



KNOWLEDGE SHARING FOR ROBOTS

FLORIAN JOHANNßEN

Gliederung

2

- ☐ Motivation
- ☐ Vorarbeit
- ☐ Status Quo
- ☐ Masterthesis
- ☐ Risiken

Motivation

3

- Menschen lernen voneinander durch das Internet
- Warum nicht auch Roboter?
- Roboter sind WLAN fähig
- Roboter haben high-level Schnittstelle
- Roboter können ausführbare Pläne austauschen
- Wiederverwendung von Roboter-Apps

Knowledge sharing for robots

4

- Verbesserung der Lernmechanismen
- Profitieren von den Erfahrungen anderer
- Bewältigung von Aufgaben in unbekannten Umgebungen
- Roboter können Aufgaben bewältigen für die sie zur Entwicklungszeit nicht vorgesehen waren

Vorarbeit

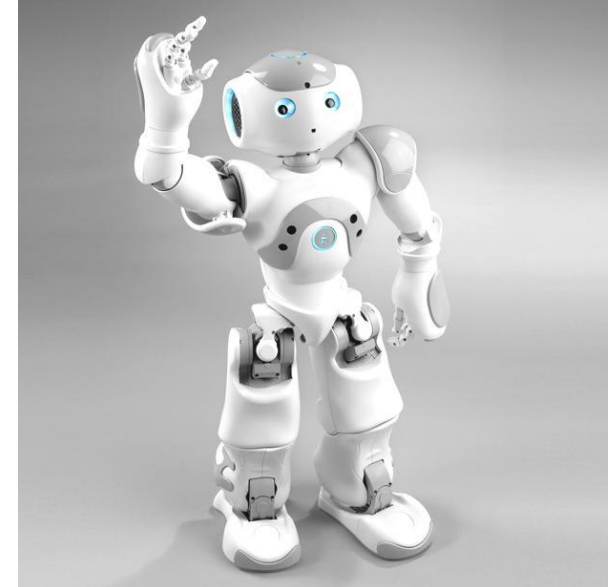
5

- Anwendungen 1
- Anwendungen 2
- Projekt 1

Anwendungen 1

6

- Recherche zum Thema Cloud Robotics [1]
- High-Level-Applikationen für Roboter (Nao)
- Plattformunabhängige Roboter-Apps
- Nao in the Cloud
- Lernmechanismen der Nao's verbessern



[2]

Anwendungen 2

7

□ Analyse

▣ RoboEarth [3]

- World Wide Web für Roboter
- Austausch von abstrakten Roboterplänen möglich

▣ RoboShare [4]

- Suchmaschine für Roboter
- Schwerpunkt liegt auf der effizienten Indizierung von Daten

▣ DaVinCi [5]

