

Cooperative Human Robot Interaction with the Nao Humanoid: Technical Description Paper for the “Radical Dudes”

Peter Ford Dominey¹, Tijn van der Zant¹, Stephane Lallee¹, Anne-lise Jouen¹, Xavier Hinaut¹, Alfredo Weitzenfeld², Herke van Hoof³, Jorge Dávila Chacón³

¹ INSERM U846 Stem Cell and Brain Research Institut, Robot Cognition Laboratory, 18 ave Doyen Lepine, 69675 Bron, France

² Computer Engineering, Instituto Tecnológico Autónomo de México, Mexico City, Mexico & Information Technology and Engineering Depts, University of South Florida - Polytechnic, 3433 Winter Lake Road, Lakeland, FL 33803

³Artificial Intelligence, faculty of Mathematics and Natural Sciences, University of Groningen, Nijenborgh 9, 9747AG, Groningen, The Netherlands
contact: robotijn@gmail.com

Abstract. Humanoid robots will increasingly interact with humans in daily life, especially in domestic situations. This document describes the physical and software configuration of our humanoid platform which is being developed in this context of human – robot cooperation. The research strategy is to use high quality commercially available robot platforms, and software for sensory-motor control, vision, and spoken language processing, in order to provide a high performance baseline system. From this baseline we then implement human-robot cooperation capabilities inspired by contemporary results in human cognitive development research. Because much of the technology that we use is off-the-shelf it is robust, versatile and reusable.

Keywords: humanoid robot, vision, sensory-motor control, spoken language, shared plans, cooperation, adaptive behavior-based architectures, learning

1 Introduction

The Radical Dudes team represents our third generation of participation in the RoboCup@home league. We initially participated as the EK-Lyon in Bremen in 2006 and qualified for the finals, then as the Robot Cognition Laboratory in Atlanta 2007 where we also qualified for the finals. In both of those competitions we used the Sony Aibo ERS-7 platform, running the Urbi system, along with a 6DOF arm in 2007. In both cases our Open Challenge task involved the robot learning from the human via demonstration and spoken language. We continue to develop this aspect of human-robot cooperation. While the Aibo provided a robust sensory-motor

platform, it was lacking in the ability to manipulate and co-manipulate objects in the context of human-robot interaction. This motivates our current platform choice for the Aldebaran Nao humanoid, running Urbi and Python code. The research and platform development described here is part of our long term effort to apply principals of computational cognitive neuroscience to robot perception (Dominey & Boucher 2005) and human-robot cooperation (Dominey et al. 2005; 2007a,b; 2008, Weitzenfeld & Dominey 2006, Weitzenfeld et al. 2008).

2 Hardware Description

The Aldebaran (<http://www.aldebaran-robotics.com/eng/index.php>) Nao is a 25 DOF humanoid robot. It is a medium size (57cm) entertainment robot that includes an onboard computer and networking capabilities at its core. Its open, programmable and evolving platform can handle multiple applications and is currently the most evolved humanoid robot available on the market. The onboard processor can run the URBI operating system and can be accessed via telnet connection over the internet via WIFI.



Fig. 1. The Aldebaran Nao (<http://www.aldebaran-robotics.com/eng/index.php>). **Left:** In this image, a spoken language command processed by RAD and sent over the internet was issued to activate a pre-recorded Choregraphe script for grasping an object and then standing up in preparation to carry the object to the user. This will contribute to the fetch and carry task. **Right:** Webots (<http://www.cyberbotics.com/>) simulator of Nao. This is perfectly compatible with the physical Nao, and thus will be significantly used for testing.

