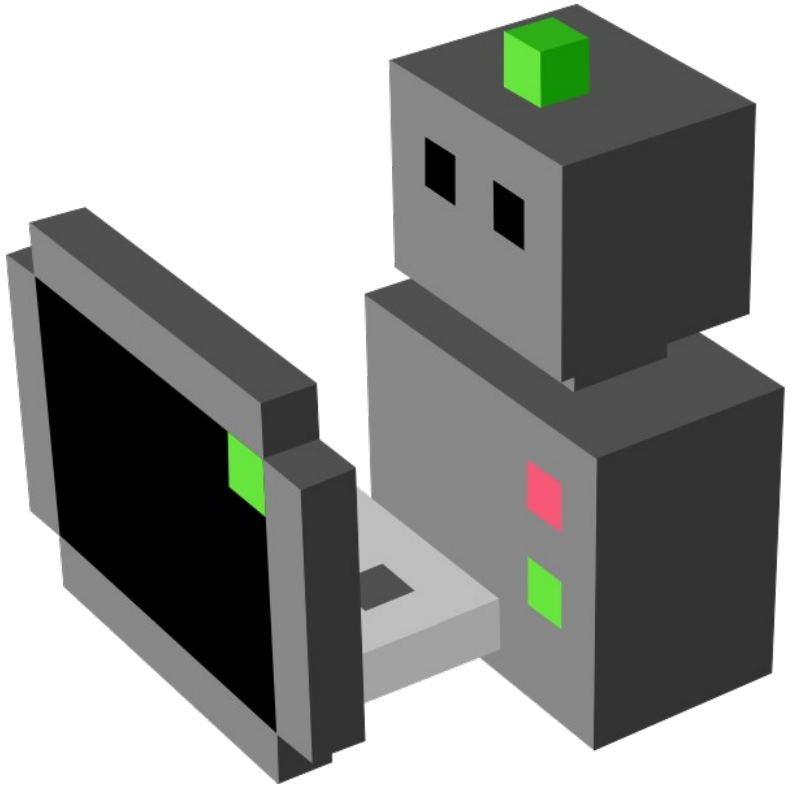
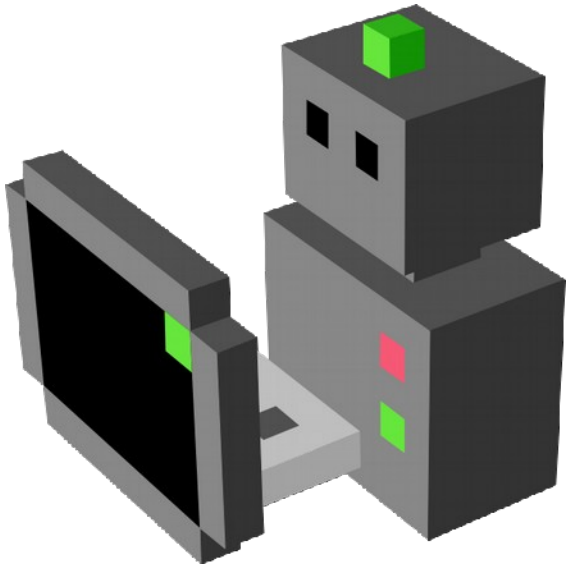


morse *for* hri

March 7th, 2014 – CITEC Bielefeld

HRI14
Human-Robot Interaction @ Bielefeld University

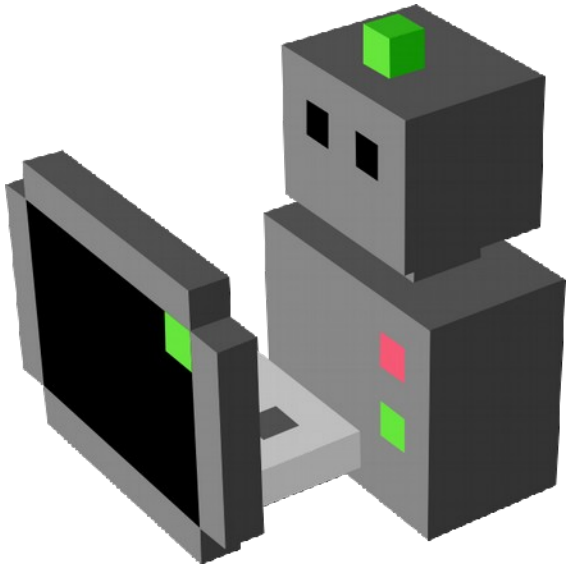


First, some demographics!

morse *for* hri

- Who has already run a simulation with MORSE?
- Who has already tried to simulate HRI scenarios (with any simulator)?

HRI14
Human-Robot Interaction @ Bielefeld University



MORSE?

