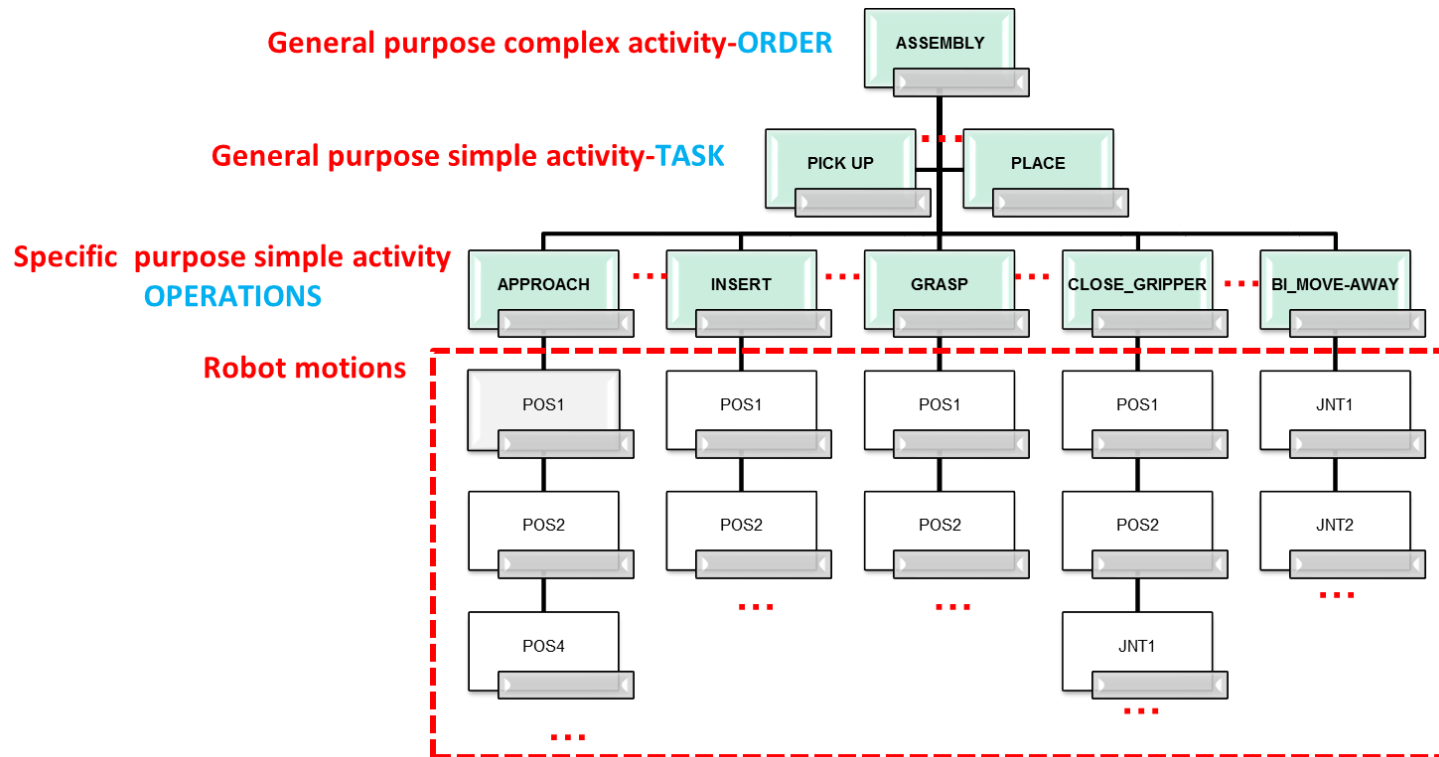


Hierarchical decomposition of operations



Decomposition of complex activities in simpler activities,

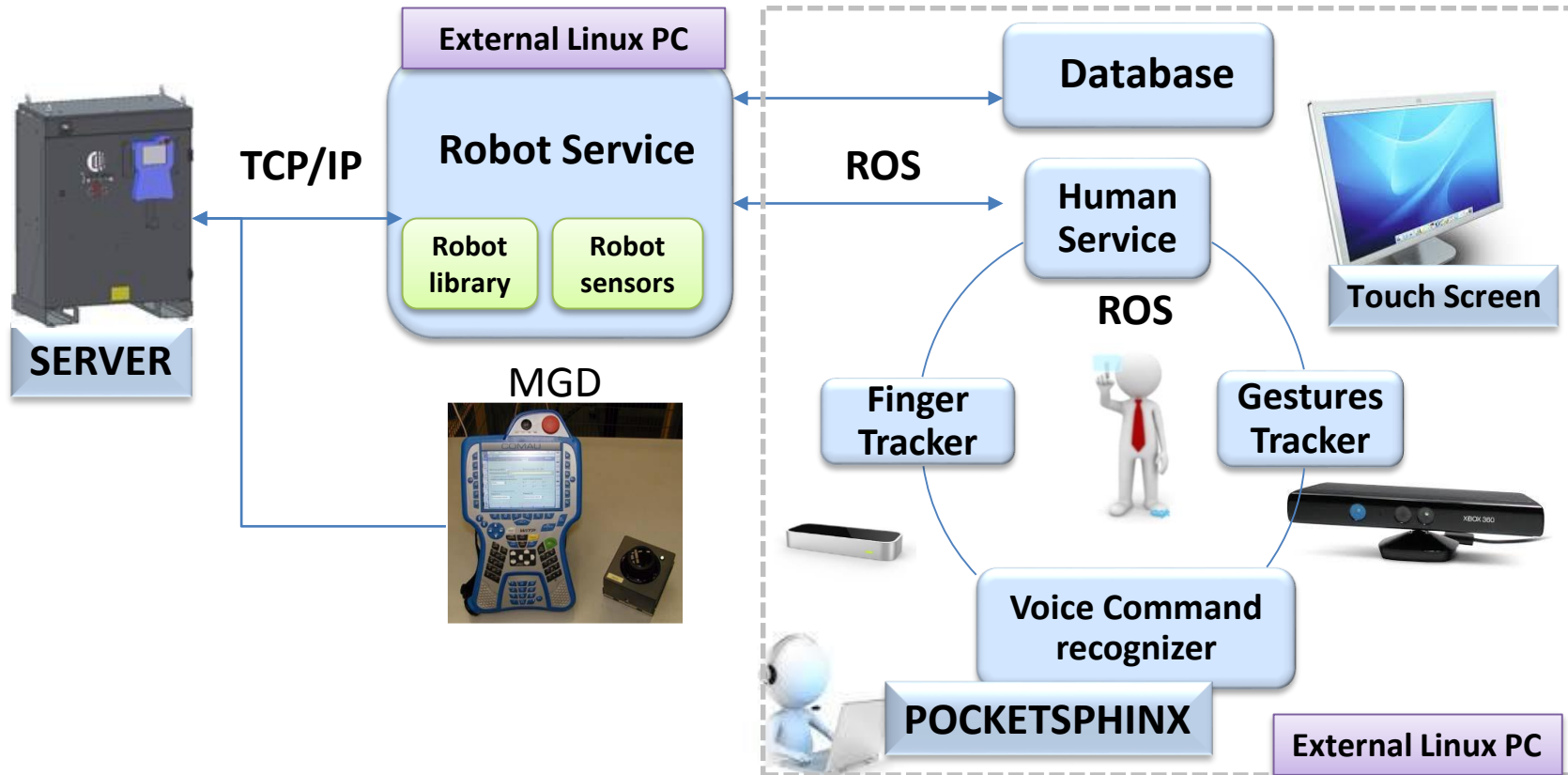
- Model assembly operations in a structured manner
- Combine with human interfaces
- Facilitate further programming of complex assembly operations



Makris, S., Tsarouchi, P., Surdilovic, D., Krüger, J., 2014, Intuitive dual arm robot programming for assembly operations, CIRP Annals - Manufacturing Technology, 63/1:13–16, DOI:10.1016/j.cirp.2014.03.017.