More specifically, the Nao is equipped with the following: CPU: x86 AMD Geode with 500 MHz, Memory: 256 MB SDRAM and 1 Gb Flash memory, WiFi

(802.11g) and Ethernet, 2 x 640x480 cameras with up to 30 frames per second (one pointing at the feet, and one pointing forward), Inertial measurement unit (2 gyro meters and 3 accelerometers), 2 bumper sensors and 2 ultrasonic distance sensors

3 Software Description

The robot is running on a Linux platform and can be programmed using a proprietary SDK called NaoQi. It supplies binding for C, C++, Ruby, Python and Urbi. It is also compatible with the robot simulator Webots by Cyberbotics. We will extensively employ the Webots simulator. Both NaoQi and Webots are available for Linux, Mac OS X and Windows.

In the context of the software architecture, our objective is to exploit as much as possible existing high performance software tools, specialized for different specific aspects of robot perception and cognition. The value added component that we provide is the framework for human-robot cooperation which holds these components together.

3.1 Behavior Editing With Choregraphe

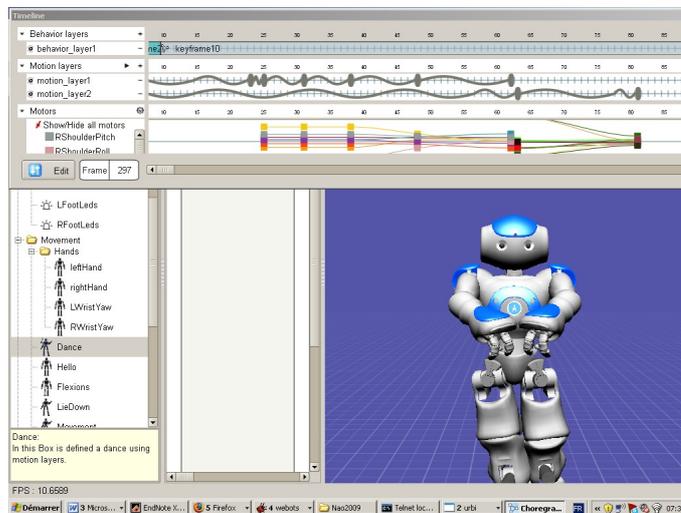


Fig. 2. Screenshot of the Choregraphe software GUI. Here we display the final posture of the Nao after testing and execution of the full body grasp behavior, corresponding to the actual robot configuration in Figure 1. Timeline in upper panel allows editing control of temporal sequences and transitions between predefined postures to create temporal behaviors.

Choregraphe is an intuitive GUI based tool that allows the user to rapidly create motion sequences, and to compose these together into complex, conditional

behaviors. In particular, in a passive mode the user can physically manipulate the Nao into a desired configuration, selectively save the concerned joints, and proceed, and then link these postures together in a temporal sequence. These posture sequences can then be loaded onto the Nao via wifi, and they can then be called from the controller software.

3.2 Conditional Behavior programming with Urbi

The Urbi system provides a concurrent and parallel programming language in which behaviors can be created and associated with a state machine, such that conditional execution of these behaviors can be realized (<http://www.urbiforge.com/>). The same code of the “follow a human” behavior that was used on the Aibo for RobotCup@home 2006 and 2007 (with full points both years in the Phase 1 component), will be used on the Nao platform. The modularity of Urbi function will ensure that this code will run minor modifications on the Nao. This code is uploaded via wifi to the Nao, and then it can be executed via telnet signals to the Nao as described below.