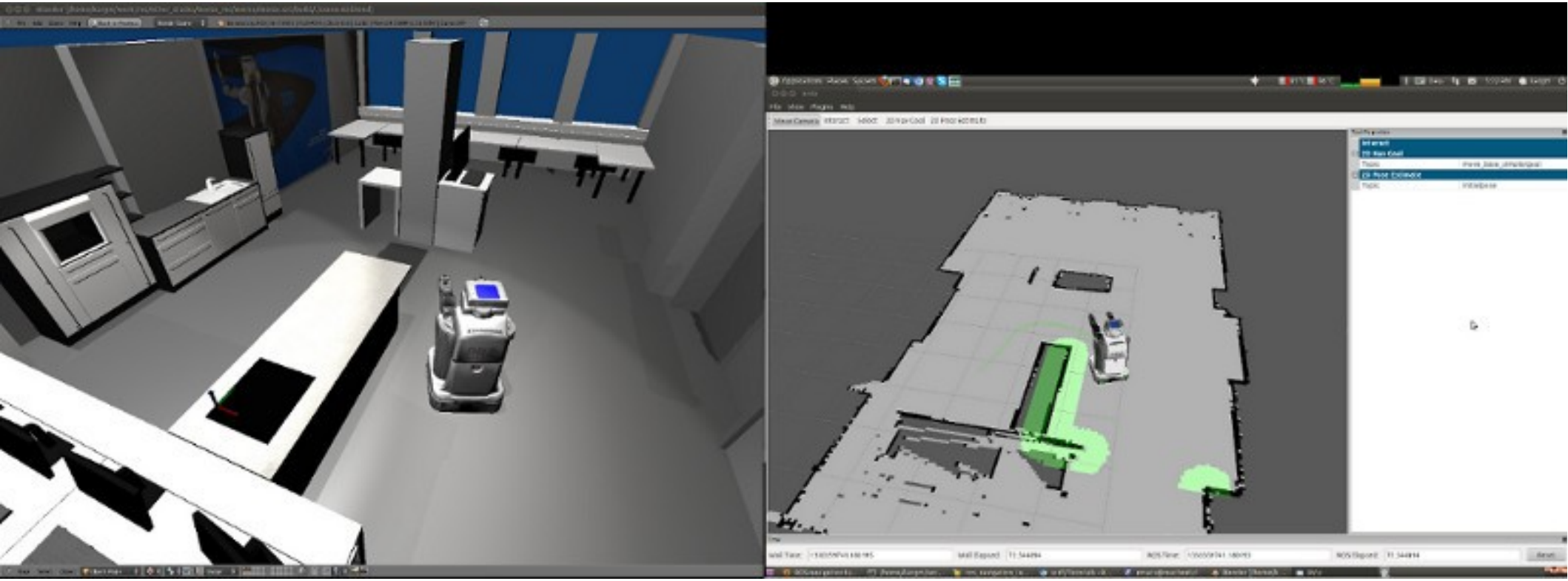
morse *for* hri

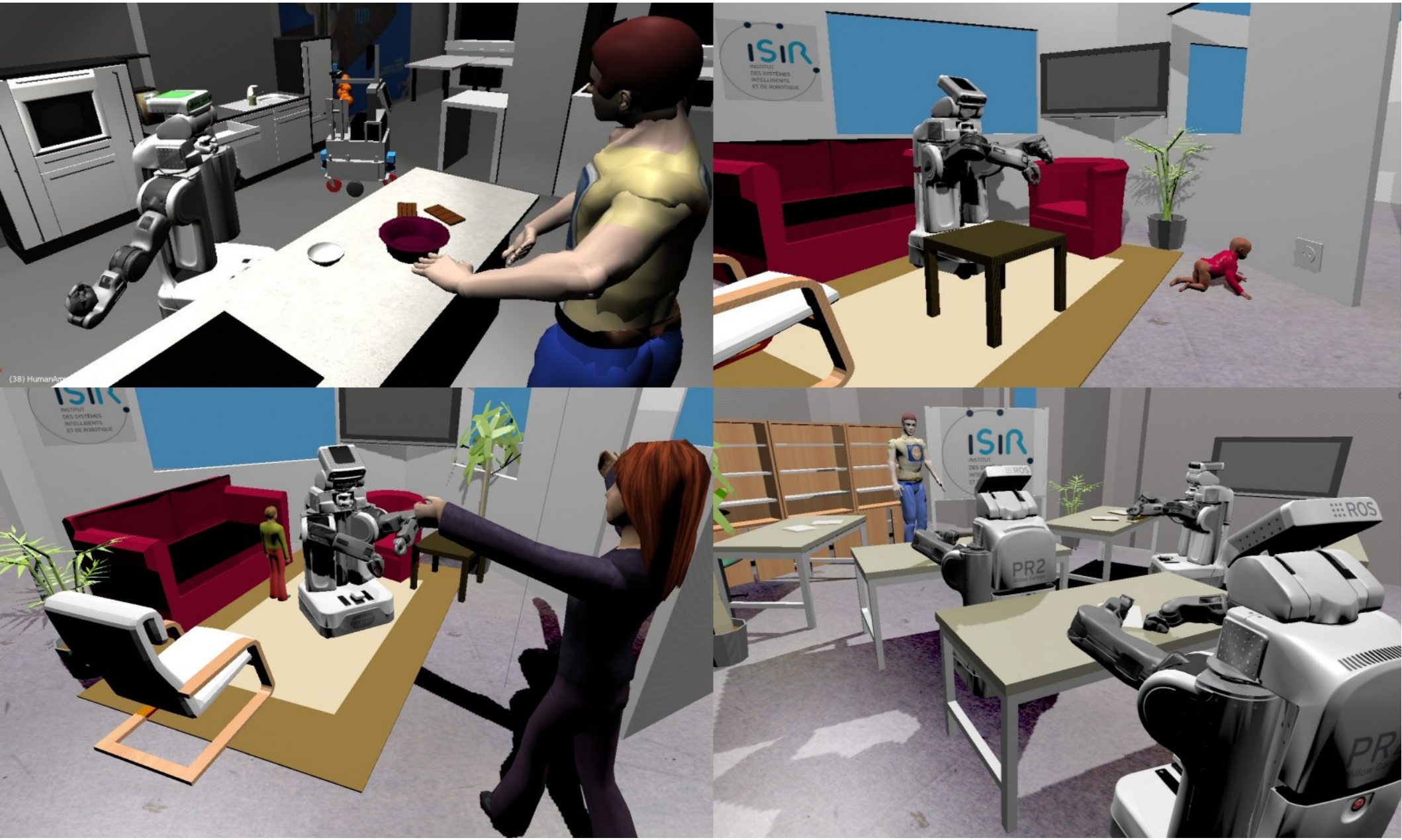
blender

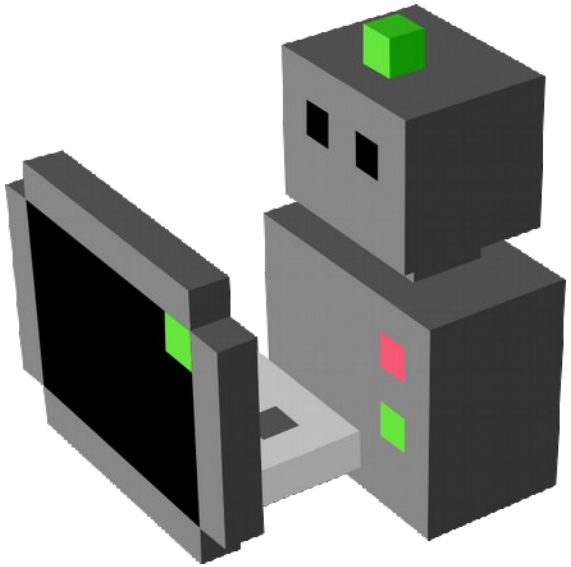
python powered

open source initiative
Approved License

HRI14
Human-Robot Interaction @ Bielefeld University




ISIR
INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE
ROS
PR2



MORSE?

morse *for* **hri**

ONERA
THE FRENCH AEROSPACE LAB

DLR

UNIVERSITY OF BIRMINGHAM

isae
Institut Supérieur de l'Aéronautique et de l'Espace