Programming module architecture

Synergy with

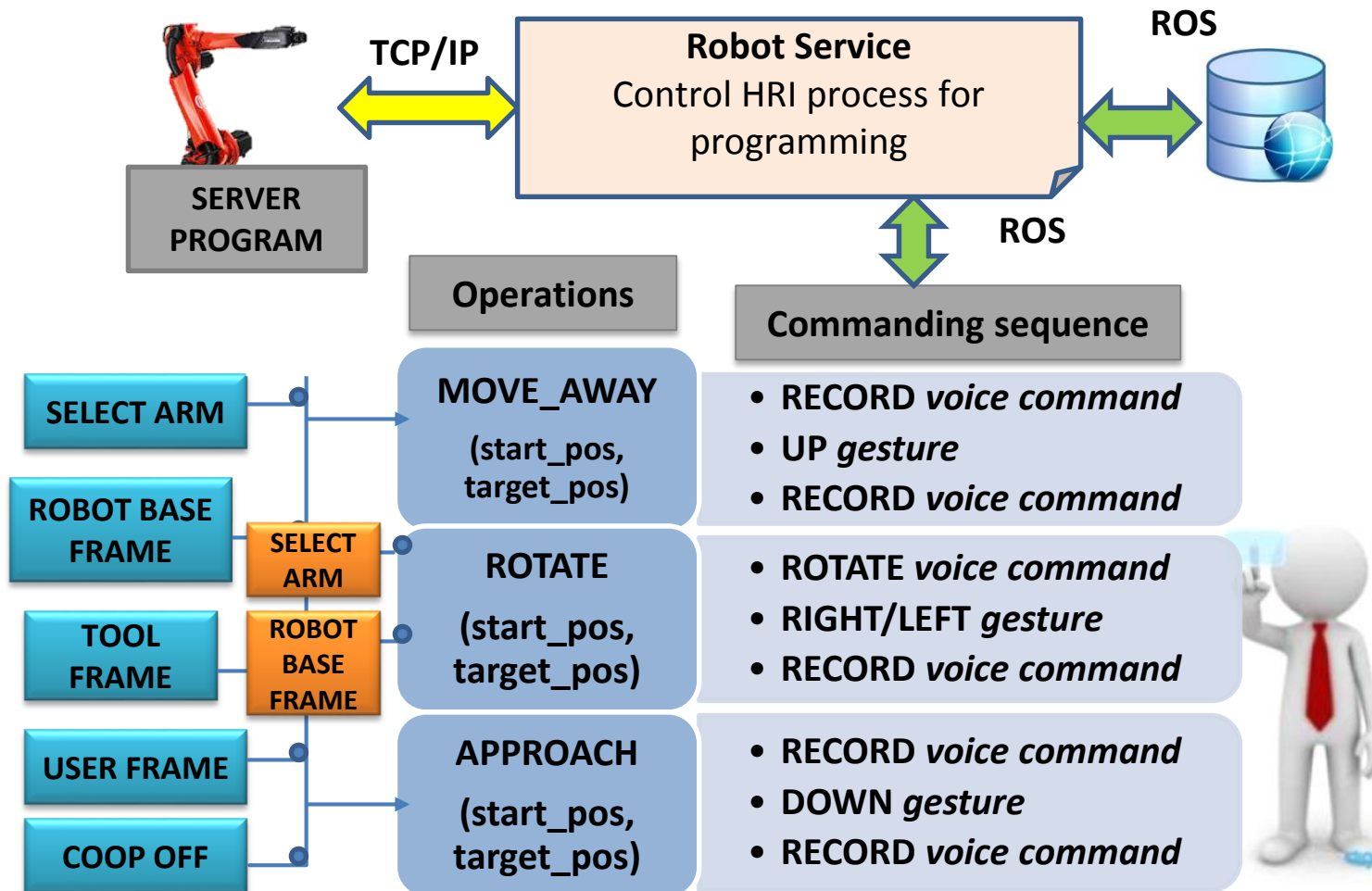


Makris, S., Tsarouchi, P., Surdilovic, D., Krüger, J., 2014, Intuitive dual arm robot programming for assembly operations, CIRP Annals - Manufacturing Technology, 63/1:13–16, DOI:10.1016/j.cirp.2014.03.017.

Human commands mapping to robot commands- an example



Synergy with



Graphical interfaces for programming



Synergy with



X-act Intuitive Interfaces

Hierarchical model

```

graph TD
    Order[Order] --> RT1[Robot Task 1]
    Order --> HT[Human Task]
    Order --> RT2[Robot Task 2]
    RT1 --> Rotate[Rotate]
    RT1 --> Approach[Approach]
    RT1 --> MoveAway[Move away]
    MoveAway --> Insert[Insert]
    MoveAway --> Extract[Extract]
    MoveAway --> Screw[Screw]
    MoveAway --> OpenGripper[Open Gripper]
    MoveAway --> CloseGripper[Close Gripper]
    MoveAway --> Custom[Custom]
    Custom --> BiApproach[Bi-Approach]
    Custom --> BiMoveAway[Bi-Move away]
    Custom --> SyncMoveAway[Sync-Move away]
    Custom --> SyncApproach[Sync-Approach]
        
```

MOVE AWAY INFO:

Under Shown Arm:

Step 1: Select Motion {SINGLE, COOPERATIVE, SYNC}

Step 2: Select Tool {Tool 1, Tool 3 etc for Arm 1 -- Tool 2, Tool 4 etc for Arm 2}

Step 3: Select User Frame { User Frame 1, User Frame 3 etc for Arm 1 -- User Frame 2, User Frame 4 etc for Arm 2}

Step 4: Select Base Frame { Base Frame 1, Base Frame 3 etc for Arm 1 -- Base Frame 2, Base Frame 4 etc for Arm 2}

Step 5: Select Motion Type {Linear, Joint, User Frame etc}

Programming Status

Operation Move away started...

TORSO	
ARM 1	ARM 2
Motion	Motion
Tool	Tool
User Frame	User Frame
Base Frame	Base Frame
COORD	COORD
Gripper	Gripper

Gesture Based Programming

End Position

J1:

J2:

J3:

J4:

J5:

J6:

End Position in RVIZ

Virtual Teach Pendant

☐ Visualize Path(Rviz)

Select Arm

Calculate Path

Test Path

Save Path

Programming interface parameters

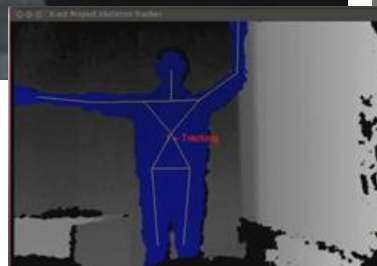
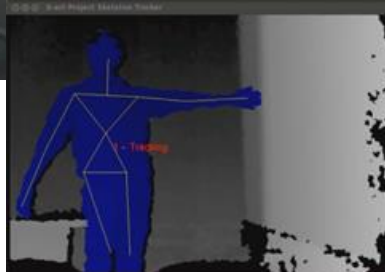
Assembly example (1/2)



Synergy with



Gesture commanding in
single arm robot



Gesture commanding in
dual arm robot



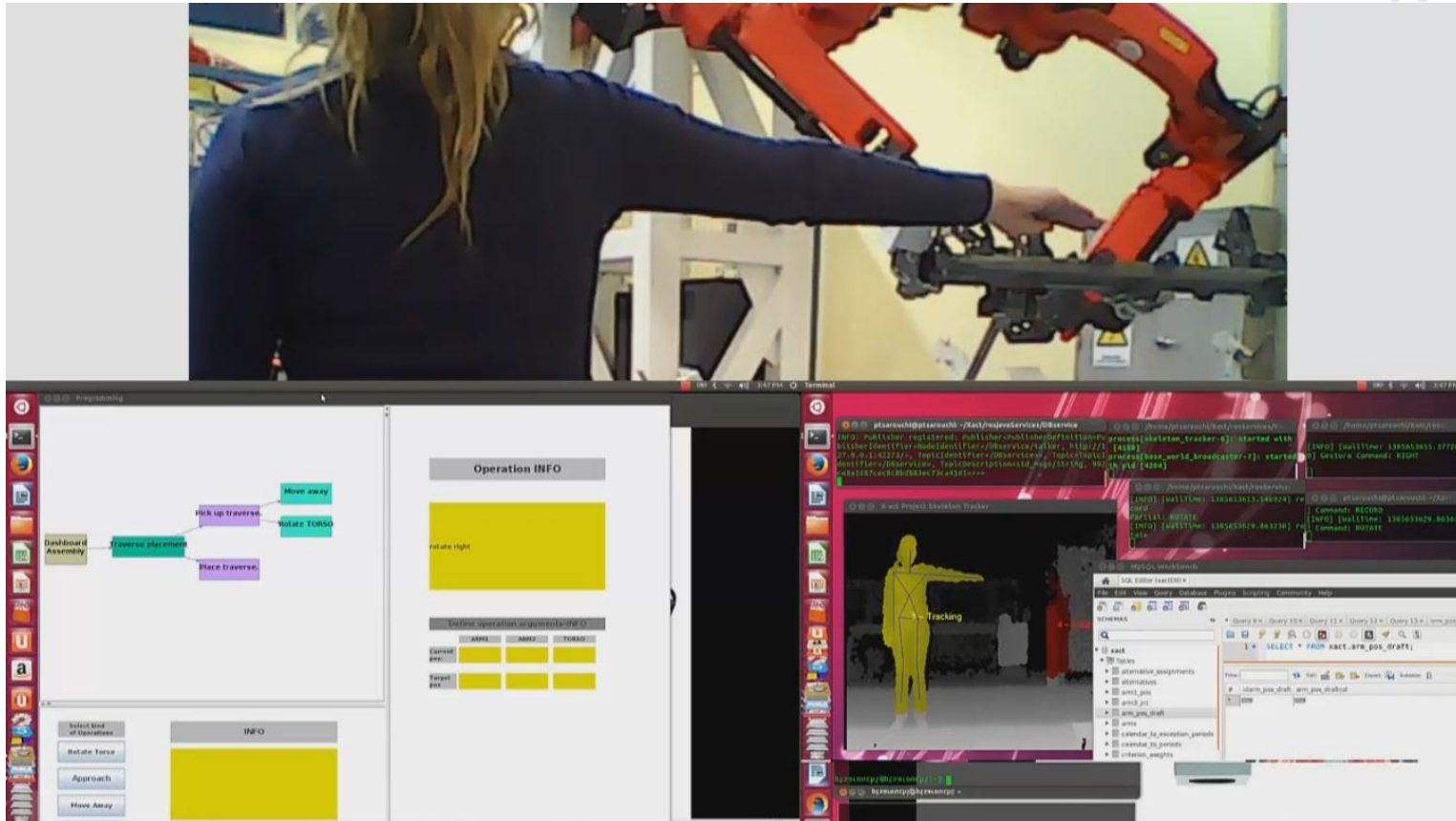
Assembly example (2/2)



Synergy with



Programming Module application in dual
arm robot



Contact Information



Presenter

Dr. George Michalos
Laboratory for Manufacturing Systems and Automation (LMS)
Tel: +30-2610-997262
Fax: +30-2610-997744
E-Mail: michalos@lms.mech.upatras.gr

Project Coordinator

Onder Tokcalar
TOFAS TURK OTOMOBIL FABRICASI A.S
Tel: +90-224-261.0350
E-Mail: onder.tokcalar@tofas.com.tr



For more information visit us at www.robo-partner.eu

Thank you for your Attention!