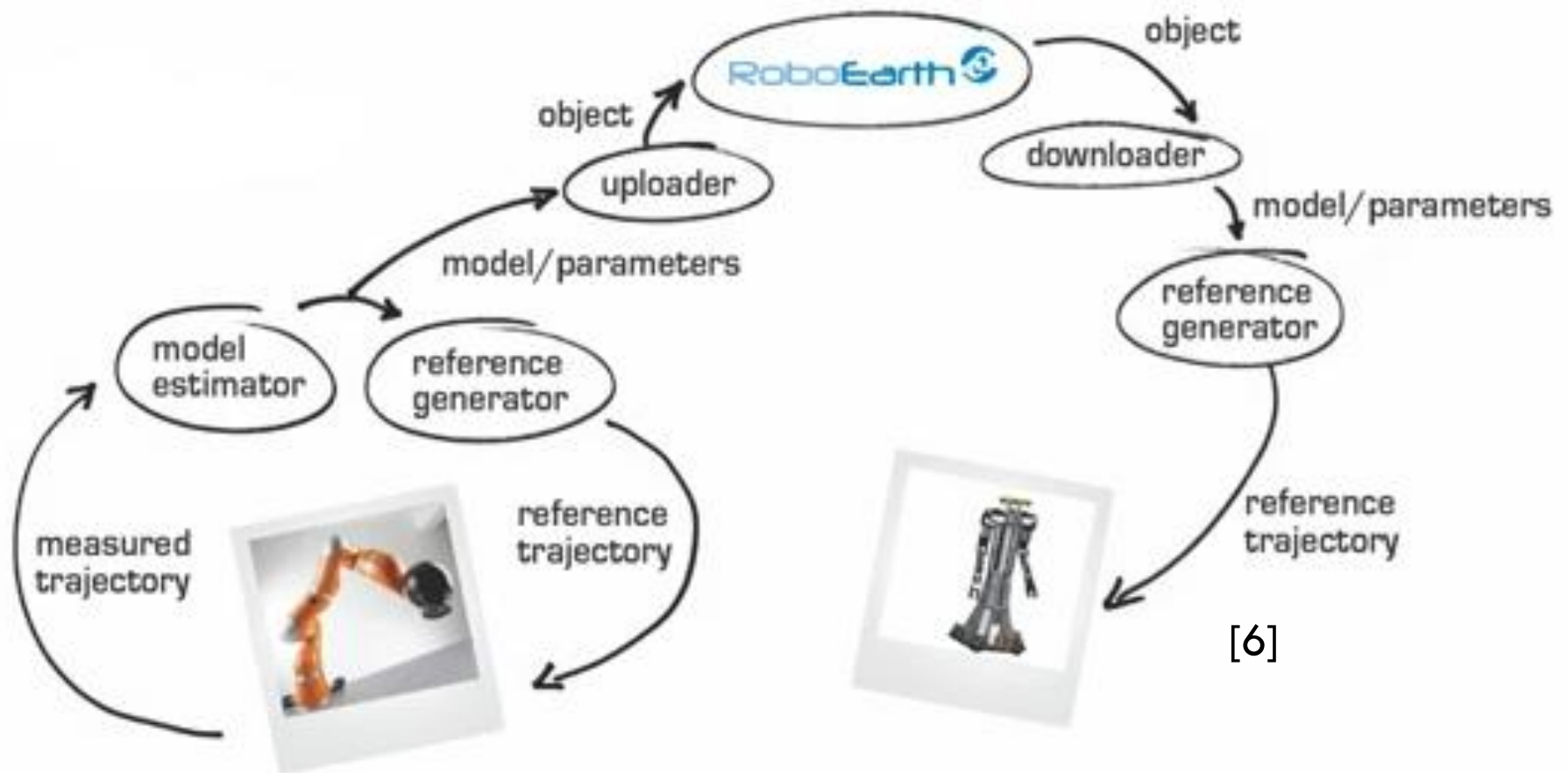
- Bietet keinen Austausch von ausführbaren Plänen
- Schwerpunkt liegt auf der Auslagerung von Algorithmen