cnrs
dépasser les frontières

LAAS-CNRS

BOĞAZİÇİ ÜNİVERSİTESİ 1863

KATHOLIEKE UNIVERSITEIT LEUVEN

KIT
Karlsruher Institut für Technologie

IRD

CITEC
Cognitive Interaction Technology
Center of Excellence
Bielefeld University

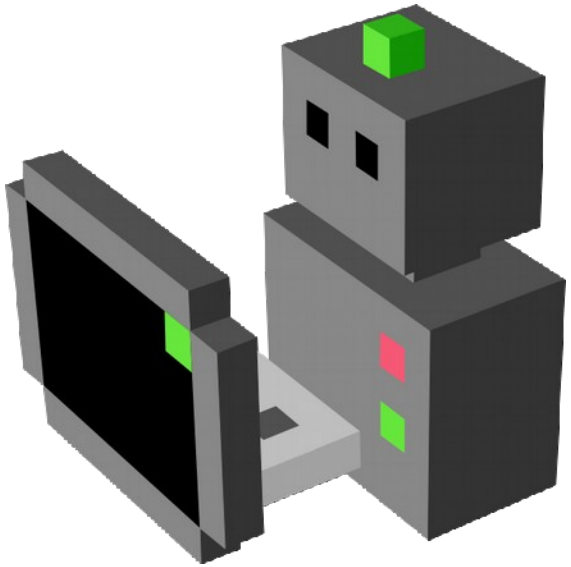
HOCHSCHULE
UNIVERSITY OF
APPLIED SCIENCES
INGOLSTADT

AUBURN
UNIVERSITY

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CORNELL UNIVERSITY
FOUNDED A.D. 1865

