The Rehabilitation Robots FRIEND-I & II: Daily Life Independency through Semi-Autonomous Task-Execution

Christian Martens*, Oliver Prenzel** & Axel Gräser**
*University of Bremen, Institute of Automation***
*Rheinmetall Defence Electronics**
Germany

1. Introduction

The rehabilitation robotic systems FRIEND[1]-I (Martens et al., 2001) and its successor FRIEND-II have been developing at the Institute of Automation (IAT), University of Bremen, Germany, since 1997 and 2003 respectively. The systems belong to the category "intelligent wheelchair mounted manipulators". They focus on users with high spinal cord injury, or with similar handicaps, who are unable to control the manipulator by means of a keyboard or joystick. The systems offer support during daily life activities and at professional life. The strategic objective of the FRIEND as well as the succeeding AMaRob[2] project, which focuses on the usage of FRIEND-II within the context of an intelligent environment, is to research into new methods to control the robotic system in such a way that their users become independent for at least 1.5 hours without support by nursing staff. Beside the aspect that this is one of the main requirements expressed by potential users, the fulfillment of this objective would have a strong impact on the commercialization of the rehabilitation robotic system itself.
This article gives an overview of the FRIEND project and the robotic systems there from evolved. It is divided into a practical part, which presents the systems from a user oriented perspective, and into a theoretical part, which satisfies the system-engineer's point of view. The user oriented part outlines the different development steps, functional improvements, hardware setups and lessons learned since 1997. Here, the facilities of the FRIEND-I system as well as a description of the innovations of the FRIEND-II system, currently under development, are described with specific emphasis of the AMaRob project. The reader becomes aware of the functionalities and services offered by the FRIEND rehabilitation robot and of the challenging technical complexity with which the development has to deal. The theoretical part is focused on the concept of semi-autonomous task-execution as a means of reasonable complexity reduction. Due to the consequent application of this concept a technically manageable robotic system emerges, which is able to execute tasks on a high level of abstraction in a reliable and robust manner. Within this context semi-autonomous task-

[1] FRIEND - Functional robot arm with user-friendly interface for disabled people
[2] AMaRob - Autonomous manipulator control for rehabilitation robots

Source: Rehabilitation Robotics, Book edited by Sashi S Kommu,
ISBN 978-3-902613-04-2, pp.648, August 2007, Itech Education and Publishing, Vienna, Austria

execution is used as a synonym for system initiated and controlled user involvement during task-execution. For the realization of this approach the control architecture MASSiVE[3] has been designed and implemented. It supports task-execution on the basis of a priori defined and formally verified task-knowledge. This task-knowledge contains all possible sequences of operations as well as the symbolic representation of objects required for the execution of a specific task. The seamless integration of user interactions into this task-knowledge, in combination with MASSiVE's user-adapted human-machine interface (HMI) layer, enables the system to deliberately interact with the user during run-time. It is shown how MASSiVE's application within the FRIEND-II system supports the future development of new services that increase autonomy of the users.

2. Evolution of Rehabilitation Robots

The development of rehabilitation robots started at the end of the 80ies in the last century. It was driven by the intention to support elderly and disabled people during daily life activities, making them more independent from care personnel or relatives. Additionally, the promising concept of an artificial assistant should be improved and technologically explored. This explains the chronological order of the appearance of different kinds of rehabilitation robots as they are presented in the following.

The first step in the evolution of rehabilitation robots was the development of fixed workstation systems that could execute pre-programmed tasks, like picking up paper from a printer or taking a book from a bookshelf. Quite popular systems of this category are DeVAR (Van der Loos, 1995; Mokhtari & Amni, 2001), ProVAR (Wagner et al., 1998; Van der Loos et al., 1999), RAID (Dallaway & Jackson , 1993; Eftring, 1994), MASTER-RAID II (Dallaway et al., 1995; Busnel et al., 1999; Mokhtari & Amni, 2001) or CAPDI (Casals et al., 1999). All these systems have in common that they consist of an industrial robot that is mounted at a workstation. Furthermore, they possess HMIs that are adapted to their users' special needs resulting from his or her disability.

Due to their structured and well known environment, fixed workstation systems could process complex task efficiently. Because this ability is limited to these predefined tasks, it turned out that these systems were too restrictive for a flexible use, e.g. in the domestic environment. This was the main motivation for the development of special purpose wheelchair mounted manipulators, like MANUS (Mokhtari & Amni, 2001), Wessex/Weston robot arm (Hillman et al., 1999) or RAPTOR (Mahoney, 2001). By means of different HMIs, e.g. joystick, keyboard or space-mouse, the user can control the gripper of the arm with respect to a Cartesian coordinate system or control each joint of the arm directly. In contradiction to industrial robots the main goal for the development of these manipulators was to create lightweight robot arms that satisfy special security requirements resulting from the direct human-machine interaction. This comes along with a loss of positioning accuracy, so that the pre-programming of complex action sequences, even in structured environments, was no longer possible. This is the reason why the control of these systems can only be performed on a low level of abstraction. For example, the user can command the direction for the arm movement or open and close the gripper. Even though this kind of control offers a great flexibility in use, it puts a high cognitive load on its users. Especially for complex tasks

[3] MASSiVE - Multilayer Control Architecture for Semi-autonomous Service Robots with Verified Task Execution

this is tiresome and comes along with a loss of concentration (Kawamura et al., 1995). Additionally, the kinds of HMIs which come along with these manipulators exclude tetraplegic (spinal cord injuries above vertebrae C5) or persons with a similar clinical picture.

A possible approach to encounter the above mentioned problems is to develop fully autonomous and mobile assistants, which are able to execute tasks on a high level of abstraction. Example systems of this category are MOVAR (Van der Loos, 1995), URMAD (Innocenti et al., 1994), MOVAID (Dallaway et al., 1995), WALKY (Bolmsjö & Neveryd, 1995) or Care-O-Bot (May & Schäffer, 1999). All these systems consist of a mobile platform with a manipulator mounted on it. In order to act in an unknown environment they are equipped with different kinds of sensors, like ultra-sonic or laser-beam distance sensor, cameras etc. All sensors as well as the platform and the manipulator itself are connected to a computer system that processes the user commands and controls all peripheral components.

At this point it has to be mentioned that, even though the above introduced idea of an autonomous robotic is as old as the history of robotics itself (Engelberger, 1989), its consequent realization has to be stated as unrealistic at the moment: Assuming that solutions for currently unsolved technical problems, like the real-time interpretation of camera images, are at hand, a fully autonomous system is cost-intensive and performs tasks with poor efficiency and reliability (Dario et al., 2004). This is because of its inherent high technical complexity. A manageable system with predictable behavior is required. With respect to the current state of science and technology, this requirement can be fulfilled only if the user's cognitive capabilities are taken into account, i.e. the robot executes the tasks semi-autonomously (Laschi et al., 2001; Martens et al., 2002; Colle et al., 2002).

Semi-autonomy takes the users' cognitive capabilities into account whenever a complex decision or environmental identification has to be made. Example systems of this category are the workstation mounted systems ISAC (Kawamura et al., 1995) and MUSIIC (Kazi, 1996; Kazi et al., 1997) as well as the wheelchair based systems KARES (Bien et al., 2001). The main principle is to offer simple but fully autonomous skills that can be activated by the user if necessary. Examples are the visually controlled grasping of objects (Lang et al., 2000), force-torque controlled drink serving (She et al., 2003b) or weight controlled pouring of a drink (She et al., 2003a). Because these skills are realized by sensor-based closed loop control processes, their execution becomes robust against dynamic environmental changes, even in unknown environments.