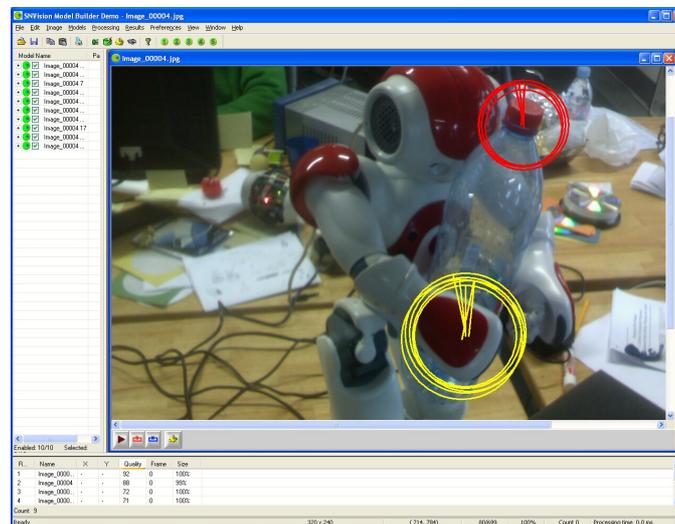


Fig. 3. Screenshot of the Spikenet Model Builder in recognition mode. Models have been constructed for recognition of the bottle top, and the hand of the robot. Here the recognition results are displayed as the superposition of oriented circles corresponding to the model recognition results on a frame of the camera image.

3.3 Vision Systems

Vision is one of the most important capabilities in human-robot interaction. A strong vision capability will allow the robot to see and orient towards objects, to walk to them, grasp them and bring them to the user. It will allow the user to introduce new objects, new faces, new scenes to the robot. We have successfully used the SNV Spikenet Vision system (<http://www.spikenet-technology.com/>) for action recognition in human-robot cooperative tasks.

We will use several slow, but very good available vision systems that work on an off-board computer, which will train a real-time vision system that runs on the Nao robot. We can this process 'learning to learn' (van der Zant et al. 2006), where we use machine intelligence to train other machine learning algorithms, because we require different attributes. In this we use slow, but good, systems which require only a few training samples, to train a fast algorithm which requires many training samples. Using this approach the robot can just wander around, and train itself.

For the navigation of the robot a robust learning approach is applied. Using a combination of standard SLAM algorithms and machine learning on salient visual cues, the robot is able to train itself. Using the same principles, the traditional SLAM method trains the vision-based machine learning algorithms.

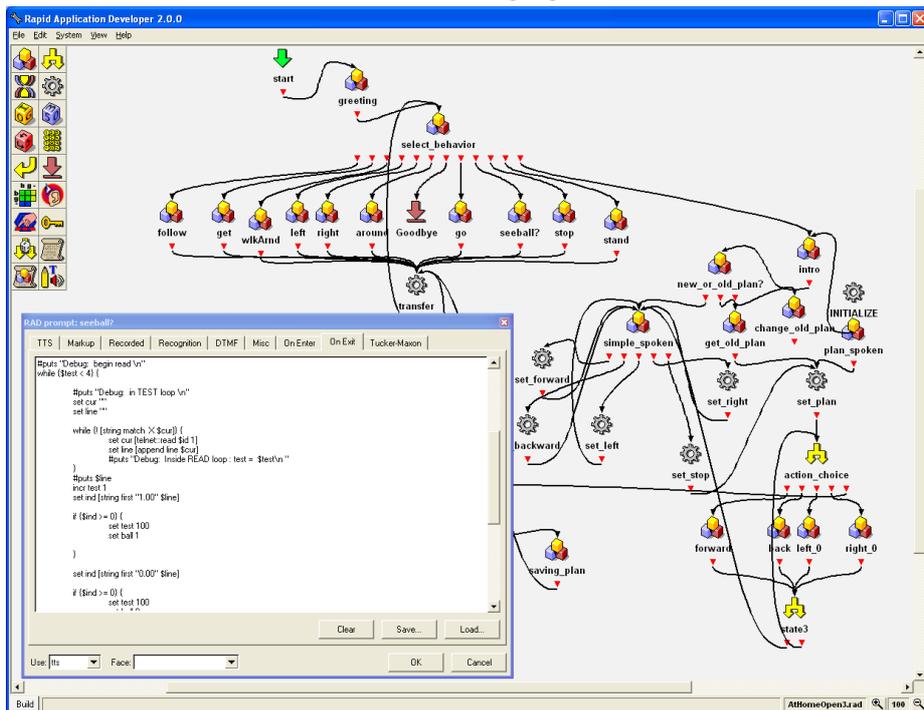


Fig. 4. Screenshot of the RAD program used in the 2006 open challenge. Nodes indicate states, with transitions conditional on spoken language recognition results and logical evaluation. Inset represents tcl code executed in one of these conditional state transitions.