HRI14
Human-Robot Interaction @ Bielefeld University



By academics, for academics

morse *for* hri

- Best entry point: reporting **issues** & **improvements ideas**
- About 30KLOC of Python: easy to **hack on**!

github.com/morse-simulator/morse

HRI14
Human-Robot Interaction @ Bielefeld University

Simulation and HRI: Recent Perspectives

Séverin Lemaignan[1], Marc Hanheide[2], Michael Karg[3], Harmish Khambhaita[4], Lars Kunze[5], Florian Lier[6], Ingo Lütkebohle[7] and Grégoire Milliez[4]

***Abstract*— Simulation in robotics is often a love-hate relationship: while simulators do save us a lot of time and effort compared to regular deployment of complex software architectures on complex hardware, simulators are also known to evade many (if not most) of the real issues that robots need to manage when they enter the real world. Because humans are the paragon of dynamic, unpredictable, complex, real world entities, simulation of human-robot interactions may look condemn to fail, or, in the best case, to be mostly useless. This collective article reports on five independent applications of the MORSE simulator in the field of human-robot interaction: It appears that simulation is already useful, if not essential, to successfully carry out research in the field of HRI, and sometimes in scenarios we do not anticipate.**

I. Introduction

Simulation of human-robot interaction is a challenge: it lies at the crossroad of *robotic simulation* (which brings in requirements like physical accuracy, low latencies, high bandwidth, integration within complex software architectures) and *embodied virtual agents* (which also brings in its own requirements like realistic human kinematics and visual rendering, complex behaviors, rich user interface), while still remaining a *tool*: in order *not* to stand in the way of our daily development workflow, it must *feel* lightweight, responsive, easy to setup and deploy.

This paper presents how the Modular OpenRobots Simulation Engine (MORSE, figure 1) [3] simulator attempts to address this challenge to eventually provide a convenient support for research in human-robot interaction. We first give a brief overview of the simulator with a focus on HRI-specific features, and then report on several real-world applications. These five case-studies illustrate the collective nature of this article: we report on contributions and experiences in human-robot interaction simulation from five unrelated projects, conducted by different people in different universities and research institutes, only sharing the MORSE simulator as common simulation platform. The sections II-A to II-E present each of these projects, and try to highlight both the positive outcomes of deploying simulation environments for HRI, and the pitfalls and more fundamental issues that simulation of human-robot interaction still faces.

Fig. 1. Simulation and HRI: A PR2 and a human avatar in MORSE.

HRI and simulation

When we mention simulators in the following, we refer to simulation engines with support for physically simulating both robot kinematics and dynamics, as well as physical objects in the environment, and where the control and sensor interface is close to or the same as it is on real robots. In this paper, we beside only consider engines that support rendering and animation of human agents suitable for HRI, and which have been actually used in this context (while many engines could theoretically support this use case, few of them have actually demonstrated it). For the interested reader, a broader overview of research in robotic simulation is provided in [1].

Besides MORSE, simulators with explicit support for controlling a human agent include USARSim (Lewis *et al.* [11]), which is commonly used in the field of rescue robotics and in dozens of HRI studies, and, to a lesser extend, experiments with character animation have been developed in the widely-used simulator Gazebo (Koenig and Howard [7]).

All of these simulators are based on game engines (the Blender Game Engine, Unreal Tournament Development Kit and Ogre3D, respectively). While their built-in support for human models varies, all of these engines require the developer to specify the agents' behavior in detail. In practice, this limits human agent behaviors to relatively simple motions and interactions, and perhaps not surprisingly, most HRI simulations so far are carried out in *tele-operation* settings,

[1]S. Lemaignan is with Computer-Human Interaction for Learning and Instruction, École Polytechnique Fédérale de Lausanne, Switzerland

[2]M. Hanheide is with Lincoln Centre for Autonomous Systems, University of Lincoln, Lincoln, United Kingdom