Even though offering autonomous executable skills reduces the amount of necessary user interactions during task-execution, a high cognitive load for the user still remains. He or she still has to remember the preconditions that have to be satisfied prior to the skill activation. For example, if a glass has to be grasped in a visually controlled manner, whereas the camera is mounted on top of the gripper (Lang et al., 2000), the user first has to move the gripper into the vicinity of the object to be grasped. Only if the underlying image processing system is able to extract significant features, the execution of the reactive grasping skill will be successful. Here, the approach of semi-autonomous task-execution, as it has been developed for the FRIEND-II robotic systems since 2003, will reduce the users' cognitive load. In the following, the rehabilitation robotic systems FRIEND-I & II are presented in general. Afterwards, the realization of the latter mentioned concept is described in detail.

3. Rehabilitation Robots FRIEND-I & II

This section gives an overview of the development steps of the FRIEND-I and FRIEND-II system as they have been undertaken since 1997. It is shown that the evolution of FRIEND-I closely correlates to the general evolution of rehabilitation robots as it has been described in the previous section. With the realization of an autonomously executed “serve-drink task” its full potential was tapped, so that the development of FRIEND-II started. FRIEND-II comes up of with a multitude of new hardware as well as software features that simplify the realization of flexible and robust task-execution. First, the new hardware setup is introduced in detail, before FRIEND-II’s role as a subsystem embedded in an intelligent environment within the context of AMaRob is described.

3.1 FRIEND-I

FRIEND-I consists of an electric wheelchair and a MANUS (Exact Dynamics, Netherlands) robot arm. The robot arm is controlled by a PC, which is mounted on the backside of the wheelchair. For user interaction an LC-display is used. In order to execute different tasks autonomously, FRIEND-I is equipped with a stereo pan-tilt-zoom camera system, mounted on the back of the wheelchair, and a “smart tray”. The tray is mounted at the front side of the wheelchair. It is characterized as “smart” due to its ability to locate object positions as well as to measure object weights, as described later in this section. A picture as well as a diagram of FRIEND-I is given in Fig. 1.

The first approach, to make the robot arm controllable for people with insufficient flexibility in their hands and arms for using a joystick or keyboard, was to equip the system with a speech control interface (Borgerding et al., 1999). By means of simple commands the robot can be operated with respect to different coordinate systems. The user can enter naturally spoken commands that are transformed into direct robot control actions. For example, if the user wants to pick up an object placed on the tray, a possible command sequence might be “Hand forward”, “Hand down”, “Stop”, “Gripper open”, “Hand forward”, “Stop”, “Gripper close”. Because the user observes the actions of the robot arm continuously, he or she might interrupt the actions in erroneous situations.

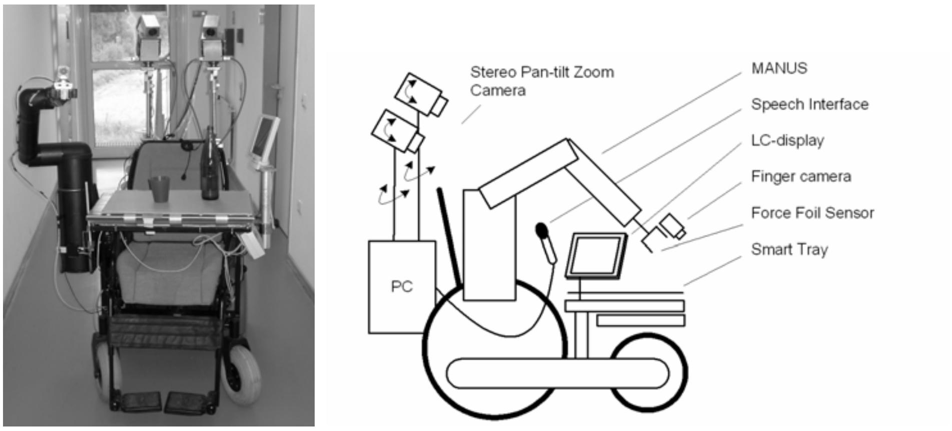


Fig. 1. Front view of FRIEND-I in summer 2003 (left) and schematic drawing (right).

By means of the speech interface it is possible even for a completely paralyzed user to command complicated tasks like grasping a bottle, pouring a drink in a glass and serve it to him- or herself. The drawback of this approach is that it requires high concentration over a long period of time. Therefore, in analogy to the robotic systems listed within the introduction, the succeeding approach in the FRIEND project was to pre-program often repeated tasks, like pouring and serving a drink or to put down objects on the tray. Even though it was convenient to activate the execution of these tasks by a single command, their application was too inflexible for the treatment of daily life activities.

A promising alternative was to increase the abstraction level of the commands by offering autonomously executable sensor-based skills[4]. The first implemented skill of this category was the camera controlled grasping of arbitrary objects. For its realization a finger camera was mounted on top of the robot gripper (Lang et al., 2000). If the user wants to grasp an object he or she moves the gripper into the neighborhood of the object to be grasped, until it can be seen by the finger camera. Starting from this position a visually controlled grasping action can be started with the command "grasp object". During the execution of the autonomous grasping action an underlying visual servoing algorithm continuously interprets image features that are assigned to natural or artificial markers on the objects.

The implemented grasping skill turned out to be robust during execution. Even under changing illumination conditions or with changes in the position of the target object the grasping tasks could be finished successfully without human intervention. It was possible to grasp an object placed in the workspace of the robot arm by means of three to five simple commands in comparison with 15-20 commands as necessary with direct voice control. Besides the reduction of necessary commands the whole execution time was reduced up to three times in average. This was the motivation to enhance the level of abstraction for task-execution again. The representative task of filling a glass with a drink and serve it to the user, short: the "serve-drink task" was chosen. The realization of this task unveils a number of challenging technical problems to be solved, which are also representative for further tasks. Therefore, the investigation and realization of the "serve-drink task" has the potential to develop a general method for robust high-level task-execution in rehabilitation robotic systems.

The underlying scenario of the task can be described as follows: A glass and an open bottle, filled with an unknown amount of drink, are arbitrarily placed on FRIEND-I's tray. After the user has entered the command "serve drink", the system has to fill the glass with the drink and move it to the mouth of the user. After the user finished drinking, the glass has to be put back on the tray. At the first sight the "serve drink task" seems to be trivial and restricted, but it deserves closer attention. First, the system has to locate the glass and bottle on the tray, grasp the bottle and move it close to the glass. Afterwards, the glass has to be filled and the bottle has to be placed back on the tray. Then the glass has to be grasped and moved close to the user's mouth. Finally, the glass has to be put back on the tray, ready for a succeeding pouring action. The execution of this scenario solely on the basis of image processing and visual servoing is very ambitious, especially if a robust behavior under all possible environmental conditions is requested. Additionally, unreliable behavior may cause dangerous situations for the user. To prevent such situations and to increase the system's reliability a "smart tray" was developed that is used in combination with the vision sensors (Volosyak et al., 2003).

The tray can be divided into two subsystems: A scale for the measurement of weight changes of objects placed on the tray and a touchpad for the detection of their positions. The

[4] These operations are characterized as skills, since they represent indivisible atomic functionalities of the system that execute autonomously sub-tasks by means of continuously processing sensor information. Examples are grasping an object or pouring a drink.

scale consists of an off-the-shelf digital scale with a measuring precision of ±1g that is connected to the main system PC. The position detection is realized by a touchpad sensor that was developed at the IAT. The touchpad consists of a 48x30 matrix, where each matrix element has binary output. Binary 1 denotes the presence of a weight greater then 3g per element, 0 indicates that there is no load on the corresponding matrix element. Hence, the result can be treated as a binary image and known image processing methods can be used for object localization. Fig. 2 (right) depicts the raw binary touchpad image resulting from a cup and a bottle that are placed on the tray. The outputs of the touchpad are processed by a microcontroller and send to the main PC. It is obvious that the positions of object segments can be easily determined on the basis of the binary touchpad image. But it is also evident that the amount of information from these images is insufficient for object identification. Here, additional sensor data is required that is fused with the touchpad information. By means of the information coming from the smart tray in combination with image processing results provided by the stereo-camera system, an autonomously executable "serve-drink task" was realized (Radchenko et al., 2004). The objective was to demonstrate that care personnel could place the involved objects, i.e. the glass and the bottle, on arbitrary positions on the tray and the system is able to execute the rest of the task autonomously. Even though this approach was independent of calibration, as it was for the pre-programmed tasks described before, the realization was fixed in the manner that the system could not be used for the processing of varying task scenarios. With the introduction of a flexible software-architecture this problem is solved within FRIEND-II, as described in the following.