□ Evaluierung

- ▣ RoboEarth → geeignetster Ansatz

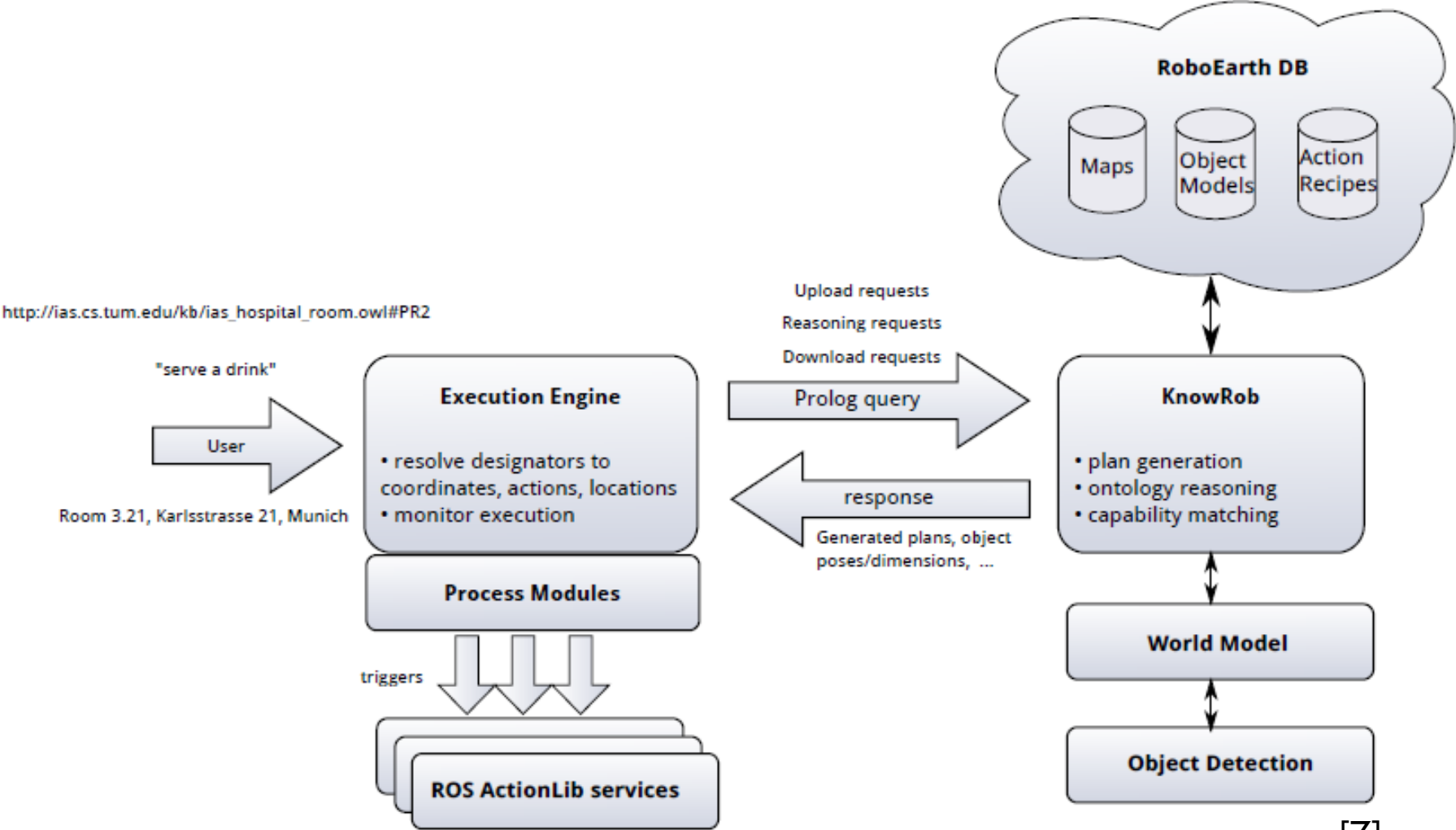
RoboEarth

8



Architektur

9



Execution Engine

10

- Pläne sind als OWL-Files definiert
- Transformation von abstrakten Plänen in ausführbare Routinen
- Basiert auf der CRAM Plan Language
- Delegiert die semantischen Anfragen zum Knowledge Processing Framework KnowRob
- Prozessmodule sind die Schnittstellen zu den physikalischen Roboter Routinen → Ros Actionlibrary

Beispiel

11

ServeADrink

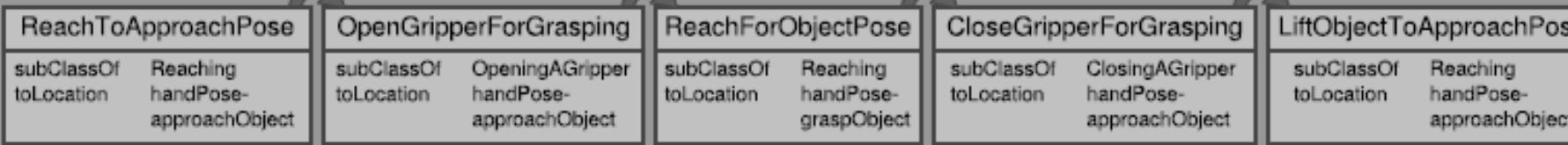
dependsOnComponent ObjectRecognitionModel AND providesModelFor.Bottle
dependsOnComponent ObjectRecognitionModel AND providesModelFor.Bed
dependsOnComponent ObjectRecognitionModel AND providesModelFor.Cabinet



reduce

GraspBottle

subClassOf GraspingSomething
objectActedOn bottle1



Status Quo

12

- Nao + ROS
 - ▣ Robotic Operation System (ROS)
 - ▣ NaoQi SDK
 - ▣ Nao-ROS-Stack
- ROS + RoboEarth
 - ▣ KnowRob-ROS-Stack
 - ▣ RoboEarth-ROS-Stack

Infrastruktur

13

```
graph TD; Ubuntu[Ubuntu] --> ROS[ROS]; Ubuntu --> NaoQi[NaoQi]; ROS --> KnowRob[KnowRob]; ROS --> RoboEarth[RoboEarth]; ROS --> NaoDriver[NaoDriver];
```