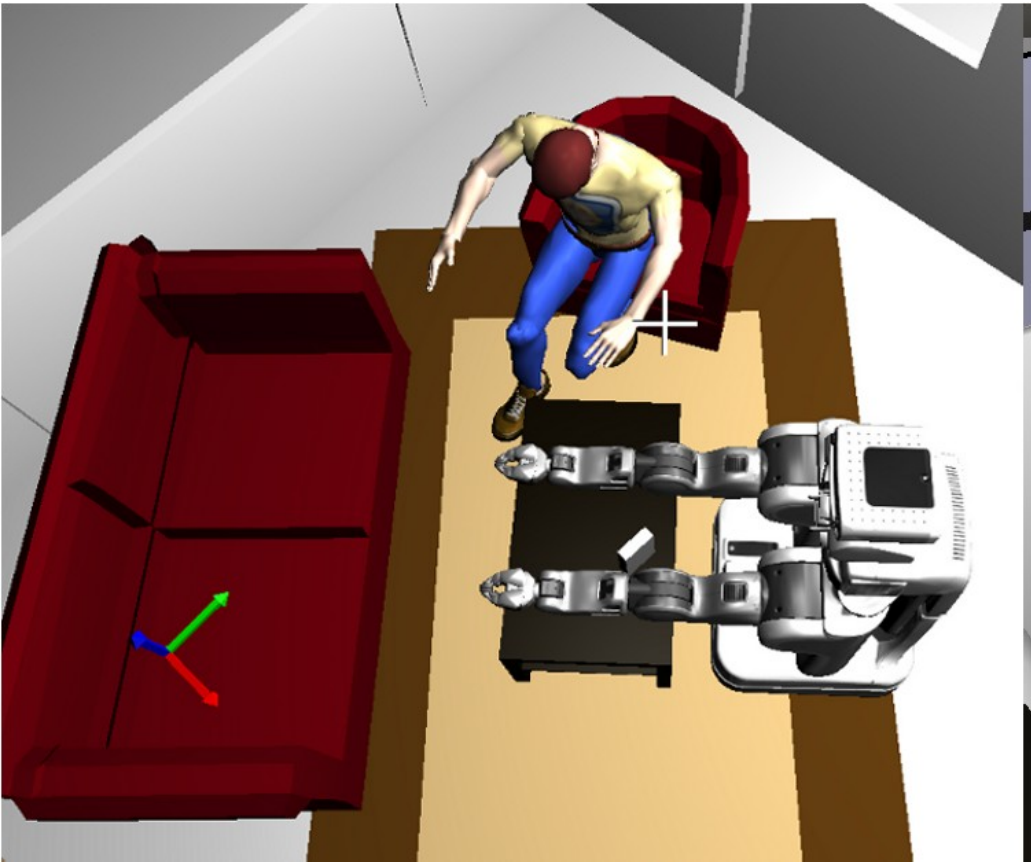
[3]M. Karg is with Institute for Advanced Study, Technische Universität München, Garching, Germany

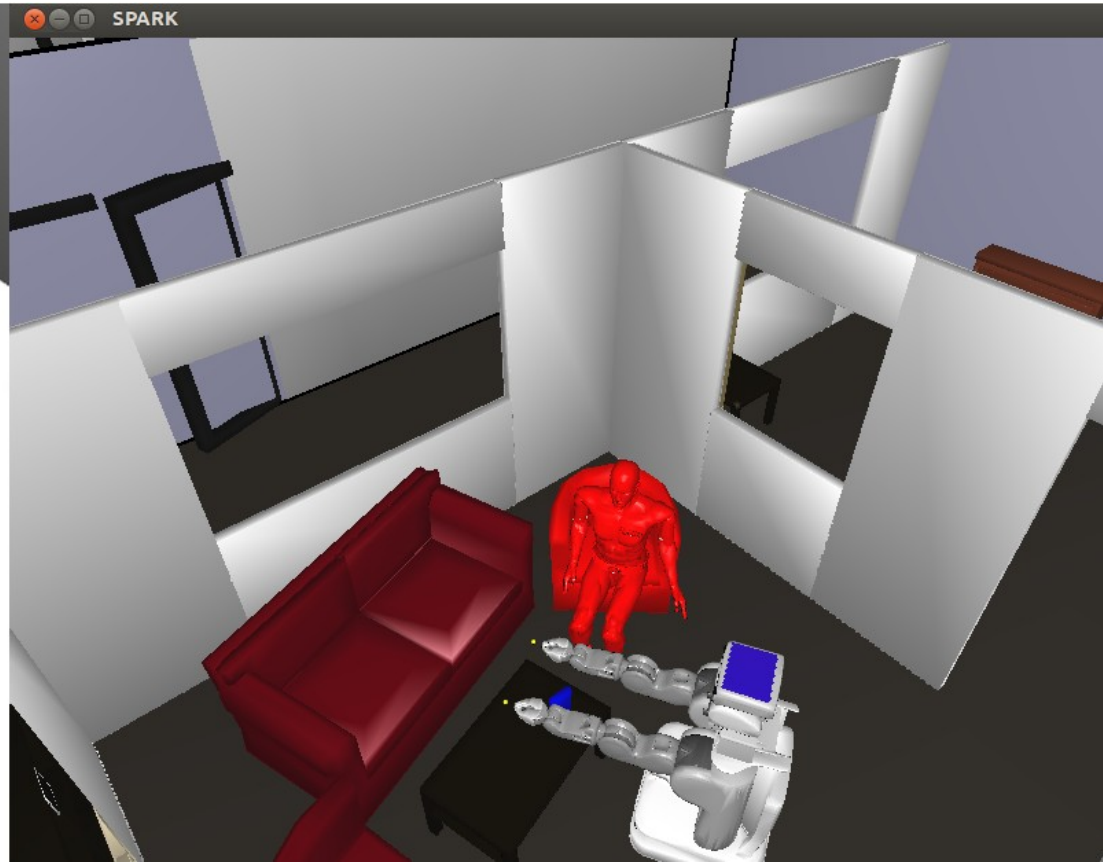
[4]H. Khambhaita and G. Milliez are with Laboratoire d'Analyse et d'Architecture des Système, Université de Toulouse, Toulouse, France