Questions?

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 608855

&

and the project X-act/ FoF-ICT-314355 (<http://www.xact-project.eu/>), funded by the FP7 framework.



Supported by:



Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag



Workshop: Hybrid Production Systems

Factory-in-a-day(Fiad)

Factory-in-a-day: Wrap-up after year 1

Prof. Gordon Cheng



Technische Universität München

Technische Universität München
Karlstr. 45
80333 München
Germany
Tel.: +49-89-289-26800
E-Mail: gordon@tum.de

Factory-in-a-day: The Dream

Analyze workflow



Design custom components for the job



Components are 3D printed



8:00 Everything is shipped to the factory



10:00 Unloading and self calibration



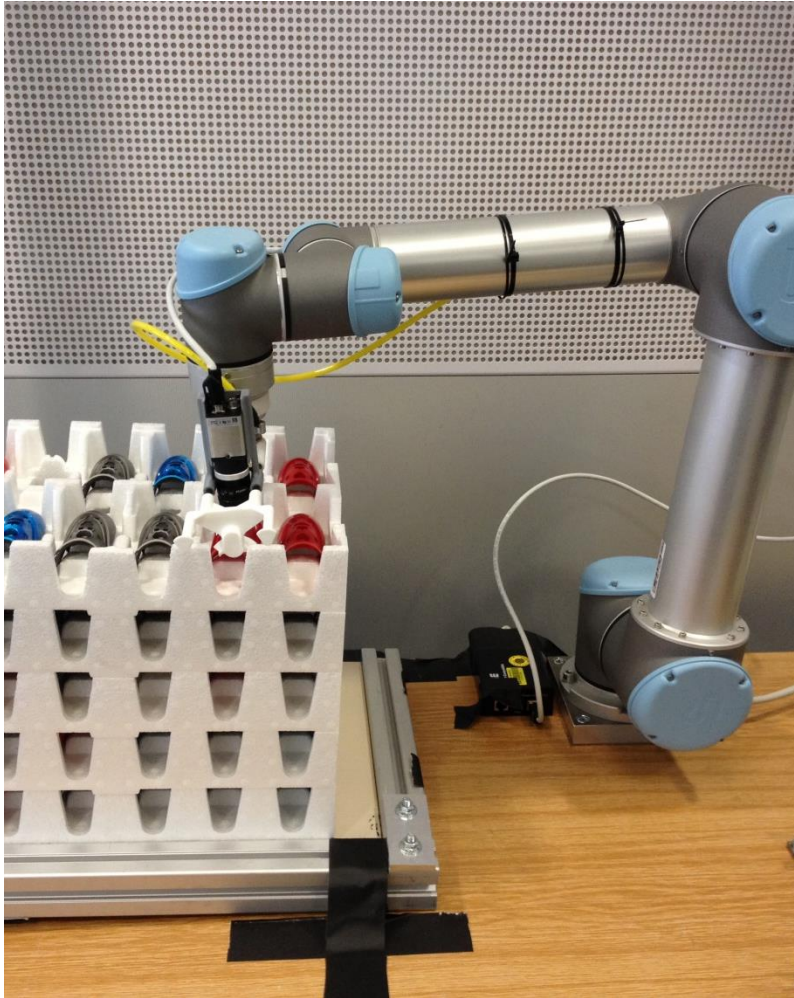
12:00 Instruction and teaching



16:00 Done!

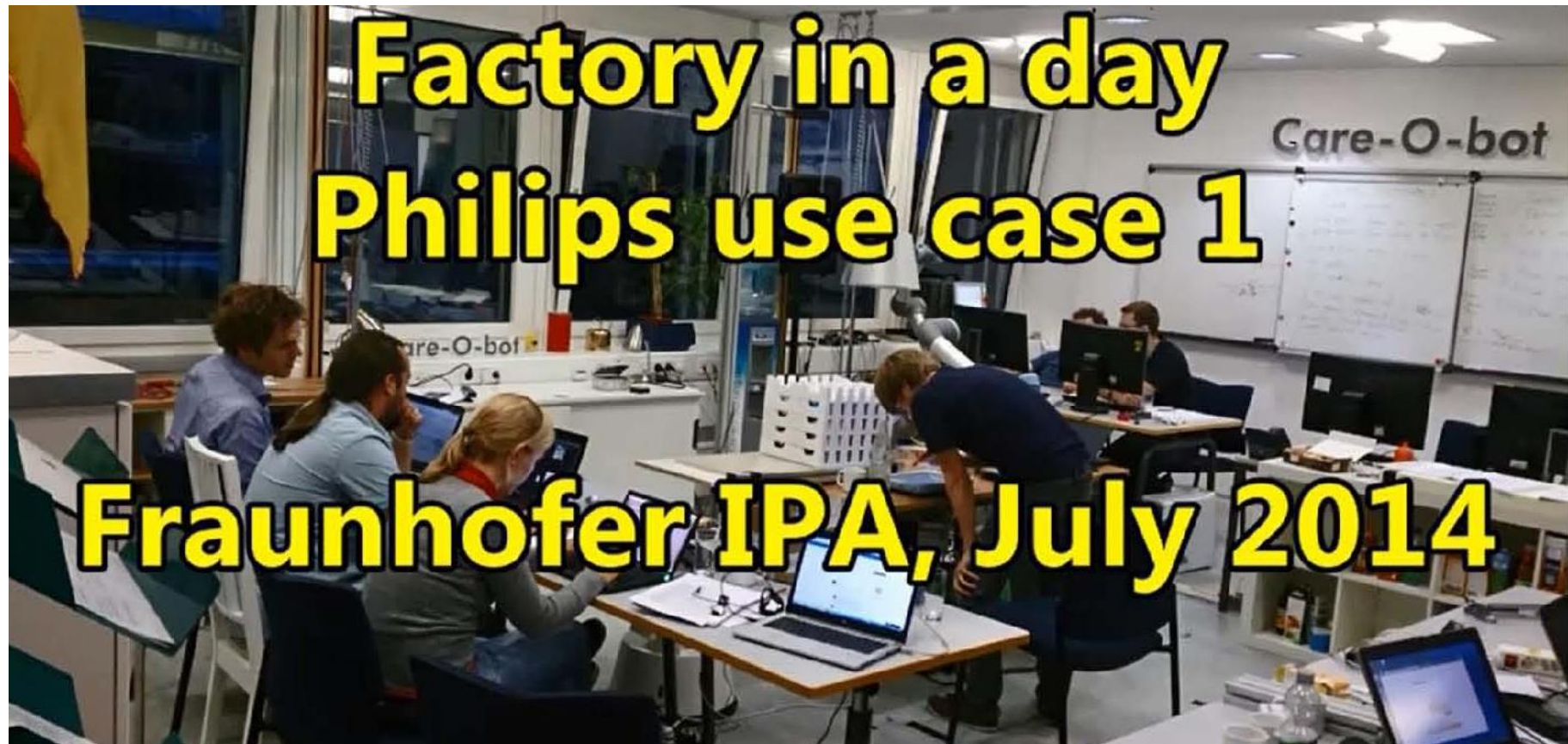


Example test case



Philips case: (un)loading of trays with razor parts for printing





http://youtu.be/p5ds-Q9_0CA

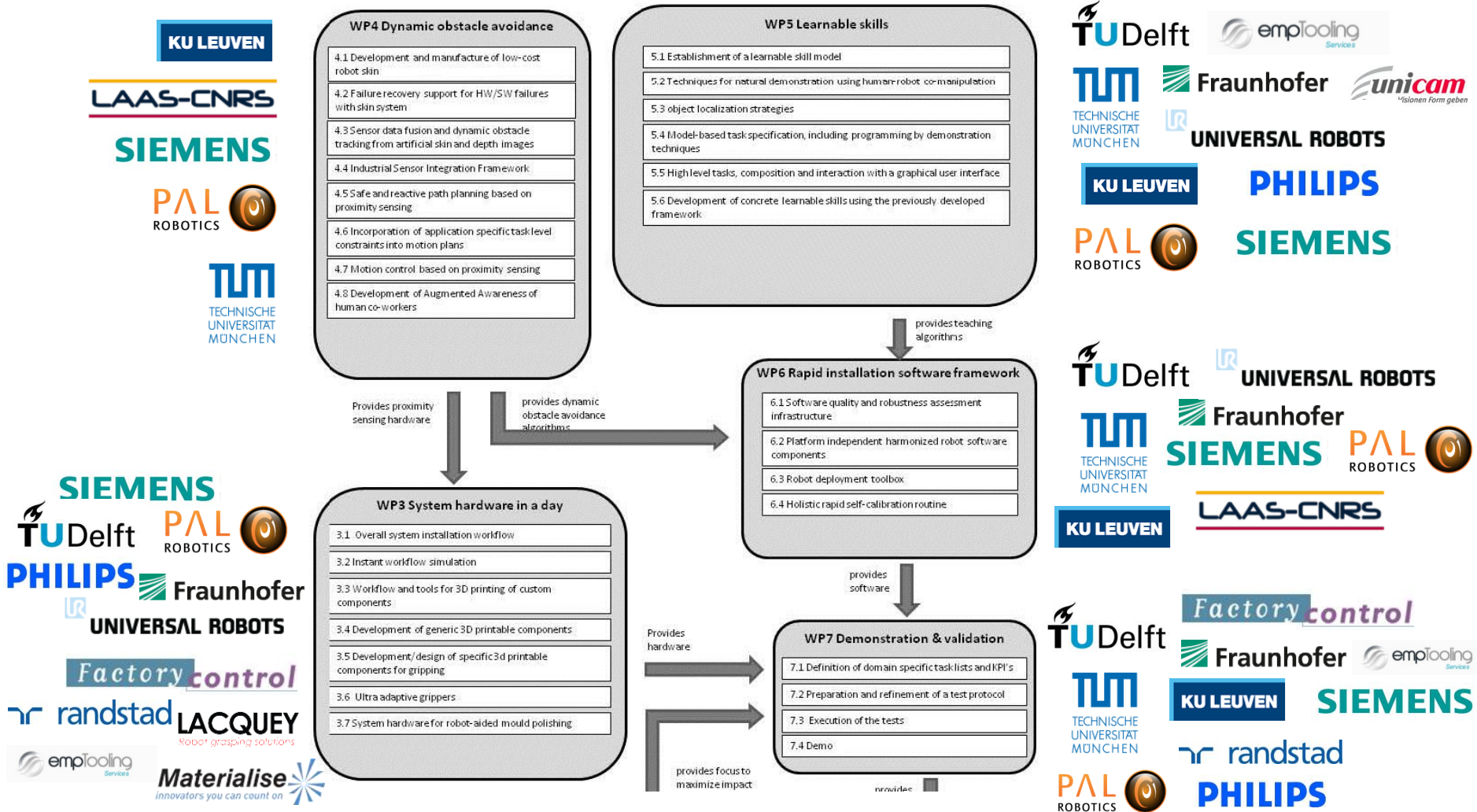
What did we do?