3.4 RAD Spoken Language Processing

At the core of the system, bringing these different software components together is the CSLU RAD Toolkit (<http://cslu.cse.ogi.edu/toolkit/index.html>). The toolkit provides spoken language recognition and synthesis. It is presented in a state based configuration in which the user can drag and drop state modules from a panel. These state module provide functions including conditional transition to new states based on the words and sentences recognized, and thus conditional execution of code based on current state, and spoken input from the user, as illustrated in Figure 4. In this example, code is executed which via a telnet connection to the robot, uses the Urbi interface to interrogate the robot vision system to determine if the red ball is visible.

3.5 Integrated System

The Nao and the above described software components are configured together as illustrated in Figure 5. On the Nao the NaoQi environment allows the use of behaviors that have been created and uploaded to the robot via Choregraphe. Likewise, Urbi scripts and programs can be launched on the Nao. The coordinated use of Choregraphe and Urbi behaviors allows for a very rich behavioral repertoire. Rad provides the backbone that connects the other components to implement this behavioral repertoire. That is, Rad can connect to the Urbi server on the Nao in order to initiate Urbi behaviors, read from the joints and all of the sensors as well as the cameras. This allows conditional control of the robot to be handled by Rad. At the same time, based on the behavioral context, Rad can also trigger the execution of behaviors previously created with Choregraphe on the Nao via a separate telnet connection.

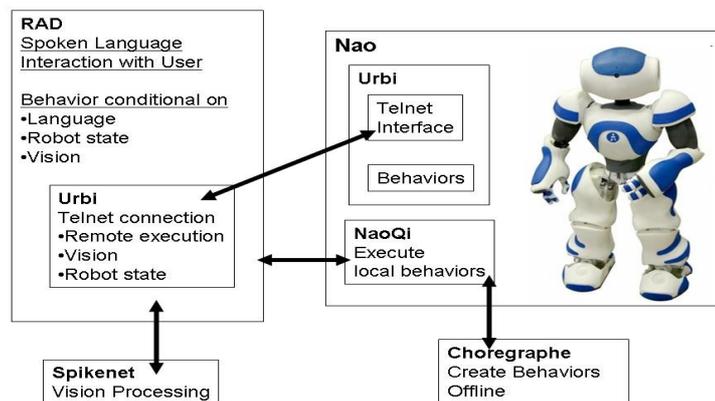


Fig. 5. Integrated system architecture overview. Running local on the Nao is the NaoQi system and the Urbi server. Note that the Webots (<http://www.cyberbotics.com/>) simulator of the Nao, can and will be used for extensive preliminary testing, combined with physical execution on the Nao platform.

4 Innovative Technology and Scientific Contribution

The Nao is a state of the art humanoid platform. The Choregraphe system allows real-time behavior creation through physical and GUI interfaces. Urbi then allows these behaviors to be executed, along with others, based on sensory motor contingencies. It allows the creation of state machines for specification of robust sensory motor behaviors. The RAD system provides a spoken language processing layer on top of this sensorimotor system. We have used a related architecture for human robot cooperation (Dominey et al. 2008, Weitzenfeld et al. 2008).

4.1 Research Focus

Our research is focused on human – robot cooperation. Recent studies of human and primate cooperative behavior (Warneken et al. 2005, 2006) have revealed that helping requires that the agent have knowledge of the goal of the other agent, and motivation to act with and / or on behalf of that agent. Going farther, cooperation requires that the agent not only represent the goal of the other agent, but that she represent their shared goal, and that from this shared goal can derive a shared plan in which each agent knows both what she and her partner will do (Tomasello et al. 2005). Based on these studies we have begun to develop robotic implementations of these principals (Dominey 2005, Dominey & Warneken 2008). Vision allows the robot to recognize action, and then form a representation of a shared plan in which the robot and the user each take their respective roles. Indeed given such a plan, the robot and human can exchange roles, and / or help each other.