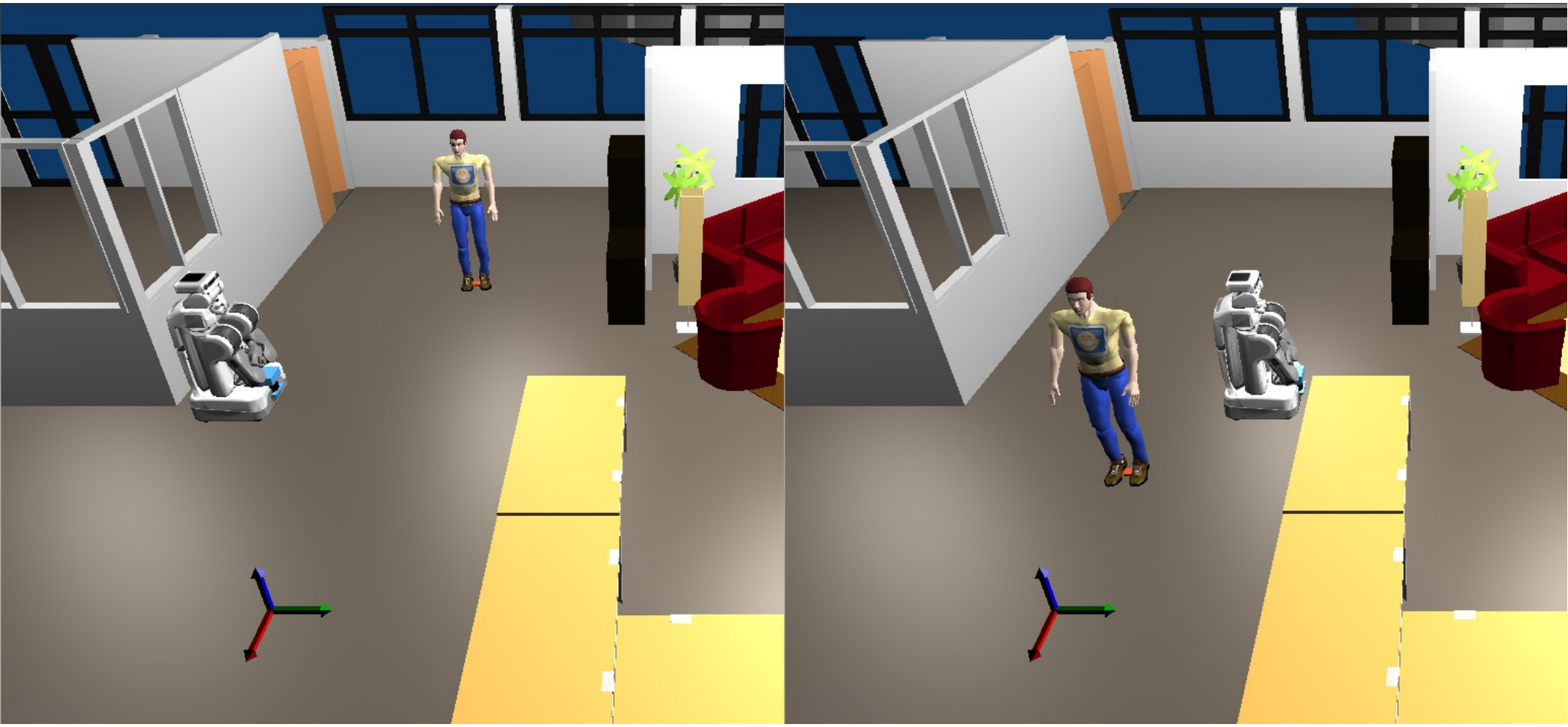
[5]L. Kunze is with Intelligent Robotics Lab, School of Computer Science, University of Birmingham, United Kingdom