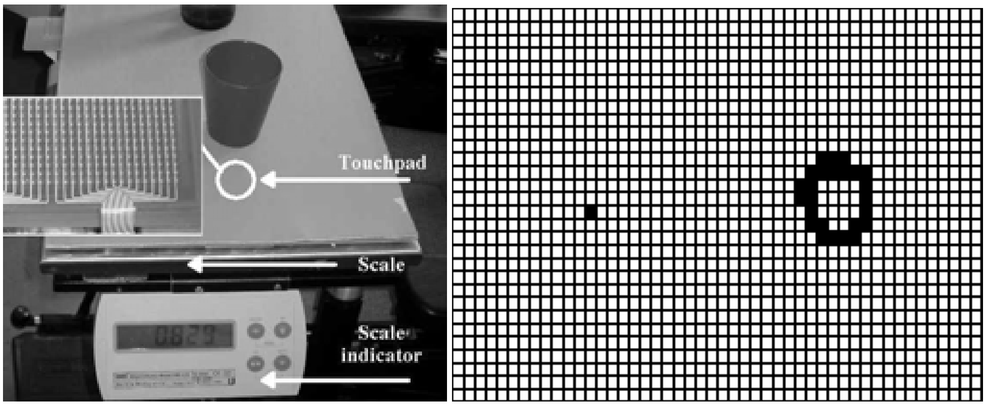


Fig. 2. Top view of the smart tray (left) and binary image of the sensor matrix (right).

3.2 FRIEND-II

The development of the FRIEND-II system started in 2003 with the objective to improve the FRIEND-I system and to benefit from the lessons learned so far. On the one hand, there were different shortcomings in FRIEND-I's hardware setup. On the other hand the flexibility of the control concept and the software-architecture in FRIEND-I was a rather basic approach that aimed to demonstrate the feasibility of rehabilitation robotic support scenarios. This section discusses the different extensions of FRIEND-II with respect to the hardware configuration, whereas the control concept MASSiVE is discussed in detail in the succeeding section.

Hardware Setup for Dexterous Manipulation in Clustered Environments

The MANUS robotic arm, which has been used in FRIEND-I, corresponds with its 6-joints kinematical structure to industrial robots. Such a structure turned out not to be suitable for dexterous manipulations. The ability to master manipulations with obstacle avoidance in clustered and mainly non-structured home environments is mandatory for the application in rehabilitation robotic scenarios.

Within FRIEND-II a dexterous 7 degrees-of-freedom robotic arm, which has been developed by Amtec Robotics (Berlin, Germany) under functional specifications by IAT, is used. It has a humanlike kinematical structure: The arm is composed of a series of turn- and pan-joints with perpendicular axes respectively. The arm is mounted on a linear axis which allows it to drive in a specific home position and reduce visibility if not in use (Fig. 3). At the wrist a multi-axis force/torque sensor, model Gamma, from ATI-Industrial Automation (NC, USA) is integrated. This compact, light and robust monolithic transducer uses silicon strain gauges, providing high noise immunity, to sense forces and torques from all three directions of the tool frame. To process the strain gauge information into CAN-Bus information a compact wrist mounted electronics unit has been developed. The robot arm is equipped with an Otto Bock SensorHand[5] as a gripper. The necessary mechanical as well as electrical adaptations were made in agreement with Otto Bock Health Care (Duderstadt, Germany). A gripper force and a slip control mode, which will be activated from the FRIEND-II system, are integrated in the SensorHand. As it has proven to be a benefit for the realization of robust operations, FRIEND-II is, like its predecessor, equipped with a smart tray for determination of object positions and weight changes. The camera Sony EVI-D70P was selected for the FRIEND-II image acquisition system. The selection was made on the basis of requirements for minimal lighting of the scene and particularly on the basis of a cost-effective connection of the camera with a pan-tilt head. The chosen pan-tilt zoom video camera system is mounted on the frame-rack behind the user.

With the help of the described improved hardware setup and the implementation of the new software control concept MASSiVE, the first successfully implemented scenario was an extended drink serving scenario, since this could serve as reference scenario for comparison with FRIEND-I. This scenario was presented at the Hannover-Fair 2005 and it turned out that during the whole week a robust execution, in the sense of the following explanation, took place. Independent of the initial configuration of bottle and glass (sometimes placed by fair visitors) the pouring action was executed as pre-determined. Beside the statement that a stable accuracy from different initial conditions is related to the improved manipulative capabilities of the new robotic arm, two other hardware components contributed also to the enhanced overall performance: The Otto-Bock gripper with its integrated intelligent force-control guaranteed a firm grip even on a slightly wet bottle. Furthermore, the force-torque-sensor enabled to implement a more flexible placing of objects even on non-tactile surfaces like an ordinary table. All in all the improved hardware setup of FRIEND-II is the basis for currently ongoing implementation of further scenarios. The well focused development of different representative and complete rehabilitation robotic scenarios on the basis of the experiences gathered with the FRIEND systems is the objective of the AMaRob project, which will be discussed in the following.

[5] http://www.ottobockus.com/products/upper_limb_prosthetics/myoelectric_hands_sensorhand.asp

AMaRob Project: Support Scenarios for Daily Life Autonomy

The project AMaRob is funded within the BMBF[6]-program "Leitinnovation Servicerobotik" (IAT, 2007; DLR, 2007). The overall objective is to demonstrate that 1.5 hours of complete autonomy from care personnel or nursing staff can be realized for tetraplegic people, solely supported by a rehabilitation robot. The suitability for daily use as well as efficiency from the economical viewpoint in one ADL (activities of daily living) and two working scenarios is to be examined. An important aspect with respect to the successful project process is the interdisciplinary cooperation. From the beginning on, therapists (Neurological Rehabilitation Center Friedehorst, Bremen, Germany), designers (i/i/d – Institute of Integrated Design, Bremen, Germany) and various experts for the different technological components (Meyra, Kalletal-Kalldorf; Otto-Bock, Duderstadt; Amtec Robotics, Berlin; IGEL GmbH, Bremen; all from Germany) are cooperation partners in the AMaRob project.

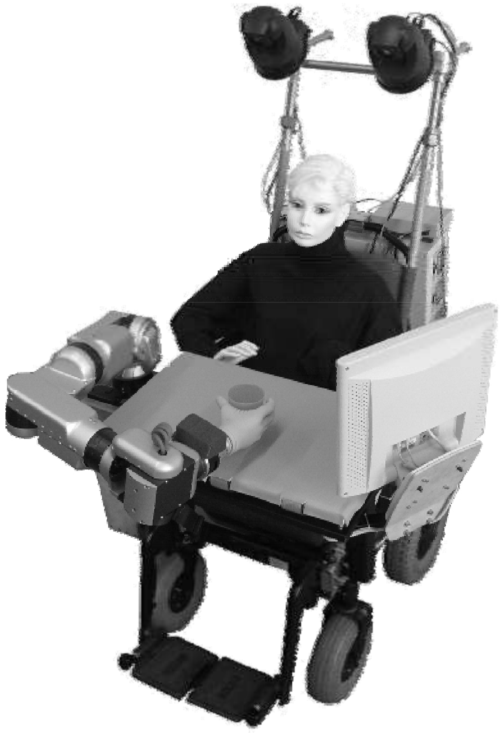


Fig. 3. Rehabilitation robotic system FRIEND-II.

