# Intuitive Human-Machine-Interaction and Implementation on a Household Robot Companion[1,2]

Christopher Parlitz, Winfried Baum,
Ulrich Reiser, Martin Hägele

Fraunhofer IPA, Nobelstr. 12, D-70569 Stuttgart, Germany
parlitz@ipa.fraunhofer.de

**Abstract.** The increasing capabilities of experimental household robot platforms require more and more sophisticated methods of interaction. While there are many developments in all directions of Human-Machine-Interaction, the integration and combination of several modalities into one robot system require some effort. To ease the development of applications supporting several types of interaction, Fraunhofer IPA has developed a framework named "Go". Within this framework we have integrated different kinds of interaction methods into one robot platform "Care-O-bot 3", a mobile service robot for accomplishing daily tasks. This framework and its interaction methods are presented here.

## 1 Introduction

Service robots or robot companions that interact with humans require a special form of Human-Machine-Interaction (HMI). This form of interaction, commonly known as Human-Robot-Interaction (HRI) is manifold. It can contain active interaction by the robot, recognizing what a person is doing and possibly why he is doing it, but also simple passive interaction like waiting for a button to be pressed. Thus HRI incorporates methods from traditional Human-Computer-Interaction, but also, due to the embodied nature of the machine, observations from human-human interaction. Based on a typical household scenario, we have developed a framework for integrating several human-robot modalities into one robot system. The robot system is our future robotic household companion Care-O-bot 3. In the second chapter, we will present our scenario in which we choose the required human-robot interfaces. The next chapter will explain our approach how we integrate all components including the HRI of the robot into one system. The fourth chapter will explain details of the implementation of HRI components derived from the scenario. The final chapter shows implementation and conclusions.

---

[1] The work described in this paper was partially conducted within the EU Integrated Project COGNIRON ('The Cognitive Robot Companion' - www.cogniron.org) and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

[2] This work was also partly funded as part of the research project DESIRE by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01A.

## 2 Scenario

We envisage our robot being useful in typical household scenarios. For more detailed analysis of, amongst others, the HRI components needed we have chosen a household application area. The gist of the experiment is to demonstrate to the robot how to lay a table, i. e. a human picks up objects from a storage area and places them in a typical place setting on a table. The robot observes the human during his movements and acquires the information necessary for repeating this task. After the initial learning phase the robot is asked to repeat the task several times for a complete table of e. g. six place settings. By analysing this scenario we have chosen the following HRI components for implementation and integration into our robot system: Speech Input / Output, Gesture Recognition and Remote Control. Additional components are the "traditional" functionalities of mobile robots like navigation, manipulation, object recognition and the like. While a remote control is not necessarily needed for the scenario above, our vision of a robot integrated into a household environment also e. g. incorporates control of the house itself. Additional visual feedback or supervision of the robot when it is not in close proximity requires some form of remote control.

We chose reliable, working implementations of the HRI components listed, adapted to our special needs of service robotics. Our focus is on the development of an integration framework for the components allowing simple integration and development of scenarios beyond the one described above.

## 3 Integration Framework "Go"

A seamless integration of robot components into one system is an ongoing topic in many robotic developments. This is due to near countless hard- and software components (Sensors, Actuators, diverse OSes, etc.) and requirements (bandwidth, real-time, etc.) that need to be integrated into one system. Several solutions already exist (e. g. ORCA [1], GenoM [2], MCA2 [3]), but are usually only suitable for some areas of applications or require a high effort of installation and maintenance. During the development of several robot systems the following key issues for a framework are indispensable:

- Modularity: A loose coupling of components is an advantage if changing setups, typical for service robot developments occur.
- Platform independence: Diverse hardware components need to be able to communicate.
- Inter-process communication: Different kinds of components and their data require diverse ways of communication, depending e. g. on bandwidth and response times.
- Resource management: Processes which use common sensors or actuators need to be coordinated, so that no conflict occurs.
- Configuration of the system: A flexible configuration of the control should separated and take place before the actual execution.

- Robustness: The framework should have a supervising functionality over the individual components.
- Tools for testing and debugging: Primarily during development, but also during run-time, tools for monitoring the processes are necessary.
- Performance analysis: If timing problems in the system's control occur, tools for performance analysis are necessary.

Our framework "Go" covers these requirements. Go is a high-level control framework implemented in Python [4], integrating and controlling lower component architectures. It was implemented as a module and a Go application is written as a Python programme using this module. This means, that all of Python's functionalities are available and therefore all advantages of a dynamic object-oriented script programming language (e. g. step-wise execution, object-orientated programming, consoles).

The extension of Python by Go allows simple and intuitive definition of hierarchical activities of individual system components. An activity is the smallest entity of a component. The activities can be executed synchronous or asynchronous to each other. Cyclic or one-shot executions are possible. Figure 1 shows some possible forms of activities.