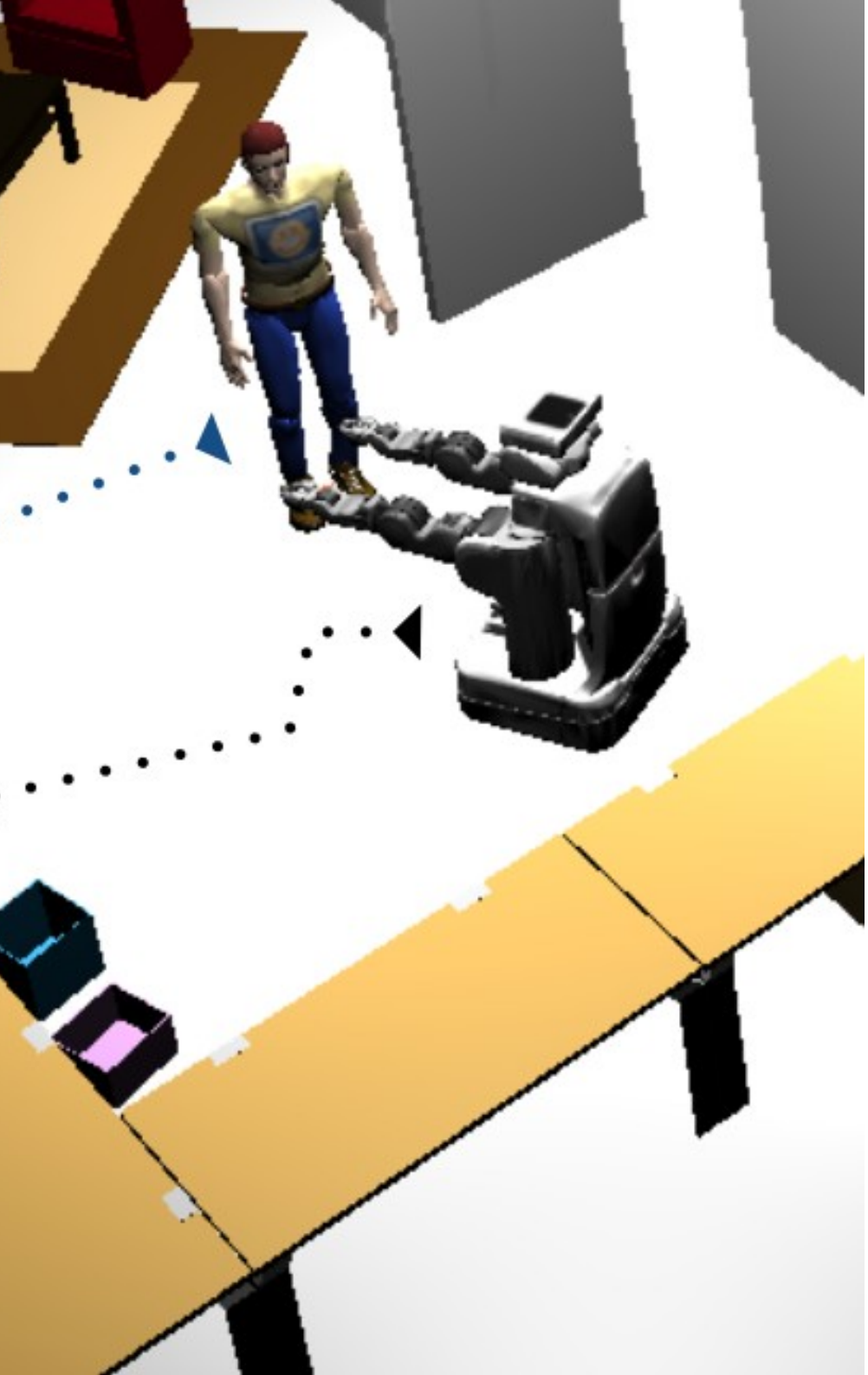
[6]F. Lier is with Cognitive Interaction Technology — Center of Excellence, Bielefeld University, Bielefeld, Germany

[7]I. Lütkebohle is with Machine Learning and Robotics Lab, Institute for Parallel and Distributed Systems, Universität Stuttgart, Germany




SPARK





1.0 DATAMODEL

- env: MORSE_ROOT = /opt
- component: roscore
- component: morse
- component: rosbag
- component: rostopic
- ...

```
<datamodel>
    <data id="environment" xmlns="http://my_fsm.c
        <variable var="PREFIX" val="/vol/robocup/
        <variable var="MORSE_ROOT" val="/vol/robo
        <!-- Further variables ... -->
    </data>
    <data id="software">
        <component val="roscore">
            <command val="roscore"/>
            <path val="/opt/ros/groovy/bin/"/>
            <executionHost val="localhost"/>
            <checkExecution val="True">
                <checkType val="pid" timeout="8"
                                     blocking="Tr
                                     ongoing="Tru
                <checkType val="stdout" timeout="
                                     criteria="st
                                     blocking="T
            </checkExecution>
        </component>
        <component id="morse"><!-- Morse content
        <!-- Further components ...-->
    </data>
</datamodel>
```

2.0 INIT

3.0 RUN

- Start roscore, check execution
- Start morse, check execution
- Parallel
  Start rosbag, check exec
  Start rostopic pub, check exec
- Wait N seconds
- Clean-up all components and fles

```
<state id="run_test" initial="ROS_startup">
    <!-- Allows to catch errors in sub-states -->
    <transition event="unsatisfiedCriteria" targe
        <log label="ERROR" expr="'recieved unsati
    </transition>
    <state id="ROS_startup">
        <onentry>
            <log label="INFO" expr="'Entering Sta
            <my_fsm:executeProgram value="roscore
        </onentry>
        <transition event="executeProgram.success
    </state>
    <!-- Further components to be executed ... --
    <state id="recordangles">
        <!-- ... --->
        <transition event="executeProgram.success
    </state>
</state>
```