The main development goal in the ADL scenario is the complete process of preparing and eating a meal. This starts with the retrieval of the desired meal from the storage, the proper heating procedure with respect to the contents of the meal, the eating and drinking procedure and finally the clearance of the dishes. In the first working scenario different electronic

[6] BMBF – German Federal Ministry of Education and Research ("Bundesministerium für Bildung und Forschung")

components are tested on the basis of manual inspection as well as functional tests. These tests are common tasks that are often accomplished by disabled people with remaining manual capabilities in the workshops at Friedehorst. The second working scenario deals with a service desk in a library, where books are rented or returned, and dues are paid, reservations are handled, etc.

During the AMaRob project, a redesign of FRIEND-II will take place. Development goals are an improved design for daily use, a smoother integration of sensors and actuators, including the new generation of Amtec Robotics manipulators. Another aspect that is valid for the development of all three scenarios is the investigation of methods to integrate distributed smart components, also known as ambient intelligence or ubiquitous computing (Korondi & Hashimoto, 2003). Its objective is to support the process of task-execution and to lower the technical complexity of the rehabilitation robotic system itself. Based on the experiences gathered with the smart tray within the FRIEND projects, it will be considered to place tactile skins on certain platforms that are in the center of manipulation, e.g. parts of the worktop in the kitchen or the desk in the workshop or library. Furthermore, RFID[7]-tags attached to objects will enable the decentralized storage of object relevant information, e.g. cooking instructions for a meal placed in a smart fridge that is equipped with RFID antenna.

Fig. 4 illustrates an exemplary setup of an "intelligent" kitchen environment. With such an intelligent environment, a distributed system evolves with hardware that is distributed physically and that is managed by different processors. In Fig. 4 on the right side, a prototypic installation of an intelligent kitchen environment is depicted. Instead of using a real fridge, the first setup includes a cupboard, representing the fridge. This cupboard is equipped with two tactile layers, which consist of the same tactile skins as that one used for the smart tray component of the FRIEND-II system. Thus, less technical complexity of the rehabilitation robot itself is necessary with respect to required sensors and algorithms to retrieve location information about the meals in the smart fridge. Under the upper cupboard layer an RFID antenna is installed which covers the space above the two layers. The meal to be prepared for the disabled user is arranged on a special meal tray, which fulfils the following conditions:

- It is graspable by the robotic gripper
- The material is suitable for the heating in a microwave oven, is food safe and can be easily cleaned after usage
- The lid is equipped with RFID tags. This lid is also manageable by the robot and can be removed before heating to avoid damage of the RFID tag in the microwave oven.

Once care personnel inserts a meal in this kind of smart fridge, meal specific information will be stored on the RFID tag. This includes a description of the meal, cooking instructions and minimum durability. During the process of meal preparation, the locally stored meal information directly drives the heating procedure in the microwave oven. This oven is also part of the intelligent environment, since it is directly controllable from the system without manipulative interaction and consequently also reduces the technical complexity of FRIEND-II.

Besides the ongoing improvements on the hardware level and the application and enhancement of an overall software control concept, other key developments will be undertaken within AMaRob. This includes fast motion planning for collision free, intelligent and

[7] RFID – radio frequency identification

smooth manipulation in clustered environments (Ojdanic et al., 2006) or the realization of improved machine vision concepts respectively.

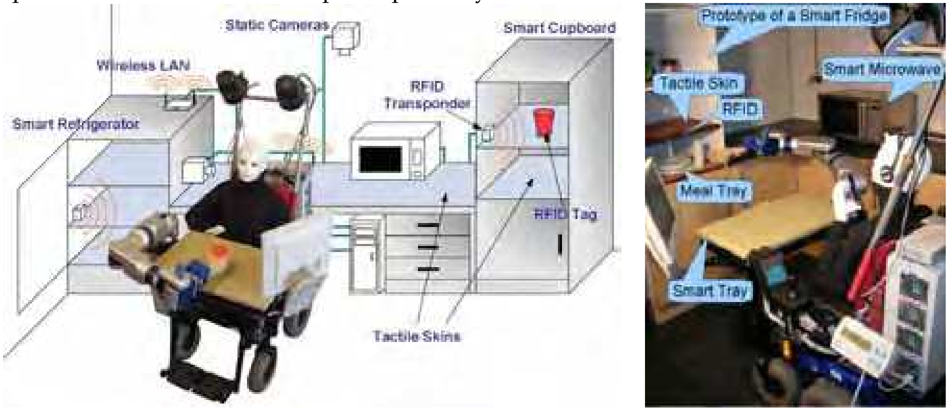


Fig. 4. FRIEND-II in intelligent environment, schematic version (left), first prototypic installation (right).

4. Semi-Autonomous Task-Execution

The preceding sections concentrated on the functional aspects of the FRIEND systems. This section is focused on the application of software-technical methods that help to cope with the inherent complexity of the robotic system and its environment. First, the design and implementation of an appropriate software-architecture is described. Afterwards, within the context of this architecture, task-knowledge data-structures, suitable for the inclusion of the user, as required for semi-autonomous task-execution, are introduced. It is shown how the consequent application of these concepts leads to a robotic system, which is able to execute tasks robustly on a high level of abstraction.

4.1 The Software-Architecture MASSiVE

From the software-technical point of view the realization of a service robot comes up with a multitude of challenging tasks to be solved within the context of an architecture: Distributed calculation, reactions to environmental changes and control of electro-mechanical devices under hard real-time requirements, processing of complex algorithms, adaptation to heterogeneous hardware interfaces, ergonomic human-machine interaction and autonomous planning of action sequences. The preceding enumeration doesn’t claim to be exhaustive.

It turned out that hybrid multi-layer architectures, like TCA (Coste-Maniere & Simmons, 2000) or 3T (Bonasso et al., 1998), are predominating in the field of fully autonomous systems, since they provide a combination of deliberative and reactive behavior. A deliberative component is necessary for the creation of a high-level plan, i.e. for having a system that is able to receive task requests and to define a mission goal. The inclusion of reactivity has proven to be a suitable mean to achieve robustness with respect to environmental disturbances, comparable to the reflex system of living organisms. Therefore, a typical hybrid multi-layer architecture consists of three layers:

- Deliberator: Plans operations on a high level of abstraction, i.e. with the help of a symbolic planning strategy.

- Reactive Layer: Has access to the actuators and sensors of the system and offers reactive operations in the form of closed control-loops (i.e. coupling of sensors and actuators).
- Sequencer: Is responsible for the activation and deactivation of operations in the reactive layer according to the plan generated by the deliberator. Therefore, it plays the role of a mediator between deliberator and reactive layer.

Besides these layers a world model is included that contains the current system's perspective on the world according to the task to be executed. Due to the hybrid approach a separation of world-model data into two categories is mandatory: The deliberator operates with symbolic object representations (e.g. *C* for the representation of a cup), while the reactive layer deals with the sensor data taken from these objects, so-called sub-symbolic information. Examples are the color, size, shape, location or weight of an object.

To fulfill the requirements in the field of rehabilitation robotics, the software control architecture MASSiVE has been developed at the IAT. This control architecture is derived from the generally successfully applied architectures of autonomous systems as introduced before. However, in the field of rehabilitation, the user can be included in to the process of task execution, which leads to modifications towards a semi-autonomous system. Furthermore, the analysis of typical rehabilitation robotic support scenarios revealed the fact that these scenarios are mostly composed of a finite set of reusable basic operations (Martens, 2003b).