Direct tasks	Indirect tasks
Loading parts in the machine Machine loading	Fetch and bring lorries with trays with products
Unloading parts from the machine	Load and unload trays from lorries on supply and exit line
Placing trays with new parts from supply line on workstation Tray handling	Machine error
Placing trays with finished parts from workstation on exit line	Mix new
Perform visual quality check	Swap tamper
Log outcome of quality check	Finetune
Quality checking	Clean
	Break
	Teammeetings
	Fix machine errors/failures

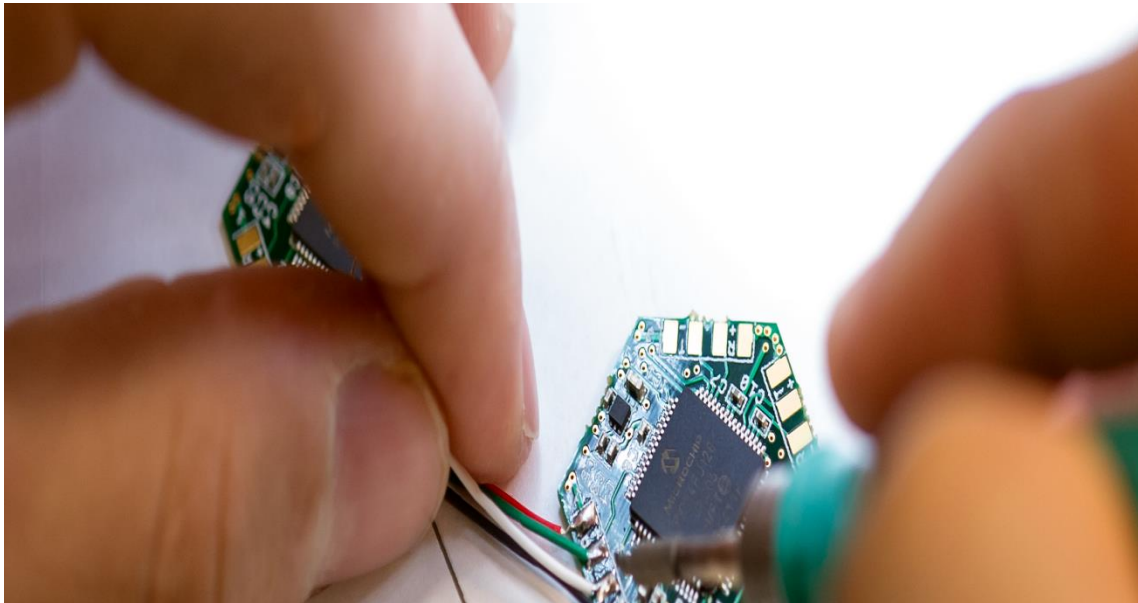
How did we do?

Metric	Example industry	Hackathon July 2014
Installation effort	8750 [hrs]	200 [hrs]
Reuse value	25 %	65 %
Installation time	40 [weeks]	30 [days]
Cost of reconfiguration	200,000 [euros]	2,000 [euros]
Reconfiguration time	20 [weeks]	2 [days]
Uptime	91 %	40%
Production rate	12.0 [parts/min]	1.2 [parts/min]
Operator load	0.1 FTE	0.6 FTE
Automation level	130 [parts/min/operator]	2 [parts/min/operator]

Technical work packages in Fiad



- CelluARSkin: hexagonal-shaped skin-sensor for robots
- Modalities: temperature, proximity, force and vibration
- max frequency, self-calibrating system
- Cell-to-cell communication



The skin-sensors are integrated in the project on different levels:

T4.1: Development and manufacture of low-cost robot skin

T4.2: Failure recovery support for HW/SW failures with skin system

T4.3: Sensor data fusion and dynamic obstacle tracking from proximity sensing and depth images

T4.5: Safe and reactive path planning based on proximity sensing

T4.7: Motion control based on proximity sensing (8pm)

Task 5.1: Establishment of a learnable skill model

Task 5.2: Techniques for natural demonstration using human-robot co-manipulation

Task 6.4: Holistic rapid self-calibration routine



Visit us in the exhibition area!

Presenter

Prof. Gordon Cheng
Tel: +49 89 289 26800
Fax: +49 89 289 26815
E-Mail: gordon@tum.de

Project Coordinator

Dr. Martijn Wisse
Tel: +31 15-27 86834
E-Mail: M.Wisse@tudelft.nl

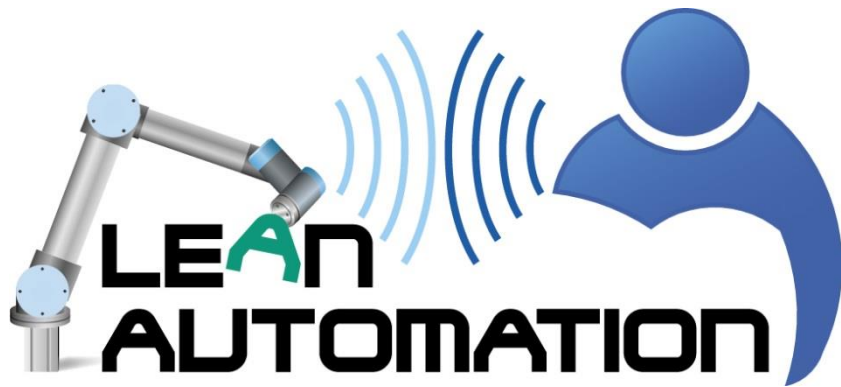
Thank you for your Attention!

Questions?



The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 609206.

For more information visit us at www.factory-in-a-day.eu



Supported by:



Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag



Workshop: Hybrid Production Systems

Lean Intelligent Assembly Automation (LIAA)

WP03: Computer-Aided Symbiotic Workplace Design



Ramez Awad

Fraunhofer IPA
Nobelstr. 12
70569 Stuttgart
Germany
Tel.: +49 711 970 -1844
E-Mail: ramez.awad@ipa.fraunhofer.de

- **Motivation:**

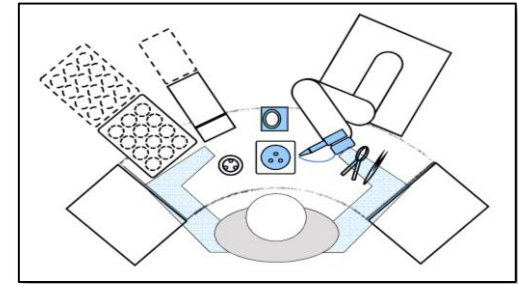
- Design and evaluation criteria of hybrid work places → scarce

- **Objectives:**

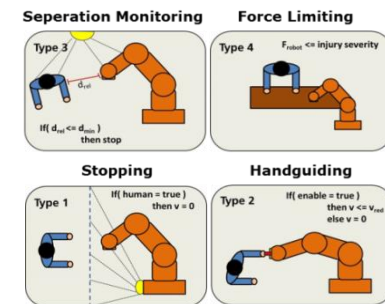
- Criteria and metrics for evaluating hybrid assembly workplaces
- Risk Assessment methodology for hybrid assembly workplace designs
- Safety Concepts for different risk levels
- Guidelines (e.g. manual / wizard) incorporating planning and design principles for symbiotic human-robot-cooperation workplaces
- 3D-simulation environment for evaluating the applicability of a workplace for symbiotic human-robot-cooperation
- Computer aided symbiotic workplace design tool, that
 - highlights risk areas
 - calculates risk levels
 - suggests safety concepts

- A set of prototypes:
 - **Design Conceptualization Tool**
 - Part Supply Calculation Tool
 - **Robot Cycle Time Estimation**
 - KPI Computation Module
 - **3D-Automate**

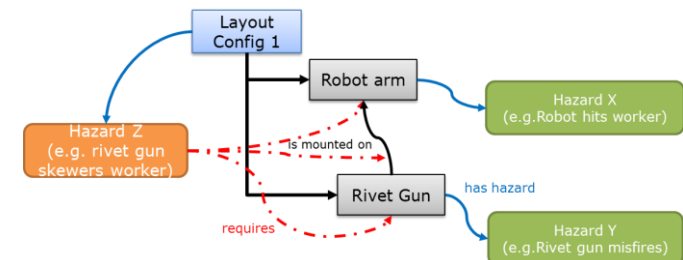
- **Lead Partner:** Fraunhofer IPA
- **Purpose:**
 - Consideration of hazards and the impact of safety measures on the KPIs of a workplace design during conceptualization phase.
- **Approach:**
 - Map potential hazards to equipment classes / processes / configuration of workplaces
 - Identify suitable Human-Robot-Collaboration (HRC-) Type w.r.t. assembly processes
 - Identify suitable Safety Principles and Measures for hazards/risk scenarios
 - Analyse impact of Safety Measures on KPIs.