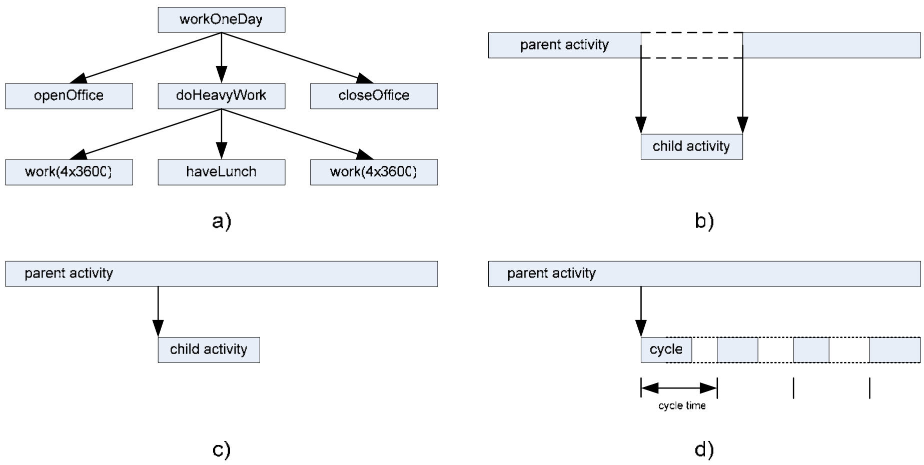


**Fig. 1.** Typical forms of activities: (a) hierarchical, (b) synchronous, (c) asynchronous, (d) cyclic

Therefore individual functions of the components, e. g. speech output can be addressed as an action, combined with other actions and joined together as one overall robot control operation.

A synchronization method guarantees that activities do not drift apart. The synchronisation is achieved by different kinds of wait commands in the Go module. These wait commands allow some activities to be suspend until other activities taking place in parallel have ended.

For process control across hardware components "GoCo" was developed, another Python module for activity communication across TCP/IP sockets. By using GoCo, activities on different computers can be treated as if they were implemented on one machine during control development. Processes that require an exchange of a lot of data, like images or movies in GUIs can be combined with the help of "GoSmaP". This is another module implemented with the help of smart pointers, which use shared memory to reduce copying time if possible. They also make sure that memory is released, if the data is no longer required by any activity. An error messaging system based on exception handling which extends through all application levels can be used for debugging and error handling. This debugging information can also be supervised remotely.

One disadvantage of Go is the lack of real-time functionality of script programming languages. While this is not required for most HRI components, it is as yet necessary for some applications like two-handed manipulation. If two components need real-time communication, this is done at a lower level. Go then causes the initialization of this communication at a higher level.

## 4 Details on HRI components

Each component below has been implemented as a module and offers one or more Go activities, which are integrated into the overall robot control. The creation of modules from existing components for usage in Go is either done manually, or can be done by an automated interface generator like Swig [5]. Figure 2 shows how the components are joined within the Go framework.

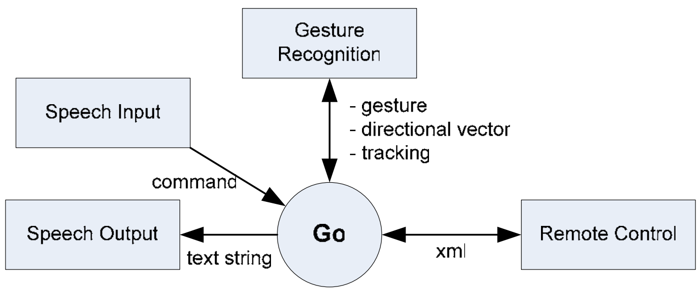


**Fig. 2.** Go and its HRI components with data flow

### Speech I/O

Although speech is only a small part [6] of human-human interaction, reliable speech I/O is a feature of service robots that seems to be expected by most people. We were able to deduce from several installed service robots (e. g. Communication Museum

Berlin). We use speech I/O for simple one word commands to the robot in different phases of the scenario and more elaborate status output by the robot. Speech I/O has several advantages. No special training is required compared to e. g. a keyboard and screen. The user has both his hands free and does not need to pay visual attention to the robot. For input we have selected a commercial application [7] based on the following conditions: Speaker independent, no training required, sufficient recognition rate and highly configurable. The key aspect for achieving a reliable speech input is the method of perception. While microphone and microphone arrays mounted on the robot are sufficient for lab and home environment, public shows make the use of speech very difficult. This problem is commonly called the "cocktail party effect" [8] and although current results are promising [9, 10] a common Bluetooth headset delivers a more reliable performance.