Ubuntu

ROS

NaoQi

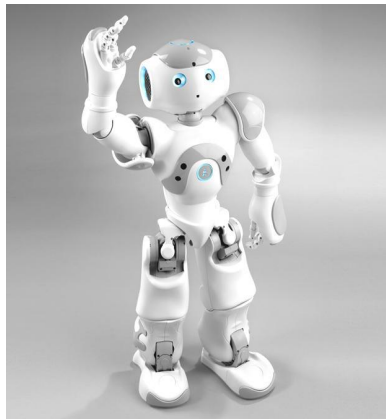
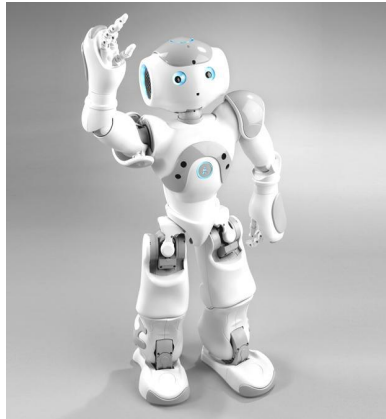
KnowRob

RoboEarth

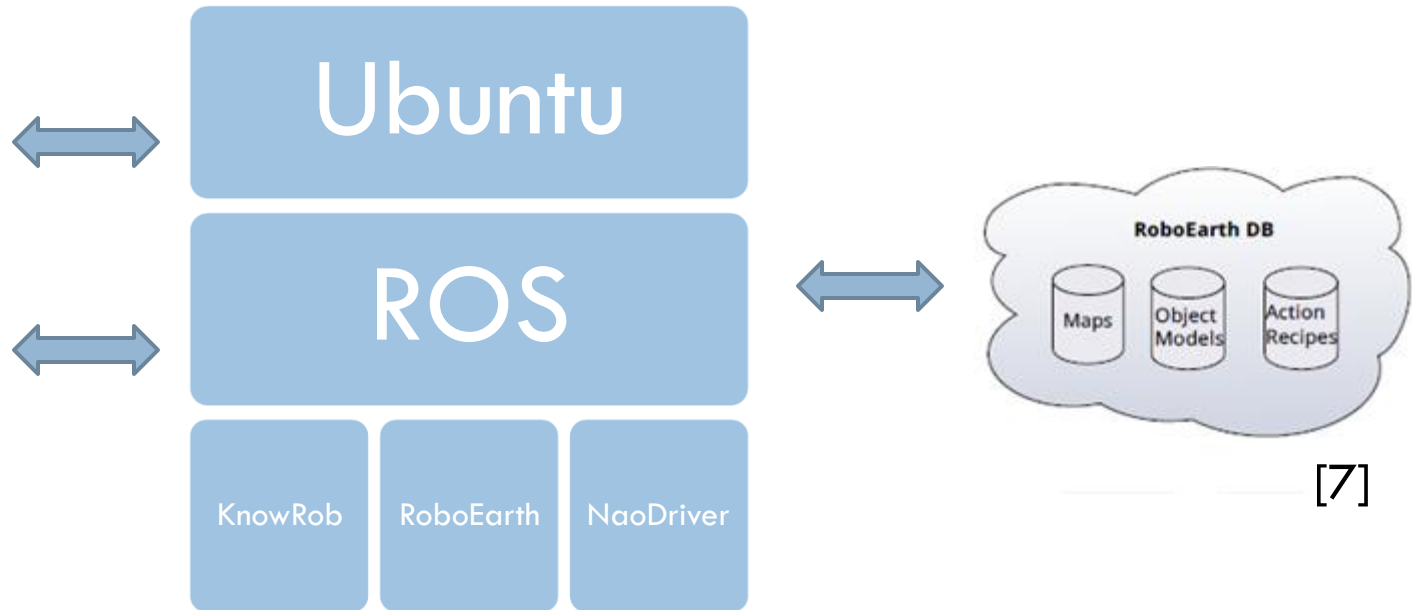
NaoDriver

Infrastruktur

14



[2]



[7]