Graphical representation of a design concept



Safety Principles



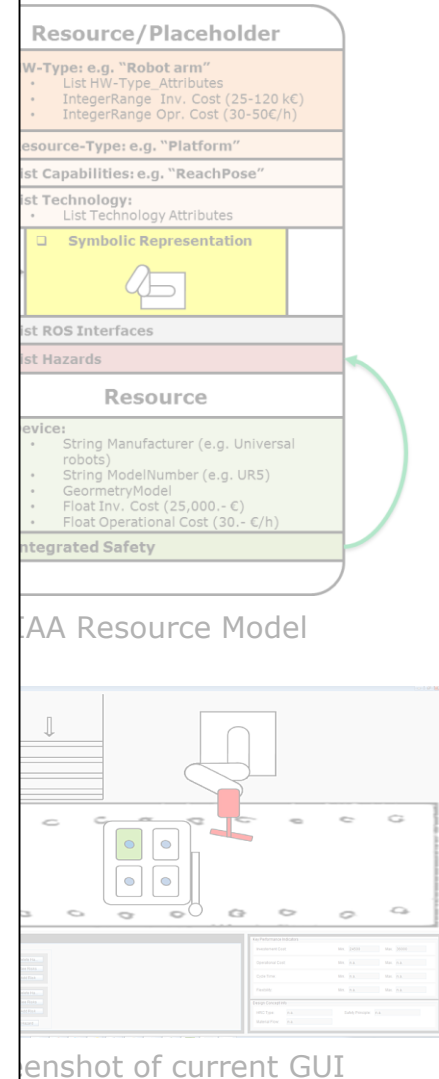
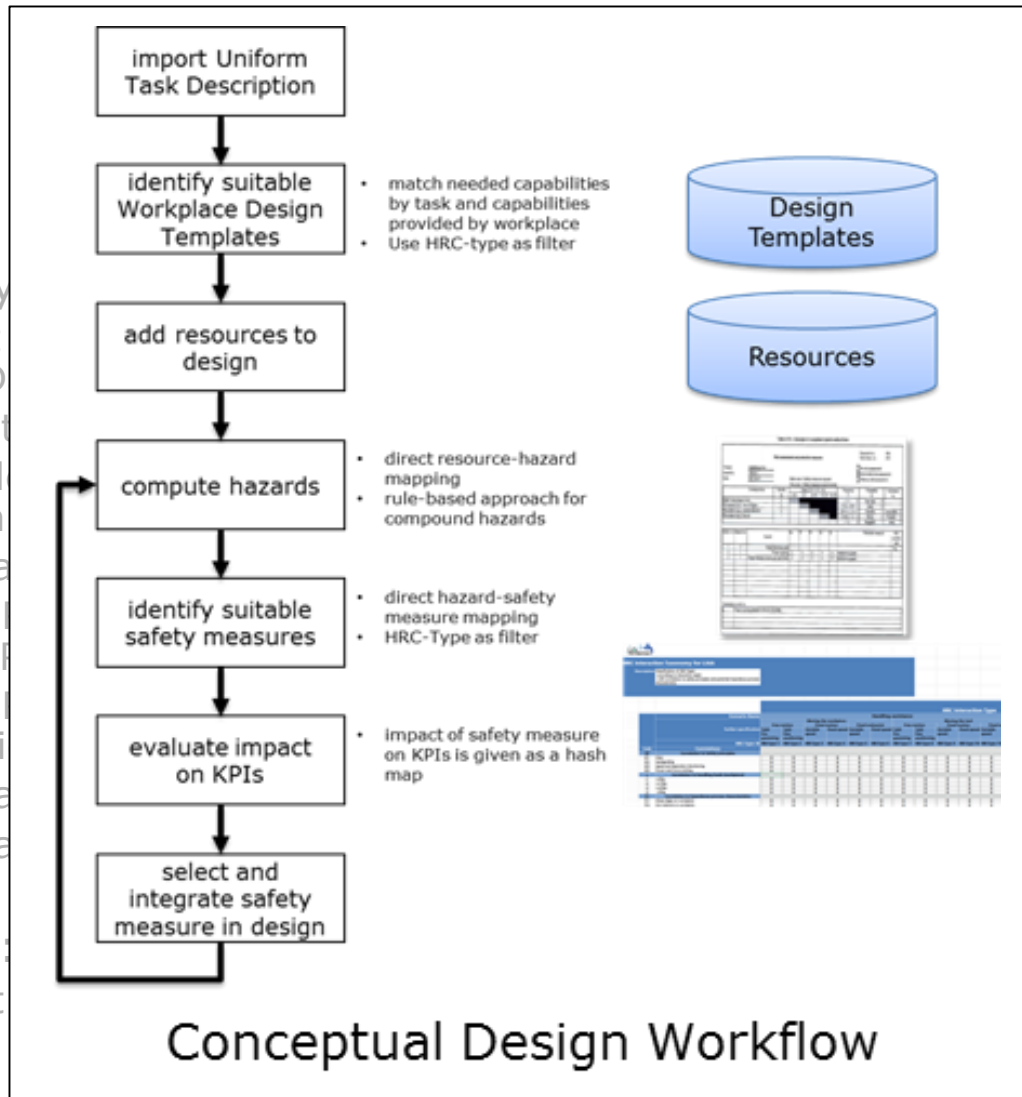
Compound Hazards

Design Conceptualization Tool (2/2)



• Status:

- Workflow
- 1st prototype
 - import from D
 - evaluation
 - add additional Design
 - Correlation
 - H
 - K
 - H
 - pi
 - sa
 - sa
- Next steps:
 - impact



- **Lead Partner:** Opel AG
- **Purpose:**
 - Cycle time is needed to select suitable resources and to validate if user demands can be met
 - KPI estimation (WP03) and task assignment (WP04) need cycle time as input
- **Approach:**
 - As current state of the art robot cycle time can be determined using simulation (high effort) or experimentally (even higher effort)
 - System similar to MTM (human cycle time estimation) will be developed for robot cycle time estimation



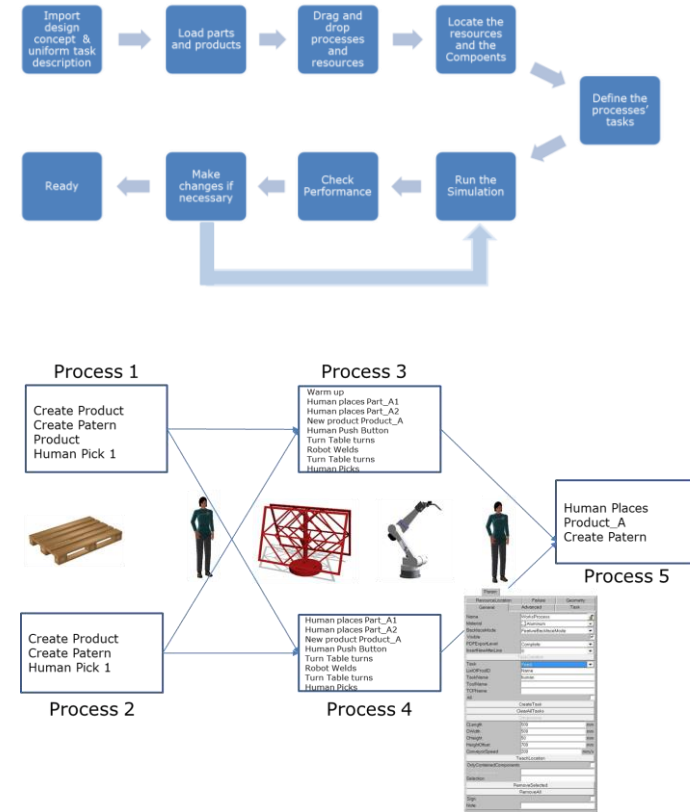
Robust cycle time estimation with low effort
for early planning stage

- **Status:**

- Prototype for articulated arm robot
- Standard operations can be combined to evaluate complex tasks:
 - **Reach, Grab**
 - **Move, Release**
 - **Place/Join**
- Validation and data generation on Kuka LBR iiwa 14
- First version with compatibility to MTM UAS