Fig. 5 depicts the emerging scheme of the modified control architecture. With respect to the semi-autonomous task-execution, the deliberator component of traditional hybrid control architectures has been replaced by an HMI, whereas symbolic planning is performed within the sequencer. The evident role of the HMI is the translation of high abstraction level task requests from the user (e.g. "Fetch cup", "Pour in a drink", etc.) into commands that are forwarded to the sequencer for further processing. Additionally, the HMI provides a complete infrastructure for task related user interactions. This includes the direct control of actuators within context-based constraints given by the sequencer. Within MASSiVE, the role of the sequencer is to act as a global control unit that coordinates the cooperation between all levels of the architecture (Martens et al., 2002). It is designed as a discrete-event-controller that operates on the basis of predefined task-knowledge, so-called process-structures, and generates task related action sequences.

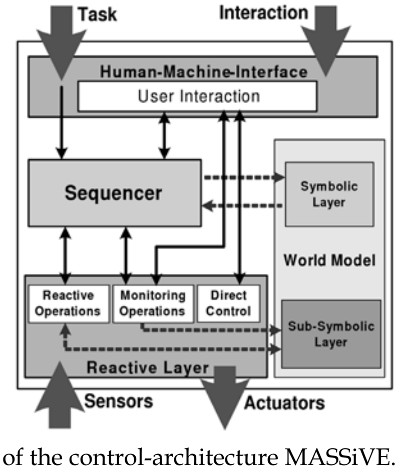


Fig. 5. Schematic overview of the control-architecture MASSiVE.

These process-structures are the basis of the principle to pre-structure the support scenarios. Despite their a-priori definition, they are still flexible enough to process a requested task under varying initial conditions, i.e. they can handle different initial situations and varying kinds of task participating objects. Due to the implicit restriction to task relevant information, this approach guarantees, in contradiction to classical AI-based task planning, a finite and, in the meaning of real-time suitability, reasonable size of the planning search space. Additionally, the formal verification of the task-knowledge with respect to reachability of situations or the correctness of its execution becomes possible. Even though this has no impact on the offered functionality of the robotic system, it is evident that the latter aspect plays a vital role for the commercialization of rehabilitation robots in general.

A planned action sequence consists of operations that are elementary from the sequencer's viewpoint. This motivates the term elementary executable operations (*EEOP*s). *EEOP*s include user interactions (e.g. identification of an object), direct control of actuators (e.g. movement of the camera system), monitoring operations (e.g. visual feature extraction of already identified object) and reactive operations (e.g. visually controlled object grasping). Thus, the basic operations already implemented for the FRIEND-I system are realized within this context as *EEOP*s. Due to the uniform software interface of *EEOP*s, the sequencer can generate action sequences independently of the kind of operation that has to be taken into account. After the generation of an *EEOP*-sequence is finished, the sequencer maps the *EEOP*s to skills that are executed asynchronously on distributed software-servers. These servers are part of the reactive layer or the HMI, respectively. In the reactive layer, a network of servers offers basic system skills that are grouped within the servers according to functional cohesion, as described in the following.

Infrastructure for Semi-Autonomous Task-Execution within Distributed Systems

The sequencer of the MASSiVE architecture (Fig. 6) consists of two modules that are designed as active objects: The *Task Planner* and the *Task Executer*. Active objects are a software design pattern to separate the execution of a method from its calling context with the help of threads, whereas the method's implementation is independent of any threading details (Gamma et al., 1995). Thus, the planner and executer are able to act independently. This enables the control of skill execution in the reactive layer as well as reactions in the planning layer like interruption of ongoing task execution or re-planning of required operation sequences.

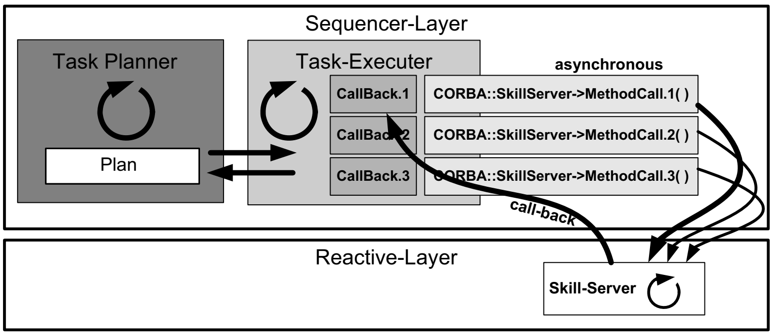


Fig. 6. Communication from sequencer layer.

To be able to execute several operations simultaneously, asynchronous calls of skill-methods are necessary. Furthermore, the operations may run on different processors, e.g. because of system-hardware that is distributed physically as it is the case for remote smart devices. The distribution of skill execution capabilities should be adaptable in a flexible manner, i.e. without changing the system structure or extensive re-implementations, to have the opportunity at hand to scale the computing power available for a single skill. All these demands are fulfilled entirely with the help of standardized and platform-independent communication infrastructures based on CORBA (Common Object Request Broker Architecture, (Gamma et al., 1995)). The following section describes the design and realization of the CORBA based reactive layer.

Reactive Layer

The name reactive layer resides from its purpose to provide reactive behavior. This means to directly couple sensorial input with the control of an actuator (i.e. to design a control loop) to establish autonomous behavior that is robust against dynamic environmental changes. As depicted in Fig. 5 the reactive layer is furthermore responsible for offering monitoring operations (based on input from the sensors) as well as direct control of the actuator (manipulative skills). The latter aspect is important for example when user interaction in the form of direct actuator-control becomes necessary. Due to this, several skill servers provide the necessary basic operations, i.e. skills, of the robotic system by accessing the sensors and actuators of the system or remote smart devices. This means, a skill layer has access to a hardware layer, whereas different hardware servers encapsulate basic hardware functionalities.

Skills have to operate on the already mentioned sub-symbolic environmental information. As shown in Fig. 5, the sequencer including the symbolic planning engine accesses the symbolic layer of the world model. Thus, the sequencer (on the basis of high-level process-structures and symbolic descriptions) is responsible for the correct abstract modeling of that segment of the environment that is relevant to the current task-execution. To administrate all sub-symbolic information in a structured manner, a sub-symbolic world model server is introduced within the reactive layer. Here, sub-symbolic information is stored with reference to symbolic information from the upper layer of the world model and consequently a connection between both layers of the world model (and therefore also between these both information layers) is established.

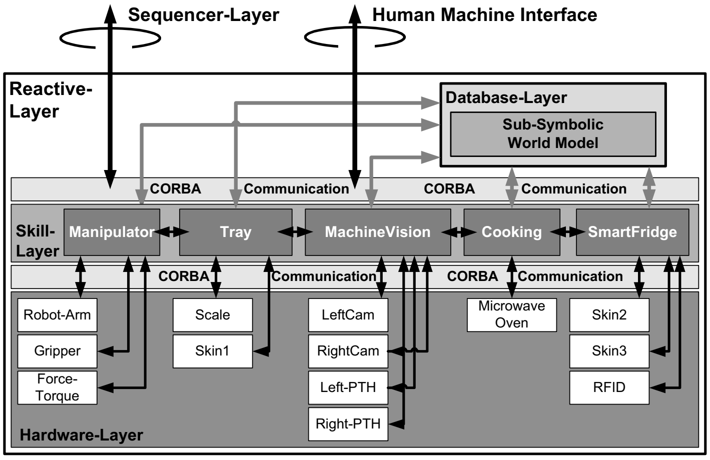


Fig. 7. Reactive layer for FRIEND-II with intelligent environment.