Another focus is on the use of different algorithms to train each other. Often algorithms are complementary, for example, trading quality for speed. Although the Nao possesses a decent processor, it is not comparable to present day quad-core systems. Therefor we will use machine vision that runs on quad-core computers, to train the real-time tracker of the Nao, which uses haar-like features and blob-detection mechanisms optimized by clustering procedures. When the tracker is trained, it operates at 30 frames per second, without having to run Sift, Numeta, SpikeNet or other vision algorithms on the robot. Since the training of the real-time system takes a while, the other vision algorithms are used as a multiple classifier system to train the tracker. This will be the first time that we will use such a system on a robot.

4.2 Re-Usability by Other Groups and Real World Applications

Because of the off-the-shelf nature of the major components of our system, the system itself and the usage of the individual components is well suited for reuse by other teams. Indeed, we are demonstrating a form of reuse already by applying much of what was done with the Aibo directly to the Nao. The platform is also well suited for the preliminary development of real world applications. In particular we focus on

the concept of Ambient Assisted Living (<http://www.aal-europe.eu/aal-2009-2>) in which such robots will provide a social interface to aged people in their place of residence. The capabilities developed in the context of Robocup@Home have direct application in this new context.

Acknowledgements:

This work is supported in part by the FP7 ICT Projects CHRIS and Organic, the French ANRs Amorces, and Comprendre.

References

1. Dominey PF (2005) Toward a construction-based account of shared intentions in social cognition. Comment on Tomasello et al. 2005, *Beh Brain Sci.* 28:5, p. 696.
2. Dominey PF, Alvarez M, Gao B, Jeambrun M, Weitzenfeld A, Medrano A (2005) Robot Command, Interrogation and Teaching via Social Interaction, *Proc. IEEE Conf. On Humanoid Robotics 2005*.
3. Dominey PF, Boucher (2005) Learning To Talk About Events From Narrated Video in the Construction Grammar Framework, *Artificial Intelligence*, 167 (2005) 31–61
4. Dominey PF, Mallet A, Yoshida E (2007) Progress in Programming the HRP-2 Humanoid Using spoken Language, Proceedings of ICRA 2007, Rome.
5. Dominey PF, Mallet A, Yoshida E (2007) Real-Time Cooperative Behavior Acquisition by a Humanoid Apprentice, Proceedings of IEEE/RAS 2007 International Conference on Humanoid Robotics, Pittsburg Pennsylvania.
6. Dominey PF, Metta G, Nori F, Natale L (2008) Anticipation and Initiative in Human-Humanoid Interaction *Proc. IEEE Conf. On Humanoid Robotics 2008*.
7. Dominey PF, Warneken (2008) The Basis of Shared Intentions in Human and Robot Cognition, In Press, *New Ideas in Psychology*.
8. Tomasello M, Carpenter M, Call J, Behne T, Moll HY (2005) Understanding and sharing intentions: The origins of cultural cognition, *Beh. Brain Sc.*: 28; 675-735.
9. van der Zant, T., Schomaker, L., Wiering, M., and Brink, A. (2006). Cognitive developmental pattern recognition: Learning to learn. volume 2, pages 1208-1213.
10. Warneken F, Chen F, Tomasello M (2006) Cooperative Activities in Young Children and Chimpanzees, *Child Development*, 77(3) 640-663.
11. Warneken F, Tomasello M (2006) Altruistic helping in human infants and young chimpanzees, *Science*, 311, 1301-1303
12. Weitzenfeld A and Dominey PF, Cognitive Robotics: Command, Interrogation and Teaching in Robot Coaching, RoboCup Symposium 2006, June 19-20, Bremen, Germany.
13. Weitzenfeld, A, Ramos, C, Dominey, P, 2008, Coaching Robots to Play Soccer via Spoken-Language, RoboCup Symposium 2008, July 14-20, Suzhou, China