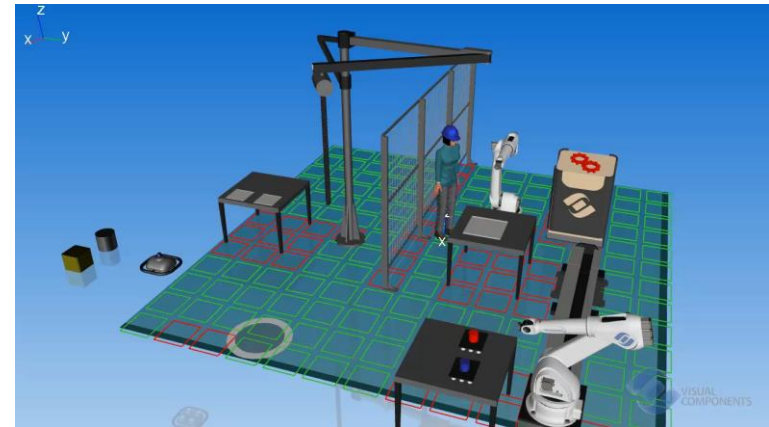
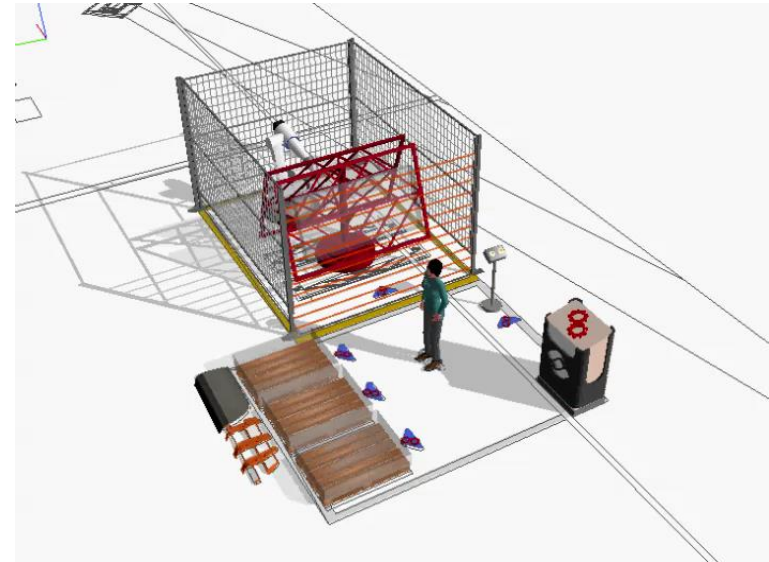
[illegible]

- **Lead Partner:** Visual Components Oy
- **Purpose:**
 - Simulate assembly tasks to evaluate geometrical and dynamic hazards/risk scenarios
- **Approach:**
 - Import Design Concept
 - Populate with geometric models
 - Import Uniform Task Description
 - Distribute tasks to resources
 - Run Simulation, evaluate hazard constraints w.r.t. speeds, forces, collisions.



- **Status:**

- 3D Automate is evolving the human model for providing a realistic human simulation, with new functionalities for the human works model
 - Adjust the carry position for components
 - Adjust the location of a human at a station
 - Visualization of transportation rules
 - Perform of human processes:
 - Creating the needed working conditions (high level definition) at a station
 - Reserve and allocate human resources to perform specific tasks
 - Avoid collision and enforce safety rules while walking
 - Execute conditional tasks



Contact Information



Presenter

Ramez Awad

Tel: +49 711 970-1844

Fax: +49 711 970-1008

E-Mail: ramez.awad@ipa.fraunhofer.de

Project Coordinator

Martin Naumann

Tel: +49 711 970-1291

Fax: +49 711 970-1008

E-Mail: martin.naumann@ipa.fraunhofer.de

Thank you for your Attention!

Questions?



DANISH
TECHNOLOGICAL
INSTITUTE



UNIVERSAL ROBOTS



InSystems
automation



dresden elektronik



For more information visit us at www.project-leanautomation.eu

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 608604.



Supported by:



Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag



Workshop: Hybrid production systems

Reusable robot applications for flexible robot systems based on Industrial ROS (ReApp)

Reasoning on Features of Robot-Centric Workplaces using Ontological Semantics



Dr. Stefan Zander

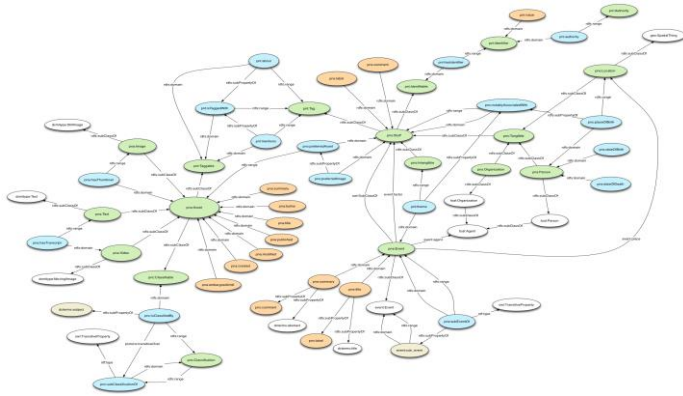
FZI Forschungszentrum Informatik am Karlsruhe Institut für Technologie (KIT)
Haid-und-Neu-Straße 10-14
76131 Karlsruhe
Germany
Tel: +49 721 9654-888
E-Mail: zander@fzi.de

- Processing the **formal, model-theoretic semantics*** of the ReApp Ontologies and deducing **implicit knowledge** through the notion of **logical consequence**.

** Encoded in the form of logical axioms*

- **Subsumption** – find out whether class C is a subclass of D , i.e., $C \sqsubseteq D$
- **Class Equivalence** – find out whether class C is equivalent to D , i.e., $C \equiv D$
- **Class Disjointness** – find out whether C and D disjoint, i.e., $C \sqcap D \sqsubseteq \perp$
- **Global Consistency** – find out whether the ReApp knowledge base is globally consistent, i.e., that it has a model
- **Class Consistency** – e.g., find out whether a given class C is consistent, i.e., show that $C \sqsubseteq \perp$ is not a logical consequence of the given knowledge base
- **Instance Checking** – e.g., find out if an individual a belongs to a class C , i.e., check whether $C(a)$ is a logical consequence of the given knowledge base
- **Instance Retrieval** – find all individuals that are member of a given class or class expression

HW-, SW-, & Capabilities Ontologies



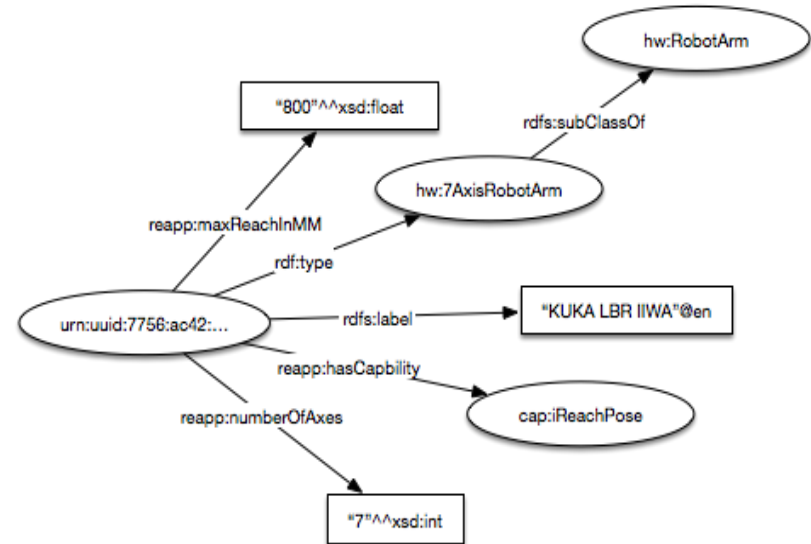
Source: <http://data.press.net/ontology/images/big-ontology-picture.png>



Terminological Knowledge
(Classes, Relationships, Constraints)

≡ Materialization

Used to represent data about a concrete instance of an App (ie., Device, ROS Wrapper, Software Component, Skill, Solution etc.)



TBox - Logic

ABox - Instances

Capabilities Computation

Problem:

- **How to represent capabilities of Components on TBox level?**
 - E.g., state that safety laser scanners (class) have the default capability safe monitoring of 2D fields

Solution:

- Encode capabilities in **subsumption axioms**
 - $\text{SafetyLaserScanner} \sqsubseteq \exists \text{hasCapability.SafeMonitoringOf2DFields}$

Feature Materializing for Components



Encoding of Capability Axioms

The screenshot displays the Protégé ontology editor interface. The left pane shows the 'Class hierarchy' for 'SafteyLaserScanner', which includes various subclasses like 'ReachPose', 'RealTimeAudioStreaming', 'RealTimeVideoStreaming', 'Recognition', 'Sensing', 'Tracking', 'Transportation', 'Communication', 'CommunicationPhysical', 'CommunicationProtocol', 'Environment', 'OperationEnvironment', 'HardwareType', 'HardwareDevice', 'Asus_Xtion_PRO_LIVE', 'AX-12A', 'ExampleWorkplace', 'Kuka_IIWA', 'Schunk_WSG-50', 'Sick_LMS511-20100PRO', 'Sick_MiniTwin4', 'Sick_S30B-2011GA', 'Sick_TIM551-2050001', 'Sick_XYZ_Sensor', 'Actuator', 'Controller', 'DigitalIO', 'HMIComponent', 'PhysicalObject', 'Sensor', 'Acceleration1DSensor', 'Acceleration3DSensor', 'Acceleration6DSensor', 'Depth1DPointCloudSensor', 'Depth2DPointCloudSensor', 'LaserScanner', 'SafteyLaserScanner', 'Radar2D', 'Depth3DPointCloudSensor', 'Force1DSensor', 'Force2DSensor', 'Force3DSensor', and 'Identifier'. The right pane shows the 'Annotations' for 'SafteyLaserScanner', including an 'rdfs:comment' in English: 'The sufficient conditions should hold those AML properties that are relevant for characterizing a component as safetyLaserScanner (2014-08-25,SZ)'. Below the annotations, the 'Description' pane shows 'Equivalent To' and 'SubClass Of' relationships, including 'hasCapability some SafeMonitoringOf2DFields', 'hasOperationEnvironment some Indoor', 'hasOutputFormat some Depth2DPointCloud', and 'LaserScanner'. The bottom status bar indicates 'Reasoner active' and 'Show Inferences' is checked.