Skill Server

The criterion for separation into several skill servers is derived from the functional entities of the system. That means, one skill server offers all the system operations that have to be assigned basically to one certain entity. In case of scenarios that take place in an intelligent environment as introduced in Chapter 3.2, the necessary skill-servers are the *Manipulator-*, *Tray-*, *MachineVision-*, *Cooking-* and *SmartFridge-Skill-Server*, with their assigned hardware-servers as depicted in Fig. 7. Here, for instance, the Cooking-Skill-Server provides skills to access the hardware-server *MicrowaveOven* and thus to control this remote device or to extract data from it.

From the software-technical point of view, skills are methods of a skill server that are executed asynchronously. This means skill-methods are non-blocking and will return immediately after their activation. The problem of asynchronous execution is that no values or parameters can be returned. Therefore, sub-symbolic data that is generated during skill execution is stored within the world model. The information on the status of skill execution (e.g. successful execution) has to be transmitted via another communication way. For this issue call-back objects are introduced, which can be accessed by the skill caller and the skill method itself. Call-backs are also used for sending information from the skill caller to the skill while it is executing. This could be for example the command to stop the skill or to re-parameterize it. Fig. 6 already showed how the task executer within the sequencer maps call-back objects to asynchronously invoked methods and Fig. 8 demonstrates the communication mechanisms between sequencer- and skill-layer in more detail.

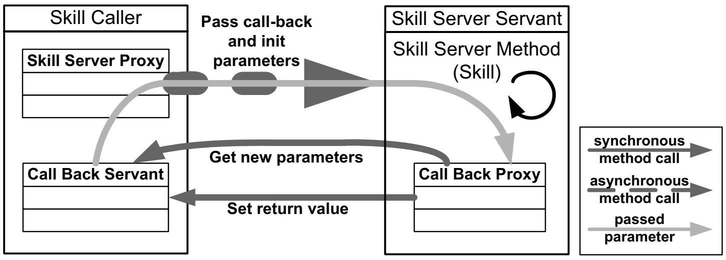


Fig. 8. CORBA-based asynchronous communication.

Hardware Server

For each hardware component, like the robot arm, a hardware server is implemented (see Fig. 7), which itself is connected to a certain skill server. The design principle is that only methods of this skill server can access the hardware directly. This decouples the upper layers of the software structure from the hardware. In case a hardware component is replaced, only the implementations of the hardware server methods have to be adapted, whereas the parameters of each method remain.

If for a hardware-component/-server a fast communication is required, i.e. the communication offset of CORBA cannot be accepted, a local instance of the hardware server implementation class can be used and the communication can be carried out via shared memory. Also the combination of client and servant in one process is possible. All three possibilities of using a hardware server are fully exchangeable and no re-design or new implementation is necessary.

Human-Machine-Interface

For the realization of the user involvement, which is initiated and controlled by the system, the HMI offers an interface that enables the sequencing layer to initiate user assistance via user interaction skills. From the user-centered viewpoint, the overall objective is to offer an intuitively controllable interaction skill, which hides all technical details, like the selection of necessary hardware resources or the control of the data flow. The connection between the HMI and MASSiVE is established by a software component that uses the interface of the sequencer to start and stop system tasks and offers a software-server to enable the sequencer initiating user interactions. From the sequencer perspective, the invocation of a user interaction is thus handled exactly like a skill activation within the reactive layer.

So far, the architectural concept designed for semi-autonomous task-execution has been discussed. In the following, process-structures as a substantial aspect for the control of user interactions and autonomous system operations within MASSiVE are presented.

4.2 Task-Knowledge Driven User-Interaction

According to the structure of the control architecture, process-structures come in two levels of abstraction: Abstract process-structures (PS_A) are associated with the abstract (symbolic) layer of task-knowledge representation and describe tasks in a user-oriented and non-technical way. This level of abstraction is suitable for task-oriented programming purposes (Martens, 2005). Elementary process-structures (PS_E) represent the part of the task-knowledge that is required within the environmental-related (sub-symbolic) level. The information on this level of abstraction subsumes necessary hardware resources, like sensors and actuators, as well as *EEOP*s. In the following, the different steps of task processing on the basis of abstract as well as elementary process-structures will be discussed. The activity diagram depicted in Fig. 9 gives a conceptual overview on the steps that are undertaken to process a task request in MASSiVE.

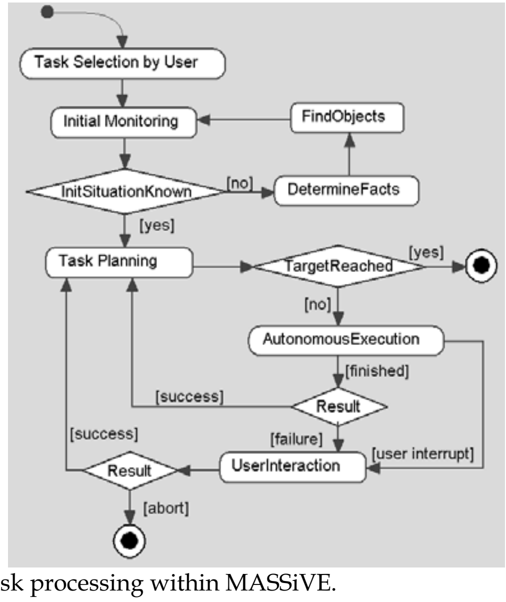


Fig. 9. Overview on task processing within MASSiVE.

Task Request

First, with respect to the task request from the HMI, the sequencer selects and loads an abstract process-structure. Abstract process-structures are derived from data-structures typically used in assembly planning, namely AND-OR-nets (Cao & Sanderson, 1998). Within the context of MASSiVE they have been enhanced with first-order predicate logic facts that assign, in a STRIPS[8]-like manner, pre- and post-conditions to the operations contained in the net. Aditionally, facts are used for the description of object states and relationships.

Fig. 10 depicts an iconic model of a possible PS_A that is assigned to the task request "Fetch cup". The iconic PS_A contains all object-constellations (*OC*s), i.e. objects in physical contact, which might appear during the execution of this task, as well as all possible operations performed by the system. Within this context *all* has to be interpreted as from an application programmer's viewpoint. The operations are called composed operators (*COP*), since they represent a composition of *EEOP*s and their decomposition takes place on the level of PS_E. They interconnect the different *OCs* of the PS_A, meaning that the system transforms the constellations of objects by executing a sub-task. Due to their derivation of AND-OR-nets, there are three different kinds of *COP*s: Assembly and disassembly operations as well as internal state transitions. The semantics of the first two types is self explaining, whereas the latter version is used to represent internal changes within a single *OC* without changing the components being in physical contact (e.g. moving a book on a table).

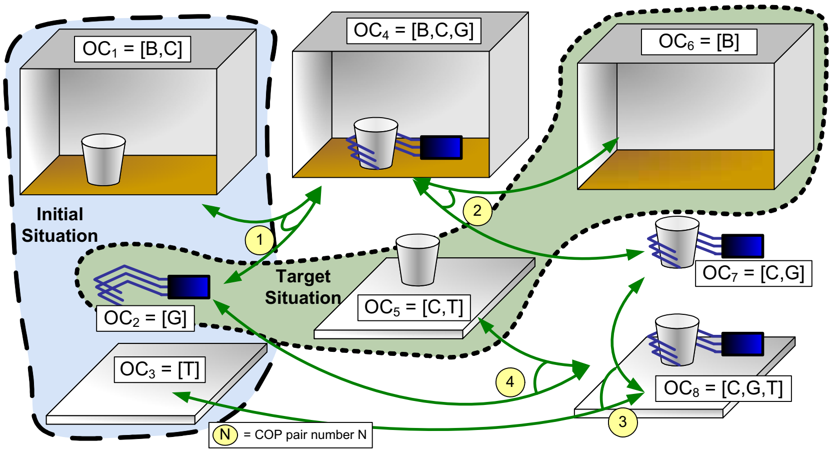


Fig. 10. Abstract process-structure (PS_A) for the task request "Fetch cup".