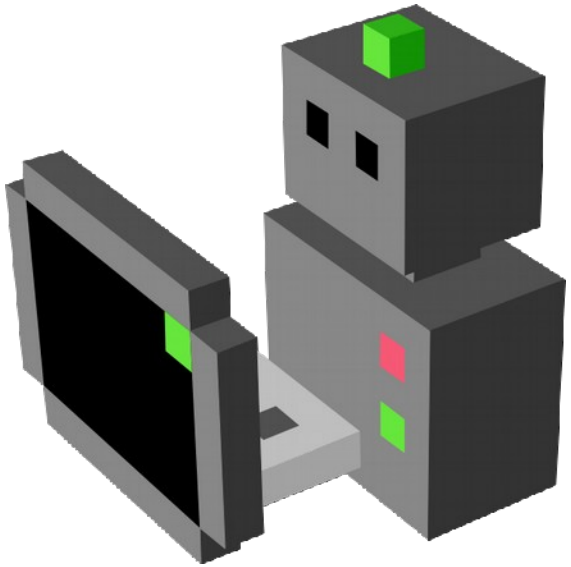
4.0 ASSESSMENT

- Start gnuplot, check execution
- Start convert log fles, check execution
- Wait N seconds

```
<state id="result_assessment">
    <initial>
        <transition target="clean"/>
    </initial>
    <transition event="unsatisfiedCriteria" targe
    <state id="clean">
        <onentry>
            <my_fsm:cleanUp expr="my_executable" v
        </onentry>
        <transition target="compute_results"/>
    </state>
    <!-- Assessment components are started here .
    <state id="waitfinish">
        <onentry>
            <send event="wait.finish" delay="'30s'
        </onentry>
        <transition event="wait.finish" target="ex
    </state>
</state>
```

5.0 Exit

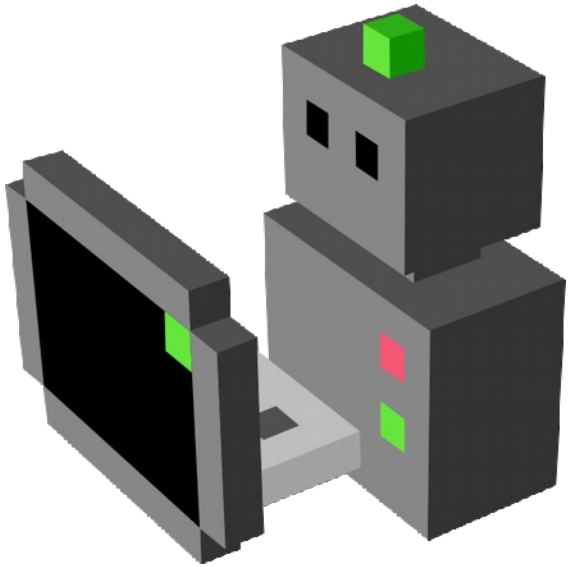
- Clean-up all components and fles



morse *for* hri

What next!?

HRI14
Human-Robot Interaction @ Bielefeld University



morse *for* hri

Agenda

- Florian's talk

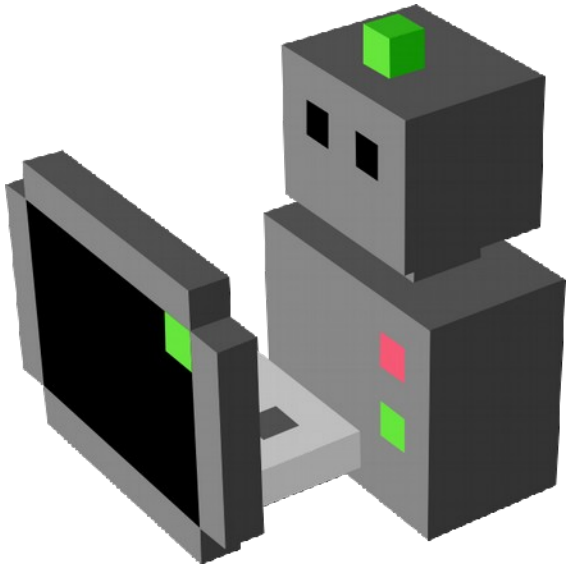
  break

- Marc's talk
- Open discussion: what road-map for MORSE in HRI?

  lunch

- Group work → *vote in a sec!*
- Synthesis

HRI14
Human-Robot Interaction @ Bielefeld University



Afternoon groups

morse *for* hri

A- Drafting features

a few ideas

- crowd simulation
- procedural generation of different models
- lightweight human models
- teleoperated human model(s)
  (Kinect-based, FPS-like or remotely - typically, over Internet)
- communication modalities
  (speech recognition/synthesis, nodding, blinks, gaze control...)
- scriptable behaviours

B- Tutorials

- Introduction to MORSE
- The human avatar in MORSE
- ...

HRI14
Human-Robot Interaction @ Bielefeld University