For speech output the most widely used system is the concatenative speech synthesis, where recorded natural utterances are combined to form new naturally sounding speech output. By using the speech synthesis program Festival [11] and the FestVox module [12], a voice based on unit selection [13] was created. While most concatenative systems use phonemes or diphones with fixed length the unit selection has variable length of the individual speech units reaching from a single phoneme to complete sentences. The speech corpus of Care-O-bot 3 created from sound files spoken by a professional narrator covers current and future scenarios of the robot. Additionally, all 48 phonemes of the German language are included, so that although we only have a limited domain of 379 different words, unknown words can be synthesized as well. If sentences are made up only from words known in the dictionary speech quality is very good. Occasional minimal variations of speech rate and prosodic problems in interrogative sentences occur. These problems are due to the relatively small speech corpus and the never quite perfect concatenative syntheses. The quality of the system declines in parts clearly, if unknown words are synthesized. This was expected and can only be overcome by a much larger phoneme inventory. The synthesis of a "normal" statement of approx. 10 words takes approx. 0.2 s on current computer systems. Therefore the speech output is fast enough for natural dialogs.

**Gesture Recognition**

To detect the human intention during the teaching phase, in which objects are placed, gesture recognition is necessary. En route to a higher system of understanding and detection of user intent, we have developed a new method [14] for the fast detection of body-parts, based on a modern range imaging sensor [15] and scale-space theory [16]. This approach has advantages over other published methods. While most methods are robust for fixed distances, certain environmental conditions or based on color (e. g. skin color [17]) or grey-level images for detection they have some simple drawbacks. If e. g. the hand is covered by a glove, there is no skin color or a poster showing a human might be falsely detected.

The idea behind our approach is to detect body-parts in 3D as extreme points over several scales as convex structure in the scale-space of range images. Since we currently only detect head and hands with this method, no complete kinematical setting of the human can be derived. Nevertheless with this information we were able to implement a demonstrator at the CeBIT 2006 computer fair and the AUTOMATICA 2006 in Munich as shown in figure 3. In this set-up a pneumatic robot torso mimicked the movements of a visitor by detecting the head and hand poses via the 3D sensor SwissRanger SR 3000.

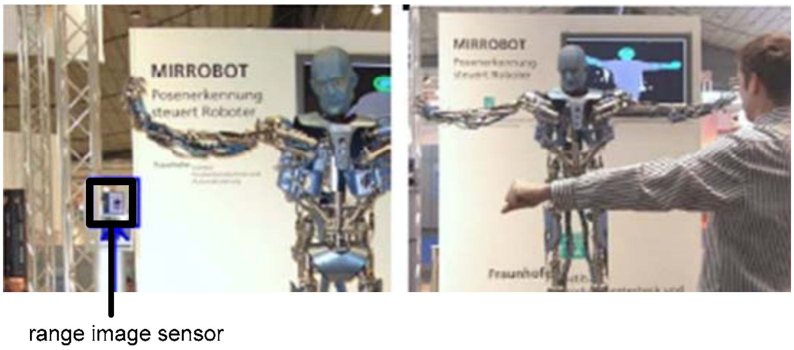


**Fig. 3.** The set-up at the fairs: A robot mimics the movements of a visitor

For a more accurate pose detection, the information gathered can be used to initialize a tracking algorithm to capture the complete human motion. Tracking algorithms can be based on color or again on using range information from a 3D sensor. The method we have currently implemented is an iterative-closest-point (ICP) algorithm presented in [18]. This combined method allows the interpretation of first simple gestures and the use of deictic information by the robot. We use it in the scenario for defining the position of the additional place settings to be laid out by the robot.

### Remote Control

While a graphical user interface (GUI) is not necessarily needed for the scenario described above, a concept for controlling a household robot companion was developed as part of the multimodal user interface. Typical elements of a GUI were combined, which allow a wide range of control based on content and context. The concept was developed within a usability engineering process following a user centred scenario based design by Rosson and Carroll [19]. Following the design guide, an analysis of the major factors was undertaken. The usage context analysis was divided into three important areas: the typical user, the task and the surroundings. For the analysis of the tasks, the user centred approach had to give way to a feasibility study, since potential users overestimated the possibilities of a household robot of today. Based on typical scenarios a user interface was realized in Java. The first prototype contained different kinds of interaction ideas, which were later used in user studies. In the following usability tests the interface was refined and some problems eliminated. A proof of concept was achieved by showing that the users were able to command the robot via the interface unknown to the users, to fulfil tasks which were chosen by us. The current state of the interface is shown in figure 4. The GUI is implemented on a tablet computer which was chosen based on criteria like size, weight and resolution.