As listed in Table 1, the "Fetch cup" scenario solely consists of *COP*s of the first two categories. The numbers of each pair of operations refer to the numbers given in Fig. 10. The objects involved in the task scenario are a box *B* (e.g. an abstract representation of an already opened cupboard), a cup *C*, the gripper of the service robot *G* and a table *T*. Some exemplary *OC*s of Fig. 10 are: OC_1 = "the cup is located inside the box", OC_4 = "the grasped cup is located inside the box" or OC_5 = "the cup is located on top of the table". Beside the enumeration of contained objects, each *OC* possesses a set of first order predicate logics facts, which represent the object states as well as their relationship respectively.

[8] STRIPS: Stanford Research Institute Problem Solver (Fikes, 1971)

No.	**List of COPs used in Fig. 10:** (AOP - assembly operation, DOP - disassembly operation)
1	AOP: GraspObjectInContainer(G, C, B) DOP: DepartFromContainer(G, C, B)
2	AOP: PlaceObjectInside(G, C, B) DOP: GetObjectOutside(G, C, B)
3	AOP: PutDownObject(G, C, T, FreePlacePos) DOP: MoveObject(G, C, T, FreePlacePos)
4	AOP: GraspObject(G, C, T) DOP: Depart(G, C, T)

Table 1. Composed operators of the "Fetch Cup" scenario.

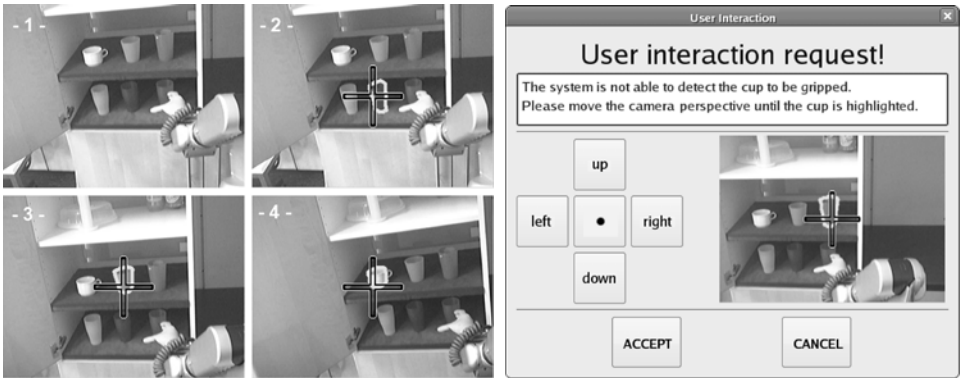
On the basis of *OC*s a situation S_k is described as follows: S_k is a set of *OC*s that contains all objects of the PS_A exactly once and is part of the situation-graph (Prenzel, 2005), which interconnects all valid situations via the operations defined within the PS_A. Assuming the correctness of the PS_A, which is verified automatically off-line, a target situation S_T can be reached from any situation *S* of the situation-graph via the application of a finite number of operations. This circumstance is the basis of the planning concept applied in MASSiVE, which is explained in more detail below. S_T has a fix association with the task to be solved according to the given task request. For the example task depicted in Fig. 10 the target state is defined as: $S_T = \{OC_2, OC_5, OC_6\}$, i.e. the task request "Fetch cup" is associated with the target situation where the cup has finally been placed on the table and the empty gripper as well as the empty cupboard remain.
So far, the explanations concentrated on the design of a PS_A. Its information is used to drive the planning process of a task request. Here, according to the conceptual sequence of activities depicted in Fig. 9, the task related PS_A is loaded first. Afterwards, the so-called process of initial monitoring is started. This process determines the initial situation S_I of the chosen PS_A that corresponds to the current environmental situation. From out this situation a path to the target situation can be processed.

Initial Monitoring

The situation monitoring process is controlled by the information contained within the set of situations associated with a PS_A and is executed in a semi-autonomous manner also (Prenzel, 2005). As illustrated in Fig. 9, the monitoring process consists of the determination of the first order logic facts that are assigned to the different *OC*s. Within the given process-structure, the set of facts listed in Table 2 has to be determined during initial monitoring.
To give an illustration of the monitoring approach in MASSiVE, the example of finding the cup to be retrieved from the cupboard will be taken into account. Fig. 11 shows a respective setup of a user interaction that will occur in this case. Here, the FRIEND-II system has been moved in front of a cupboard in which several cups are placed. During this user interaction the user is requested by the system to control the pan-tilt head devices (and thus move the cameras), so that the object of interest is in their sight (the left camera image is displayed to the user). During the movement of the cameras, the image processing system segments objects positioned in the middle of the camera image in real-time. The segmentation result is overlaid to the image, so that the user is able to decide whether the desired object is highlighted or not.

Fact	Description
HoldsNothing(G)	The gripper G holds nothing.
IsInFreePos(G)	The gripper G is in a free position.
IsInsideContainer(C, B)	The cup C in inside the container B (box).
HasFreeStoringSpace(B, C)	The box B has enough free space for the cup C.
IsGripped(G, C)	The cup C is gripped by the gripper G.
IsPlacedOn(C, T)	The cup C in place on top of the table T.

Table 2. Set of facts in "Fetch Cup" scenario.



The system is not able to detect the cup to be gripped.
Please move the camera perspective until the cup is highlighted.

Fig. 11. Experimental setup for finding a cup in a cupboard.

With respect to finish the initial monitoring, the determination of other task relevant objects is performed analogously within a user interaction in case the autonomous object detection fails (or is not unique). After the initial object detection is completed, additional object specific sub-symbolic information is extracted according to the concept of object-anchoring (Prenzel, 2005). This information is stored in the sub-symbolic layer of the system's world model (Fig. 5), to be accessible for all skills that are executed within the reactive layer. Thus, during further task processing steps the degree of the system's autonomy increases with the increasing amount of acquired environmental information.

Task Planning and Execution

According to Fig. 9 task planning and execution takes place, after the initial situation S_I is monitored successfully. Planning is necessary in order to determine a sequence of operations that is adapted to the current environmental situation. The applied planning method is equivalent on both levels of process-structures, i.e. PS_A and PS_E, as exemplified in the following: In the scenario of Fig. 10 with its assigned PS_A it shall be assumed that the initial situation has been determined during initial monitoring. Thus, the task planning objective is to transform this initial situation $S_I = \{OC_1, OC_2, OC_3\}$ into the target situation $S_T = \{OC_2, OC_5, OC_6\}$. Since the transformation of situations is realized by means of the execution of operations, the generation of action sequences is reduced to a graph search problem. Here, the search is realized by means of the Dijkstra algorithm (Martens, 2003a). According to the assumed initial situation S_I, the first *COP* in the sequence of planned *COP*s is *GraspObjectInContainer*. Because this *COP* cannot be executed directly within the architecture, it has to be decomposed to a level of abstraction that deals with *EEOP*s.