KnowRob - Ontologie

15

□ Abfrage von Subtypen

Example

```
$ rosrun rospirolog rospirolog knowrob_tutorial

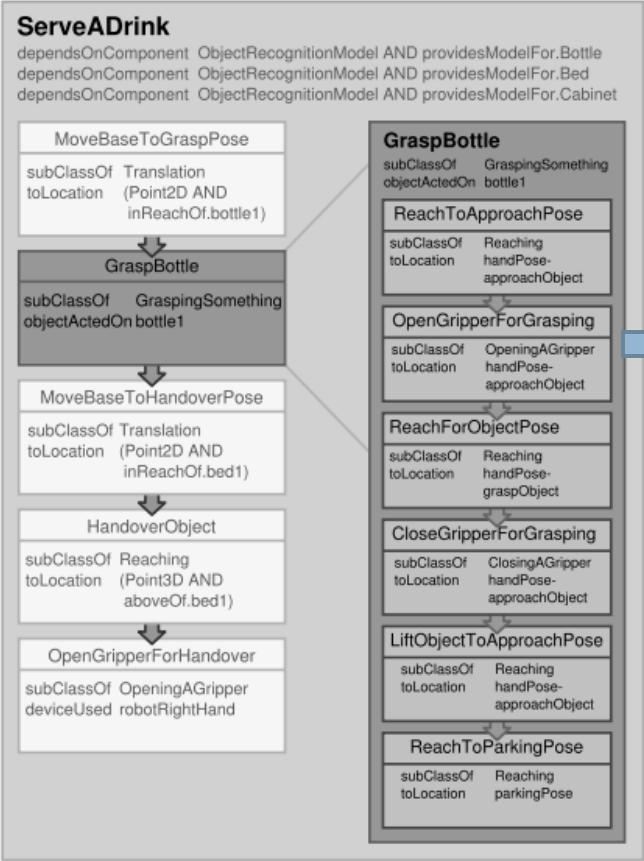
?- owl_subclass_of(A, knowrob:'FoodOrDrink').
A = 'http://ias.cs.tum.edu/kb/knowrob.owl#FoodOrDrink' ;
A = 'http://ias.cs.tum.edu/kb/knowrob.owl#Drink' ;
A = 'http://ias.cs.tum.edu/kb/knowrob.owl#Coffee-Beverage' ;
A = 'http://ias.cs.tum.edu/kb/knowrob.owl#InfusionDrink' ;
A = 'http://ias.cs.tum.edu/kb/knowrob.owl#Tea-Beverage' ;
A = 'http://ias.cs.tum.edu/kb/knowrob.owl#Tea-Iced'
```

[8]

KnowRob - Request

16

re_download_action_recipe('serve a drink', nao: 'MyNao',
Recipe), re_generate_cpl_plan(Recipe, CplPlan).



```
(def-top-level-plan
  serve-a-drink()
  (with-designators(
    (bottle1(object'((name bottle1)
      (type drinking-bottle))))
    (bed1(object'((name bed1)
      (type bed piece-of-furniture))))
    ...
```

