Extended abstract for the Nao Tech Day @ Aldebaran Robotics

Nao Tech Day 2013

Paris, France, 4 Juin 2013

# Nao in the Cloud

## Knowledge sharing for robots via cloud services

*Florian Johannßen*
Department of Computer Science
University of Applied Sciences Hamburg
Hamburg, Germany
florian.johannssen@haw-hamburg.de

***Abstract*—This paper deals with the subject area knowledge sharing for robots. It demonstrates how robots can improve their learning mechanism via Cloud Computing and it shows how heterogeneous robots can exchange information about plans, objects and environments among each other. This work introduces our current research which realizes the approach with the aid of the humanoid robot Nao and the cloud service RoboEarth. The practical part implements an interface between the Nao and the cloud service and demonstrates a use case in which several Nao robots download and execute abstract plans from RoboEarth. So Robots are no longer on their own - they can benefit from the experience of other robots.**

***Keywords*—Knowledge sharing; Cloud Robotics; RoboEarth; Robot Operating System; Nao; Cloud Computing**

## I. Introduction

The Internet has become one of the most important communication media. It provides the opportunity to publish and retrieve knowledge globally. We are able to solve unknown tasks efficiently and share knowledge with other people. If you are involved in the research area of robotics, it would be preferable to apply this paradigm to robots. Nowadays, companies like Aldebaran Robotics[1] and Willow Garage[2] are able to deliver wireless capable and programmable robots with abstract interfaces. Specific tasks, such as face recognition, voice recognition and path planning are mostly solved. Thus the preconditions have been created to connect robots with the internet. Kuffner [1] and Quintas et al. [2] have introduced the topic Cloud Robotics.This idea provides a physical separation between the hardware and software components of the robot. Conventional hardware devices of a robot, such as sensors, actuators, cameras and speakers are still on the robot. The difference to the usual approach, which designs the software on the robot, is that the brain of the robot is outsourced to remote servers. This consideration to outsource complex and computationally intensive operations to remote servers has already been researched in the 1990s by Inaba [3]. His idea was to build robots without a brain. At this time there were no processors as nowadays, which are powerful and space saving at once. Inaba tried to minimize the needed hardware resources. In the approach Cloud Robotics represents the robot a client that consumes services from the cloud infrastructure and the server-side specifies a cloud of servers. This idea can be used as Inaba [3] to outsource time consuming tasks on powerful remote servers. In addition, it offers the possibility that robots communicate with each other to improve their learning mechanism. The idea of knowledge sharing for robots describes the problem how to exchange information between heterogeneous robots to benefit from the experience of others.

## II. Related Work

Recently, more and more companies and universities are interested in researching the topic of knowledge sharing for robots via Cloud Computing. The work of Arumugam et al. [4] presents DaVinCi, a Cloud Service for robots to swap out computationally intensive robot algorithms like image processing or path planning to remote servers. DaVinCi uses the Map-Reduce method to parallelize common robotics algorithms. It focuses on the aggregation of sensor data from heterogeneous robots to create global maps. Another similar approach by Fan and Handerson [5] developed a search engine like Google for robots. Robots send requests to the knowledge repository RobotShare for information about objects and plans. The knowledge is efficiently indexed. When robots request the web repository for something like how to wash cookware, they get the URL[3]-reference of the knowledge. Thereby the received information is defined in natural language and isn`t executable. Zweigle et al. [6] introduced RoboEarth which represents a World Wide Web for robots. This cloud service can be used by heterogeneous robots to share information among each other. RoboEarth isn´t only a database for knowledge sharing. Beetz, Mösenlechner and Tenorth [7] have developed the CRAM[4] System which

[1] http://www.aldebaran-robotics.com
[2] http://www.willowgarage.com

[3] Uniform Resource Locator
[4] Cognitive Robotic Abstract Machine

includes mainly the Cognitive Robotic Abstract Machine Plan Language and KnowRob for knowledge processing and reasoning. With the aid of CRAM, robots are able to send semantic requests for a plan to the RoboEarth database. The CRAM System translates the abstract plan like *grasp a bottle* to the CRAM Plan Language, which is a Lisp-like language for the implementation of abstract plans. If a robot is compatible with the open source middleware ROS[5] [8], it can execute CRAM-Plans. Another approach is to use the Google object recognition engine from Google Goggles[6]. Usually a Smartphone can send an unknown picture to this service to get information about it. Kehoe et al. [9] realize a cloud-based robot grasping system with the aid of Google Goggles.

## III. Nao in the Cloud

One of the main problems of the robots is, that they are limited in many ways. The computing power and hardware resources are bound by costs and their physical properties. Moreover, robots are usually on their own and only programmed for one specific area. So the behavior of the robot is greatly limited by inflexible programming. The standard SDK[7] of a robot doesn´t provide solutions for sharing executable information with heterogeneous robots. The next figure presents an architecture which provides a knowledge sharing among different robot platforms with the aid of cloud services to solve this problem.