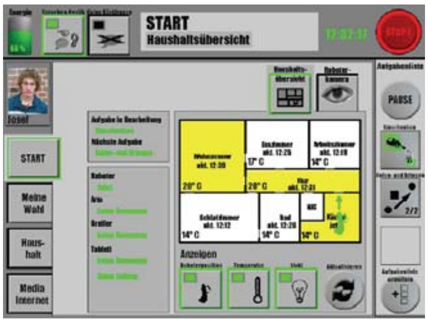


**Fig. 4.** Design of a GUI for a robot household companion

The actual current display status of the GUI is configurable via XML and is controlled by Go. In this case Go and GoCo only see that the XML strings are sent between the control on the robot and the tablet computer.

## 5 Implementation and Conclusion

Every interface has been individually tested on appropriate hardware and was designed to allow implementation on Care-O-bot 3. It is created for helping people in a household in an everyday environment. The preliminary design can be seen in figure 5a. The HRI interfaces described above are currently being implemented on a test setup (figure 5b) of Care-O-bot 3. Since the HRI components have not been fully integrated into the system, no results on performance or efficiency can be yet given. First versions of Go have already been used on several service robots (Care-O-bot generations, museum robots, etc. [20]).

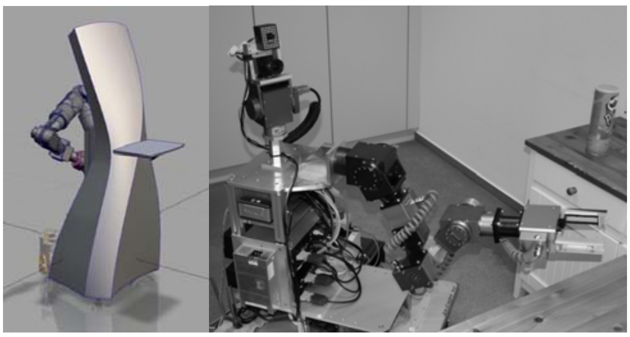

**Fig. 5.** Care-O-bot 3 in an early design phase (a) and test setup (b)

## References


1. http://orca-robotics.sourceforge.net/ (Retrieved Nov. 2006)
2. http://softs.laas.fr/openrobots/tools/genom.php (Retrieved Nov. 2006)
3. http://mca2.sourceforge.net/ (Retrieved Nov. 2006)
4. http://www.python.org/ (Retrieved Nov. 2006)
5. http://www.swig.org/ /Retrieved Nov. 2006)
6. Delhees, K.: Soziale Kommunikation. Psychologische Grundlage für das Miteinander in der modernen Gesellschaft. Westdeutscher Verlag (1994)
7. http://www.sympalog.de (Retrieved Nov. 2006)
8. Handel, S.: Listening: An Introduction to the Perception of Auditory Events. MIT Press (1989)
9. Rodeman, T. et. al.: Real-time Sound Localization With a Binaural Head-system Using a Biologically-inspired Cue-triple Mapping. In: Proceedings of the 2006 IROS, Beijing, China (2006)
10. Tamai, Y., Sasaki, Y., Kagami, S., Mizoguchi, H.: Three ring microphone array for 3D sound localization and separation for mobile robot audition. In: Intelligent Robots and Systems, 2005. (2005)
11. http://www.cstr.ed.ac.uk/projects/festival/ (Retrieved Nov. 2006)
12. http://festvox.org/ (Retrieved Nov. 2006)
13. Möbius, B.: Corpus-based speech synthesis: methods and challenges. In: Arbeitspapier des Instituts für Maschinelle Sprachverarbeitung (Univ. Stuttgart) AIMS. (2000)
14. Kubacki, J., Reiser, U.: Mirrobot: Fast Detection of Body-Parts in the Scale-Space of Range Images with an Application to Human Motion Copying, pp. 552-557, RO-MAN, University of Hertfordshire, Hatfield, United Kingdom (2006)
15. Oggier, T., Büttgen, B., Lustenberger, F.: SwissRanger SR3000 and first experiences based on miniaturised 3D-TOF cameras. In: Proc. Of the First Range Imaging Research Day, pp.97-108 (2005)
16. Lindeberg, T.: Scale–space: A framework for handling image structures at multiple scales. Technical Report CVAP-TN15, Royal Instute of Technology (1996)
17. Bretzner, L., Laptev, I., Lindeberg, T.: Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In: Face and Gesture, pp. 423-428, Washington, DC (2002)
18. Knoop, S., Vacek, S., Dillmann, R.: Modeling joint constraints for an articulated 3D human body model with artificial correspondences in ICP. In: International Conference on Humanoid Robots (Humanoids 2005); Tsukuba, Japan (2005)
19. Rosson, M., Carroll, J.: Usability Engineering. CA: Academic Press, San Diego (2002)
20. http://www.care-o-bot.de/ (Retrieved Nov. 2006)