Query for Capability Information

DL query:

Query (class expression)

HardwareDevice **and** hasCapability **some** MonitoringOf2DFields

Execute Add to ontology

Query results

Direct sub classes (3)

- Sick_LMS511-20100PRO
- Sick_...
- Sick_...

Instances (1)

- ◆ ind_S...

☐ Direct super classes

☐ Super classes

DL query:

Query (class expression)

HardwareDevice **and** hasCapability **some** SafeMonitoringOf2DFields

Execute Add to ontology

Query results

Direct sub classes (1)

- Sick_S30B-2011GA

Instances (1)

- ◆ ind_Sick30B

☐ Direct super classes

☐ Super classes

☐ Equivalent classes

☒ Direct sub classes

☐ Sub classes

☒ Instances

Computing Complex Capabilities

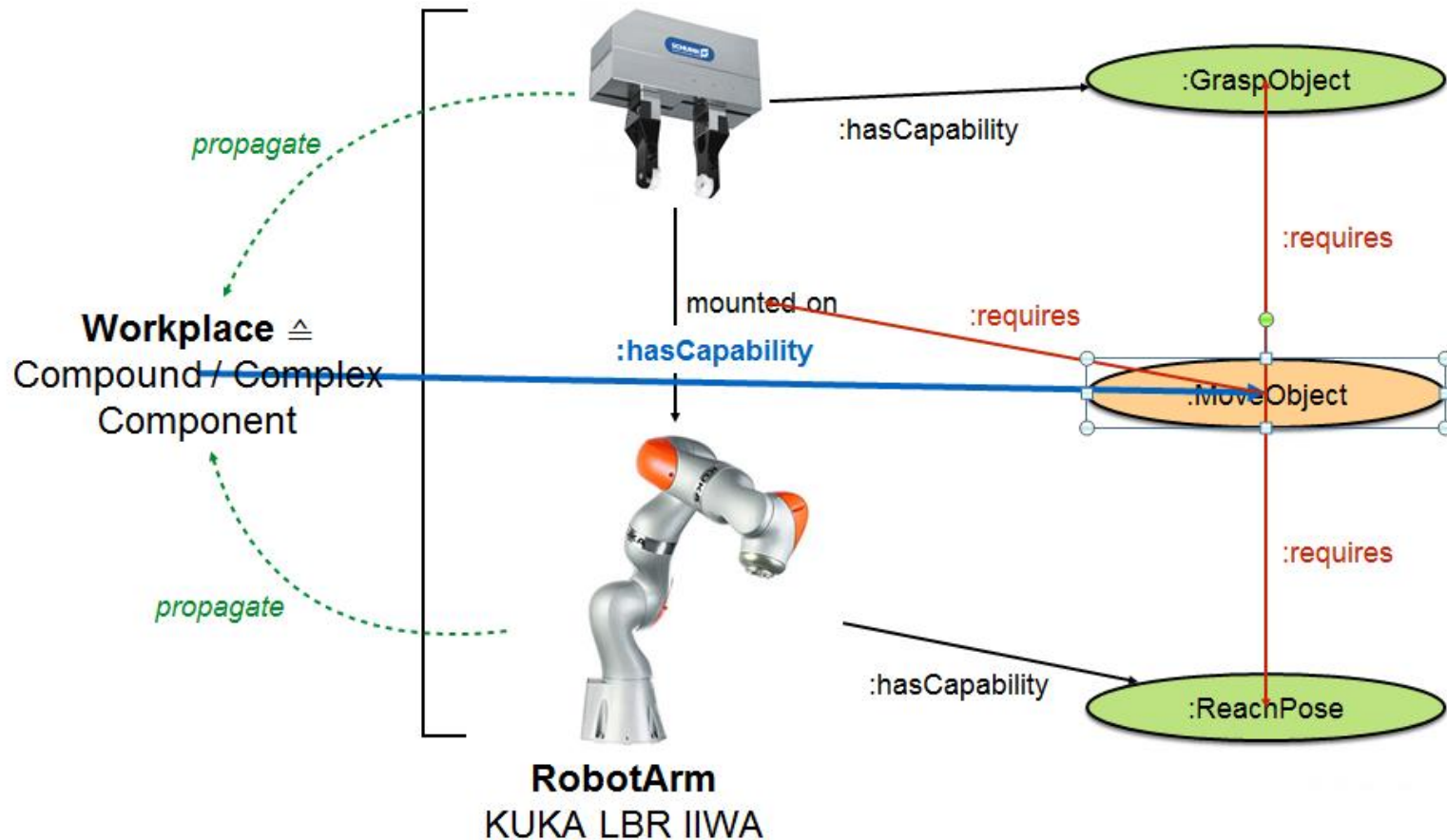
Problem:

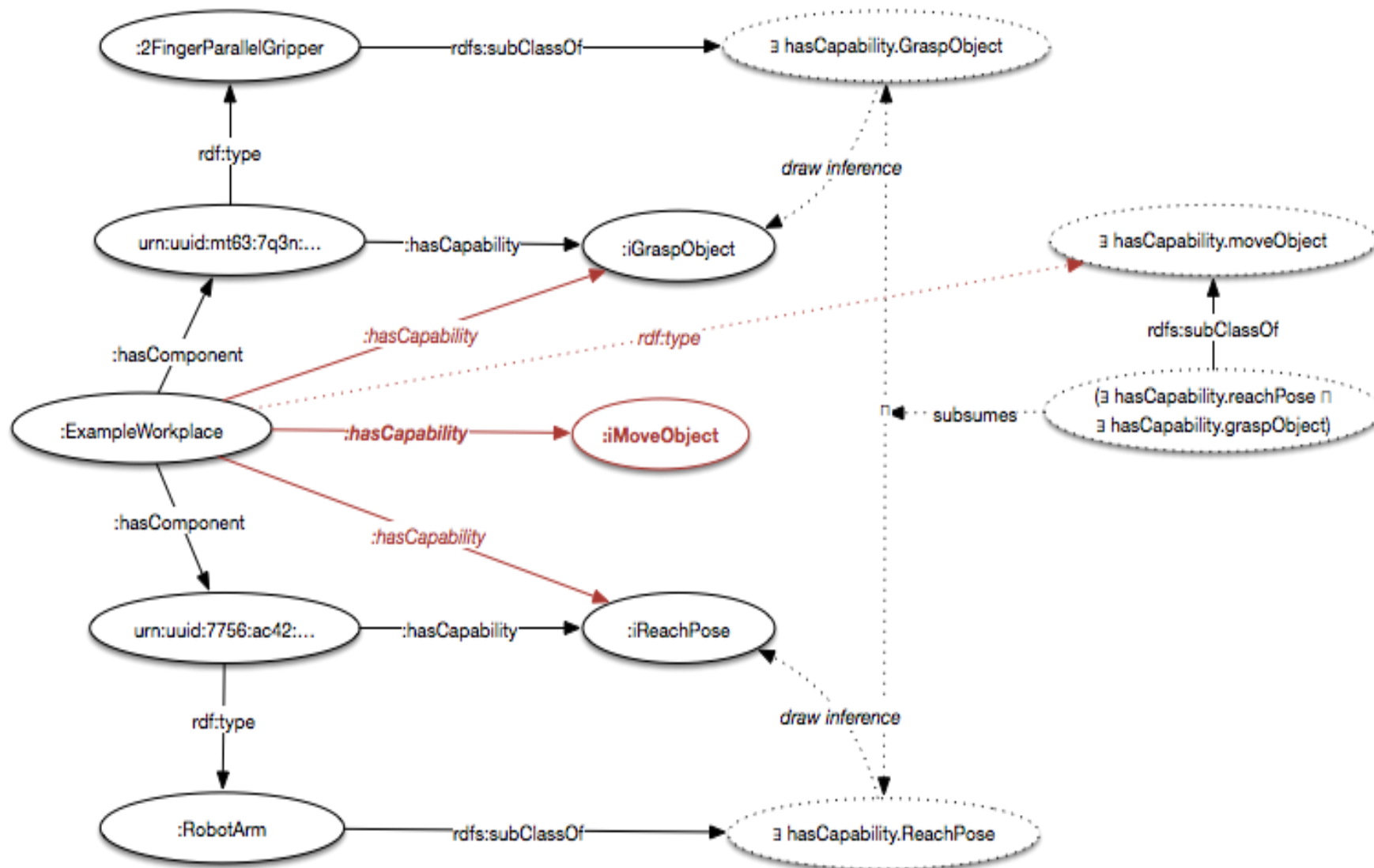
- Computing the capabilities of **compound components**
 - e.g., find out what are the capabilities of a component that consists of other components (e.g., robot arm and gripper)?

Solution:

- Capability propagation via **property chain axioms** and **general class inclusion axioms**
 - e.g., $\exists \text{hasCapability.MoveObject} \sqsubseteq \exists \text{hasCapability.ReachPose} \sqcap \exists \text{hasCapability.GraspObject}$

2-Finger Parallel Gripper Schunk WSG 50





The screenshot shows the Protégé ontology editor interface. The left pane displays the 'Class hierarchy' for 'Schunk_WSG-50', which includes a tree structure with classes like 'ReachPose', 'RealTimeAudioStreaming', 'RealTimeVideoStreaming', 'Recognition', 'Sensing', 'Tracking', 'Transportation', 'Communication', 'CommunicationPhysical', 'CommunicationProtocol', 'Environment', 'OperationEnvironment', 'HardwareType', 'HardwareDevice', and various sensor models like 'Sick_LMS511-20100PRO' and 'Sick_XYZ_Sensor'. The right pane shows the 'Annotations' and 'Description' for 'Schunk_WSG-50'. The 'Annotations' tab is active, showing a comment: 'This class is used to demonstrate automated classification based on the evaluation of some datatype properties: for the given component (gripper), the amount of fingers are evaluated. When the amount satisfies the restriction of the super class (2-finger gripper), this class is added as subclass'. The 'Description' tab shows the class is a subclass of 'HardwareDevice' and has several properties: 'hasCapability some GrabAndRelease', 'hasCapability some GraspObject', 'hasFingers value 2', and 'weightInKG value 20'. A red box highlights the 'SubClass Of' section, and a yellow box highlights the 'General class axioms' section.