**Figure 1: Robots in the cloud [10,13,14]**

Firstly, we implement this scenario only with the Nao robot to avoid problems because of the heterogeneity of the robots. The required components and middlewares which provide a cloud-enabled Nao are already installed on each laptop. The robots communicate with the cloud by accessing this infrastructure on a remote laptop. RoboEarth stores the information in an ontology web language-like format. If a robot sends a semantic request for a plan like *grasping a bottle* to RoboEarth, this abstract plan will be translated to the Cognitive Robotic Abstract Machine Plan Language. The ROS interface of the robot provides the execution of such a CRAM-plan. The next figure shows the components and middlewares which are installed on the laptop and provides an architecture to connect the Nao robot with the cloud service RoboEarth.

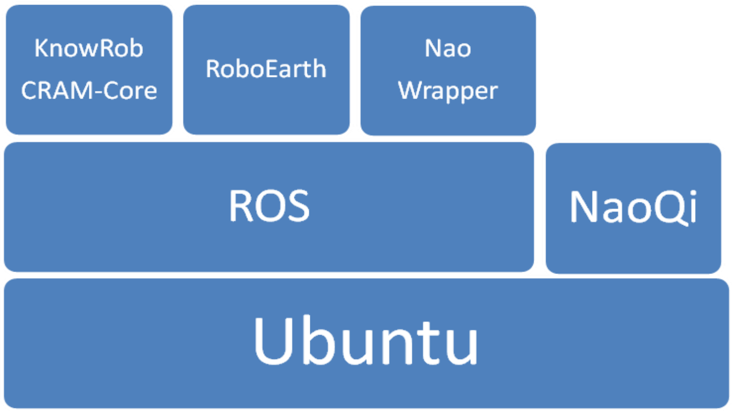


**Figure 2: Architecture**

Figure 2 includes the required ROS-Stacks Nao-Wrapper, RoboEarth, CRAM-Core and KnowRob which are needed to connect the Nao robot with the RoboEarth cloud service. NaoQi SDK represents a programming interface to control the Nao robot. The CRAM-Core stack contains the CRAM Plan Language to develop abstract control programs. It includes mechanisms to access the RoboEarth database and to describe meta information about objects and plans.

```
def-top-level-plan grasp-bottle ()
```

**Listing 2: CRAM-Plan [15]**

This code must be translated to the Nao specific commands of the Nao-Wrapper stack to execute it on the robot.

## IV. Vision

The concrete target of our current research [16] deals with the realization of the still unexplored approach of knowledge sharing for robots via cloud services. It explores how heterogeneous robots are able to improve their learning behavior by exchanging knowledge via cloud services and it presents the challenges which have to be solved. Firstly, the practical part of the work will handle this topic with two homogenous robots which are connected with a cloud service for downloading and executing robot plans. The Nao robot and the cloud service RoboEarth will be used for the implementation. The practical part will include the development of an interface between the humanoid robot Nao and the RoboEarth cloud service as well as the realization of a scenario in which several Nao robots download and execute information from the cloud service. This work provides an architecture which realizes the preconditions to connect the Nao robot with the cloud.

---

[5] Robot Operating System

[6] http://www.google.com/mobile/goggles

[7] Software Development Kit

## *References*


[1] Kuffner. Robots with their Heads in the cloud. 2011.

[2] J. M. Quintas, P. J. Menezes, J. M. Dias. Cloud Robotics: Towards context aware Robotic Network. 2011

[3] M. Inaba. Remote Brained Robots. Tokio, 1993

[4] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. S. Kumar, K. D. Meng, G. W. Kit. DAvinCi: A Cloud Computing Framework for Service Robots. 2010

[5] X. Fan, T. C. Henderson. RobotShare: A Google for Robots. 2007.

[6] O. Zweigle, R. Molengraft, R. Andrea, K. Häussermann. RoboEarth – connecting Robots worldwide. Eindhoven, 2009

[7] M. Beetz, L. Mösenlechner, M. Tenorth. CRAM – A Cognitive Robot Abstract Machine for Everday Manipulaiton in Human Enviroments.

[8] M. Quigley, B. Gerkey, K. Conley, J. Fausty, T. Footey, J. Leibs, E. Berger, R. Wheeler, Ng.ROS: an open-source Robot Operating System.

[9] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, K. Goldberg. Cloud-Based Robot Grasping with the Google Object Recognition Engine.2013

[10] Aldebaran-Robotics. NAO H25 Humanoid Robot Platform. 2012

[11] M. Di Marco, M. Tenorth, K. Häussermann, O. Zweigle, P. Levi. RoboEarth Action Recipe Execution. 2011

[12] M. Tenorth. KnowRob Wiki. 2012. URL: http://ias.in.tum.de/kb/wiki/index.php/Exchanging_information_via_RoboEarth

[13] https://kforge.ros.org/turtlebot/trac/chrome/site/turtlebot320.png

[14] http://ftp.isr.ist.utl.pt/pub/roswiki/attachments/Robots(2f)Husky/husky-a200-unmanned-ground-vehicle-render.jpg

[15] L. Mösenlechner. The Cram Plan Language - Plan-based Control of Autonomous Robots

[16] F. Johannßen. Knowledge sharing for robots. 2013