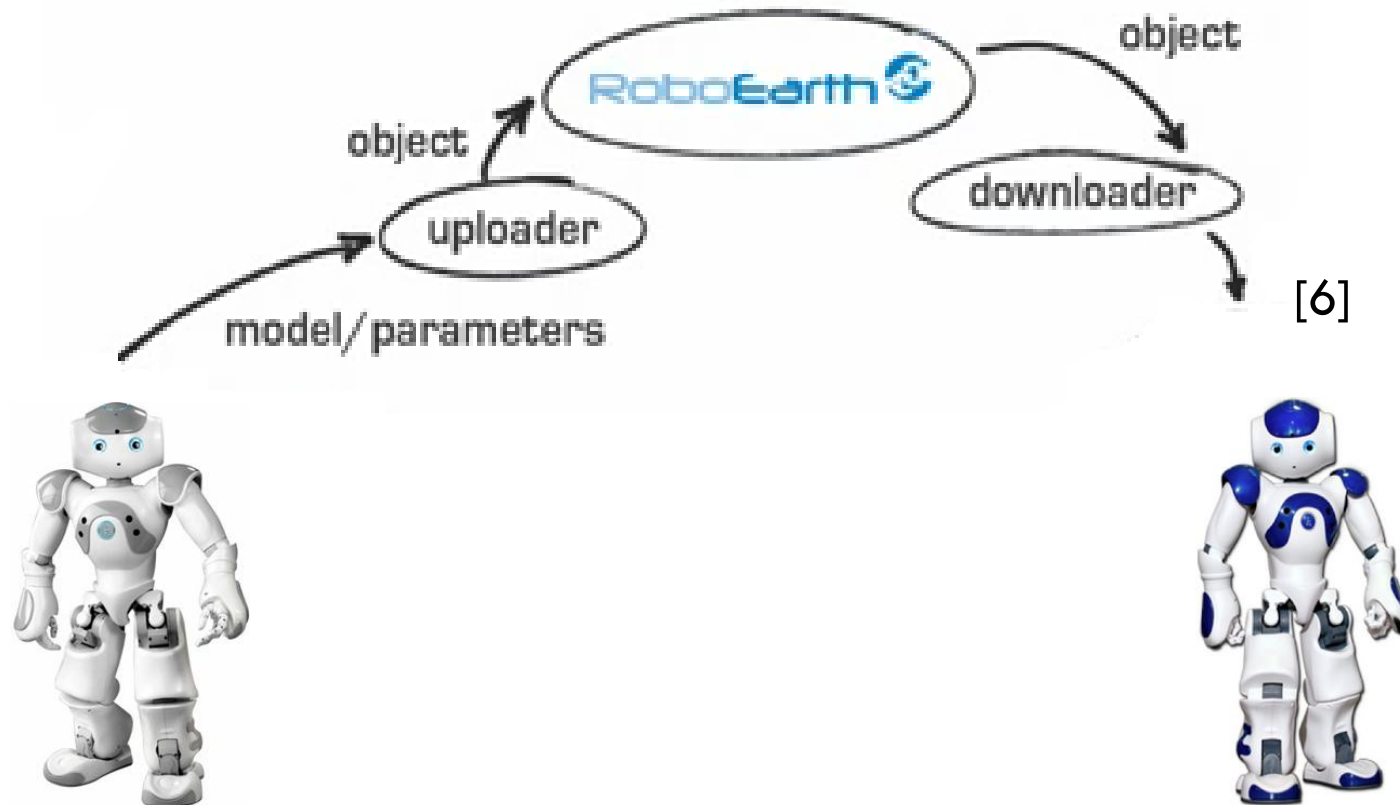
[8]

[7]