The screenshot shows the Protégé ontology editor interface. The left pane displays the 'Class hierarchy' for 'Kuka_IIWA', listing various classes like AX-12A, ExampleWorkplace, Schunk_WSG-50, and RobotArm. The right pane shows the 'Description' for 'Kuka_IIWA', which includes a 'SubClass Of' list. This list is highlighted with a red box and contains the following entries: HardwareDevice, hasCapability some ReachPose, maxNumberOfAxis value 7, MaxPayloadInKG value 1000, and 7AxisRobotArm. Below this, the 'General class axioms' section shows 'SubClass Of (Anonymous Ancestor)' with the axiom 'maxNumberOfAxis value 7'. The bottom right of the interface indicates 'Reasoner active' and 'Show Inferences' is checked.

hw-taxonomy (http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy) : [/Users/zander/Projects/ReApp/svn/ont/hw-taxonomy/branch/2014-07-03_capability_components/HW-Taxonomy_new,...]

hw-taxonomy (http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy)

Active Ontology | Entities | **Classes** | Object Properties | Data Properties | Annotation Properties | Individuals | OWLviz | DL Query | OntoGraf | Ontology Differences | SPARQL Query

Class hierarchy | Class hierarchy (inferred)

Class hierarchy: Kuka_IIWA

- AX-12A
- ExampleWorkplace
- Kuka_IIWA**
- Schunk_WSG-50
- Sick_LMS511-20100PRO
- Sick_MiniTwin4
- Sick_S30B-2011GA
- Sick_TIM551-2050001
- Sick_XYZ_Sensor
- Actuator
 - ConveyorBelt
- Gripper
- MobilePlatform
- Motor
- RobotArm
 - 7AxisRobotArm
- RobotHand
- Tool
- Controller
- DigitalIO
- HMIComponent
- PhysicalObject
- Sensor
 - Acceleration1DSensor
 - Acceleration3DSensor
 - Acceleration6DSensor

Annotations: Kuka_IIWA

Annotations +

Description: Kuka_IIWA

Equivalent To +

SubClass Of +

- HardwareDevice
- hasCapability some ReachPose
- maxNumberOfAxis value 7
- MaxPayloadInKG value 1000
- 7AxisRobotArm

General class axioms +

SubClass Of (Anonymous Ancestor)

- maxNumberOfAxis value 7

Reasoner active ☒ Show Inferences

The screenshot shows the Protégé ontology editor interface. The top navigation bar includes tabs for Active Ontology, Entities, Classes, Object Properties, Data Properties, Annotation Properties, Individuals, OWLViz, DL Query, OntoGraf, Ontology Differences, and SPARQL Query. The left pane displays the 'Object property hierarchy: hasCapability' with a tree view of properties. The right pane is divided into three sections: 'Annotations: hasCapability' (empty), 'Characteristics: hasCapability' (with checkboxes for Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive), and 'Description: hasCapability'. The 'Description' section is highlighted with a red box and contains the following axioms:

- Equivalent To +
- SubProperty Of +
- Inverse Of +
- requiresCapability**
- Domains (intersection) +
- Ranges (intersection) +
- Action**
- Disjoint With +
- SuperProperty Of (Chain) +
- hasComponent o hasCapability SubPropertyOf hasCapability**

At the bottom right, the status bar indicates 'Reasoner active' and 'Show Inferences' is checked.

hw-taxonomy (http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy) : [/Users/zander/Projects/ReApp/svn/ont/hw-taxonomy/branch/2014-07-03_capability_components/HW-Taxonomy_...

hw-taxonomy (http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy)

Search for entity

Active Ontology | Entities | Classes | Object Properties | Data Properties | Annotation Properties | Individuals | OWLViz | DL Query | OntoGraf | Ontology Differences | SPARQL Query

Ontology header:

Ontology IRI <http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy>

Ontology Version IRI e.g. <http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy/1.0.0>

Annotations +

Ontology imports | Ontology Prefixes | General class axioms

General class axioms:

General class axioms +

- hasStartAngle some xsd:integer[> 0 , < 135] SubClassOf hasCapability some Navigation
- (hasCapability some GraspObject) and (hasCapability some ReachPose) SubClassOf hasCapability some MoveObject

General Class Axiom

RDF/XML rendering:

```
</members>
</rdf:Description>
<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.reapp.com/ontologies/2014/06/hardware_...>
    <rdf:Description rdf:about="http://www.reapp.com/ontologies/2014/06/hardware_...>
    <rdf:Description rdf:about="http://www.reapp.com/ontologies/2014/06/hardware_...>
  </members>
</rdf:Description>
</rdf:RDF>
```

Reasoner active ☒ Show Inferences

The screenshot shows the Protégé ontology editor interface. The left pane displays the 'Class hierarchy: ExampleWorkplace' with a tree structure. The right pane shows the 'Annotations: ExampleWorkplace' and 'Description: ExampleWorkplace' tabs. The 'SubClass Of' section in the description tab is highlighted with a red box, showing 'HardwareDevice' and two instances: 'hasComponent some Kuka_IIWA' and 'hasComponent some Schunk_WSG-50'. The bottom status bar indicates 'To use the reasoner click Reasoner->Start reasoner' and 'Show Inferences' is checked.

hw-taxonomy (http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy) : [/Users/zander/Projects/ReApp/svn/ont/hw-taxonomy/branch/2014-07-03_...]

hw-taxonomy (http://www.semanticweb.org/reapp/ontologies/2014/7/hw-taxonomy)

Search for entity

Classes | Object Properties | Data Properties | Annotation Properties | Individuals | OWLViz | DL Query | OntoGraf | Ontology Differences

Class hierarchy | Class hierarchy (inferred)

Class hierarchy: ExampleWorkplace

- CommunicationProtocol
- Environment
 - OperationEnvironment
- HardwareType
 - Actuator
 - Controller
 - DigitalIO
- HardwareDevice
 - Asus_Xtion_PRO_LIVE
 - AX-12A
 - ExampleWorkplace**
 - Kuka_IIWA
 - Schunk_WSG-50
 - Sick_LMS511-20100PRO
 - Sick_MiniTwin4
 - Sick_S30B-2011GA
 - Sick_TIM551-2050001
 - Sick_XYZ_Sensor
- HMIComponent
- PhysicalObject
- Sensor
- Measurement
 - Distance
 - RotationInDegrees
- OutputFormat
 - Acceleration1D
 - Acceleration3D
 - Acceleration6D

Annotations: ExampleWorkplace

Annotations +

Description: ExampleWorkplace

Equivalent To +

SubClass Of +

- HardwareDevice
- hasComponent some Kuka_IIWA
- hasComponent some Schunk_WSG-50

General class axioms +

SubClass Of (Anonymous Ancestor)

Members +

Target for Key +

Assign Components to Workplace

To use the reasoner click Reasoner->Start reasoner ☒ Show Inferences

Querying Complex Capabilities

The image shows three overlapping screenshots of a DL query interface. Each window has a title bar 'DL query:' and a search icon. The first window shows a query 'HardwareDevice and hasCapability some ReachPose' and an 'Execute' button. The second window shows a query 'HardwareDevice and hasCapability some GraspObject' and an 'Execute' button. The third window shows a query 'HardwareDevice and hasCapability some MoveObject' and buttons 'Execute' and 'Add to ontology'. The 'Query results' section of the third window is expanded, showing 'Direct sub classes (1)' with 'ExampleWorkplace' and 'Instances (0)'. A legend on the right of the third window lists: 'Direct super classes' (unchecked), 'Super classes' (unchecked), 'Equivalent classes' (unchecked), 'Direct sub classes' (checked), 'Sub classes' (unchecked), and 'Instances' (checked).

DL query: ⌵ ⌶ ⌷ ⌸

Query (class expression)

HardwareDevice and hasCapability some ReachPose

Execute

Query results

Direct sub class

- Examp
- Kuka_

Instances (0)

DL query: ⌵ ⌶ ⌷ ⌸

Query (class expression)

HardwareDevice and hasCapability some GraspObject

Execute

Query results

Direct sub class

- Examp
- Schun

Instances (0)

DL query: ⌵ ⌶ ⌷ ⌸

Query (class expression)

HardwareDevice and hasCapability some MoveObject

Execute Add to ontology

Query results

Direct sub classes (1)

- ExampleWorkplace ?

Instances (0)

- ☐ Direct super classes
- ☐ Super classes
- ☐ Equivalent classes
- ☒ Direct sub classes
- ☐ Sub classes
- ☒ Instances

Conclusion

- Relation of features to components via subsumption axioms
- Feature propagation via role composition axioms
- Computation of complex features via a combination of
 - Role composition axioms
 - General class inclusion axioms

Presenter

Dr. Stefan Zander
Tel: +49 721 9654-888
E-Mail: zander@fzi.de

Project Coordinator

Dr. Ulrich Reiser
Tel: +49 711 970-1330
E-Mail: ulrich.reiser@ipa.fraunhofer.de

Thank you for your Attention!

Questions?



For more information visit us at www.reapp-projekt.de

The research leading to these results has received funding from the German Federal Ministry for Economic Affairs and Energy (BMWi) under grant agreement n° 01MA13001A.