Masterthesis

17

- Austausch von ausführbaren Plänen zwischen mehreren Nao-Robotern

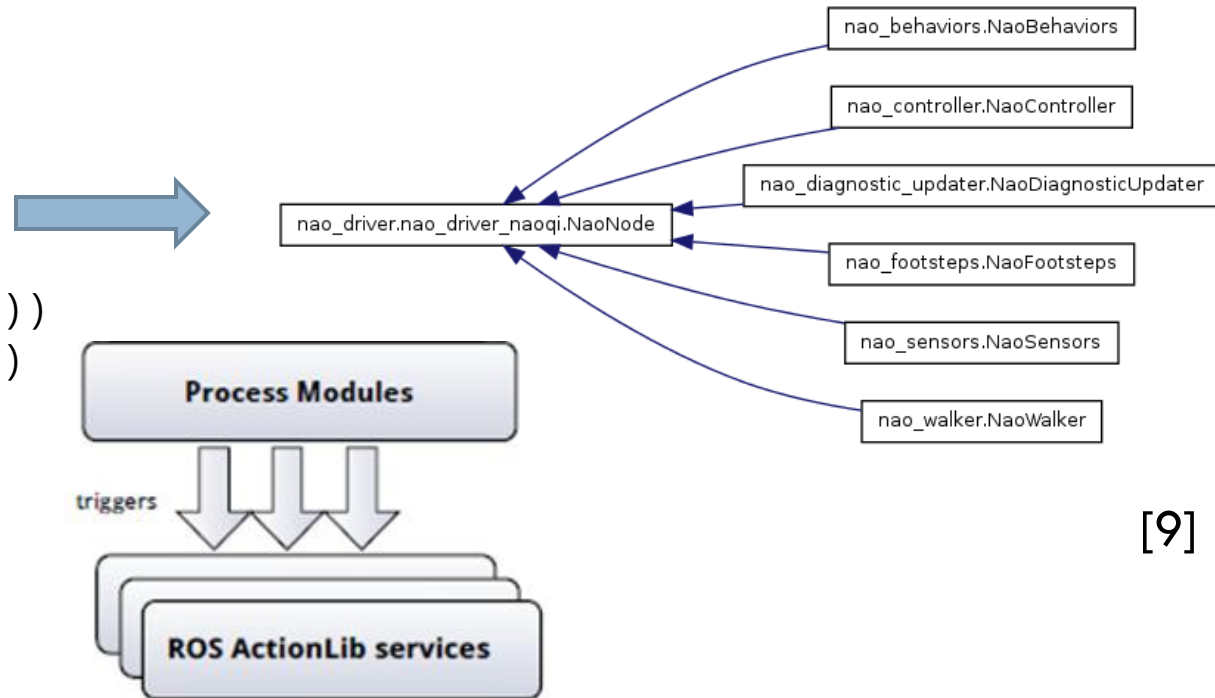


Schnittstelle

18

- Schnittstelle zwischen NAO und RoboEarth über ROS entwickeln

```
(def-top-level-plan
  serve-a-drink()
  (with-designators(
    (bottle1(object'((
      (name bottle1)
      (type drinking-bottle))))
    (bed1(object'((name bed1)
      ...
```



[9]

Szenario 1

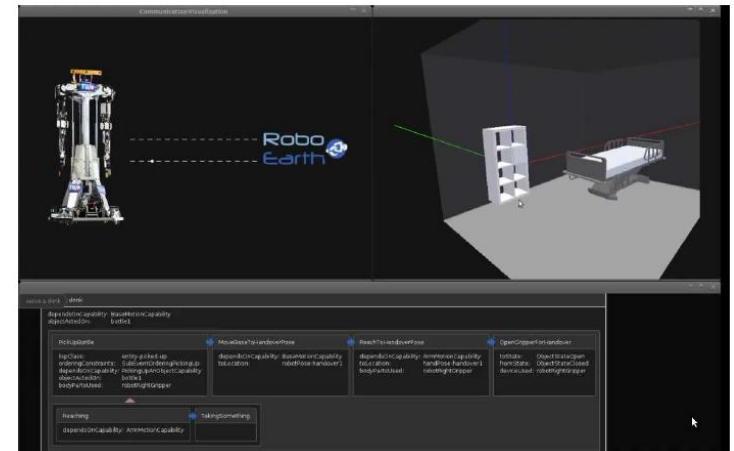
19

- Nao erlernt durch Interaktion mit dem Menschen einen bestimmten Plan
- Nao transformiert erlernte Informationen in ein CRAM Plan Language-fähiges Format
- Nao lädt Informationen in die RoboEarth Datenbank

Szenario 2

20

- Ein anderer Nao muss dieselbe Aufgabe bewältigen
- Download des ausführbaren Plans über KnowRob
- Ausführung des Plans
- Simulation mittels `visualisation_canvas()`



Evaualierung

21

- Performanz messen
- Korrektheit prüfen
- Vision: Appstore für Nao mithilfe von RoboEarth

Risiken

22

- Abhängigkeiten von RoboEarth und dem Nao SDK
- Viele Schnittstellen
- Probleme bei den Transformationen
- Zu hohe Latenzzeiten beim Download

Quellen

23

1. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05719709>
2. <http://www.generationrobots.de/site/news/Robot-NAO.png>
3. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5876227>
<http://www.doc.ic.ac.uk/~xf309/Misc/IJHR.pdf>
4. http://vikasreddyenti.com/wp/wp-content/uploads/2010/06/DAvinCi-CloudComputingRobots_final.pdf
5. <http://www.youtube.com/watch?v=YjI6vaWdseI>
6. <http://as.cs.tum.edu/media/spezial/bib/ias12execution.pdf>
7. [http://mirror.umd.edu/roswiki/attachments/Events\(2f\)CoTeSys\(2d\)ROS\(2d\)School/knowrob.pdf](http://mirror.umd.edu/roswiki/attachments/Events(2f)CoTeSys(2d)ROS(2d)School/knowrob.pdf)
8. http://ros.org/doc/groovy/api/nao_driver/html/inherits.html
9. <http://ias.cs.tum.edu/media/spezial/bib/tenorth09knowledge.pdf>
10. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6224858>
11. <http://ias.cs.tum.edu/~tenorth/thesis.pdf>
12. <http://common-lisp.net/~trittweiler/bachelor-thesis.pdf>
13. <http://homepages.laas.fr/slemaign/wiki/lib/exe/fetch.php?media=workshop-ias-laas-dec2010-slides:moesenlechner.pdf>
14. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6224812>



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

VIELEN DANK
GIBT ES NOCH FRAGEN?