The result of the *COP* decomposition, i.e. its assigned elementary process-structure (PS_E), is depicted in Fig. 12. Here, to reduce the complexity of the illustration, only a segment of the complete PS_E is shown – the upper part, as indicated by the dashed box, has been left out. However, the principle of a PS_E can be clarified as follows: The flow of control is modeled on the basis of a Petri-net. As depicted, transitions of this Petri-net are arranged in pairs (or tuples), which represent the execution of an *EEOP*. Each transition of a pair (or tuple) represents the execution of an *EEOP* with different execution results. This is denoted by the different return values. The different nature of *EEOP*s, as introduced in Chapter 4.1, is specified with the help of the prefix in the transition name: *DC* stands for "direct control" of an actuator, *USER* for "user interaction and *REAC* for "reactive operation", i.e. an *EEOP*, which is in fact a closed control loop that couples sensors and actuators. *EEOP*s of the fourth category, the "monitoring operations", with the prefix *MON,* are not part of the depicted segment of PS_E.

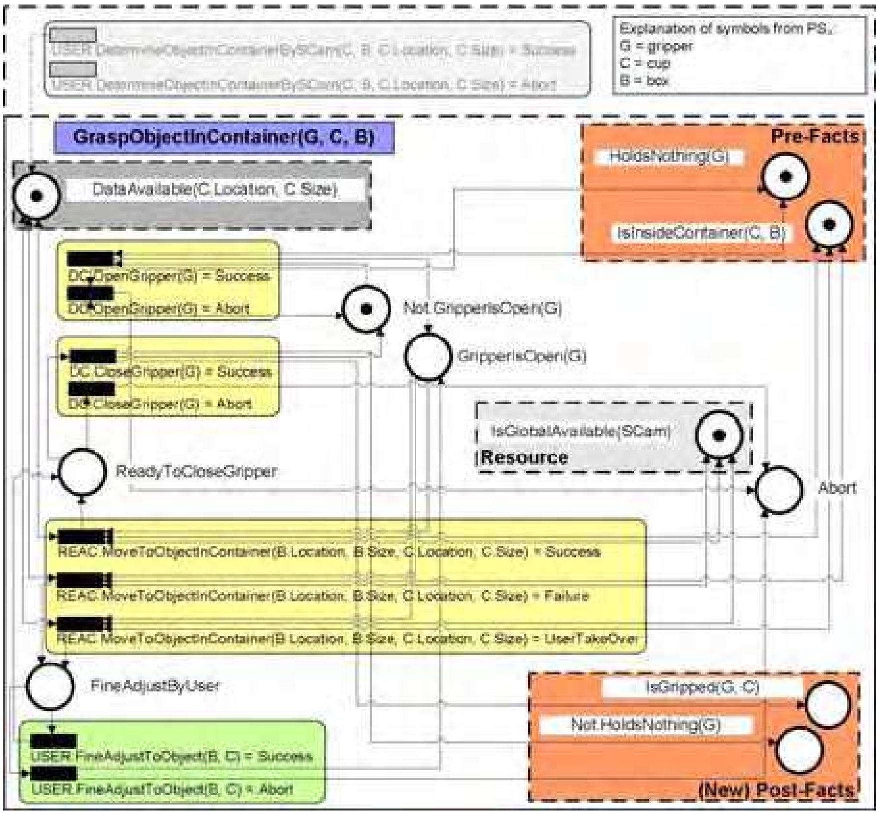

USER.DetermineObjectInContainerBySCam(C, B, C.Location, C.Size) = Success
USER.DetermineObjectInContainerBySCam(C, B, C.Location, C.Size) = Abort
Explanation of symbols from PS_A:
G = gripper
C = cup
B = box
GraspObjectInContainer(G, C, B)
Pre-Facts
HoldsNothing(G)
IsInsideContainer(C, B)
DataAvailable(C.Location, C.Size)
DC.OpenGripper(G) = Success
DC.OpenGripper(G) = Abort
Not.GripperIsOpen(G)
GripperIsOpen(G)
DC.CloseGripper(G) = Success
DC.CloseGripper(G) = Abort
IsGlobalAvailable(SCam)
Resource
ReadyToCloseGripper
Abort
REAC.MoveToObjectInContainer(B.Location, B.Size, C.Location, C.Size) = Success
REAC.MoveToObjectInContainer(B.Location, B.Size, C.Location, C.Size) = Failure
REAC.MoveToObjectInContainer(B.Location, B.Size, C.Location, C.Size) = UserTakeOver
FineAdjustByUser
IsGripped(G, C)
Not.HoldsNothing(G)
USER.FineAdjustToObject(B, C) = Success
USER.FineAdjustToObject(B, C) = Abort
(New) Post-Facts

Fig. 12. Section of elementary process-structure (PS_E) representing the *COP* "*GraspObjectInContainer*" (upper part is left out).

Task planning on the basis of the Petri-net is equivalent to the planning process performed on the basis of the PS_A. Starting from an initial situation (initial marking with tokens in the Petri-net) the shortest sequence of operations (fired transitions) is calculated by means of the

Dijkstra algorithm. During task execution, a single transition from out a pair (or tuple respectively) is fired according to the actual return value of the respective *EEOP*. If the return value fits to the pre-planned one, the next *EEOP* of the plan is chosen. Otherwise, a replanning step has to be performed. Because decomposition has the meaning of replacing the formal parameters of a PS_E with the actual parameters determined by the PS_A, the resulting sequence of *EEOP*s considers the objects that have been localized during the initial monitoring process.

The elementary process-structures in the form of a Petri-net include different modeling aspects and represent the following:

- Control flow (e.g. controlled involvement of the user into data monitoring or direct actuator control)
- Resource usage (e.g. *IsGlobalAvailable(SCam*[9]*)*)
- Data flow and data availability (e.g. *DataAvailable(C.Location, C.Size)*)
- Interconnection to abstract layer and to the world model via facts (e.g. *HoldsNothing(G)*)

The advantage of using Petri-nets as a modeling tool is the availability of mathematical methods for the structural analysis of PS_E. Due to the illustrated stringent and application specific syntax within a PS_E, offline verification becomes possible. This includes the verification with respect to dead-locks, resource-conflicts, modeling errors, state-reachability, availability of sub-symbolic data and the direction of data flow (Martens, 2003b).

During runtime, the initial marking in a PS_E is set according to the current status of task execution. For the example depicted in Fig. 12, since the task participating object *B* (box) and *C* (cup) have already been identified during the initial monitoring process, a new acquisition of the data is not required. Thus, the token in the place "*DataAvailable(C.Location, C.Size)*" will be part of the initial marking. In any other case, where this information which is required for the subsequent manipulative operation *REAC.MoveToObjectInContainer* is not yet available, it will be generated within the upper part (not depicted) of the PS_E via monitoring *EEOP*s or user interactions respectively. The target marking in a PS_E, which is required to apply planning by graph search, is specified during the definition of a PS_E. One rule with respect to the target marking specification is that it has to include the post-facts associated with the *COP* assigned to a certain PS_E. Post facts in Fig. 12 are *IsInsideContainer(C, B)* (also pre-fact), *IsGripped(C, G)* and *Not.HoldsNothing(G)*. With the help of post- and pre-facts, the connection and consistency between the sub-symbolic control level (PS_E) and the abstract level (PS_A) are guaranteed.

Here, according to the given state of the Petri-net, the manipulative *EEOP*s *DC.OpenGripper, REAC.MoveToObjectInContainer* and *DC.CloseGripper* will be executed. Within the reactive manipulation *EEOP*, a motion planning algorithm is started, which retrieves and operates on the previously (during initial monitoring) generated environmental information from the world model (Prenzel et al., 2006; Ojdanic et al., 2006). Due to the structure of the task-knowledge contained within the PS_E and the usage of the Dijkstra algorithm, the result of the planning process is that autonomous *EEOP*s are executed first and user interaction *EEOP*s follow in case the system fails (this general idea is also illustrated in Fig. 9). Furthermore, manipulative *EEOP*s can be interrupted by the user, e.g. in case he can already foresee a possible collision. This is indicated by the "[user interrupt]" transition in Fig. 9 or the third

[9] SCam = Stereo-camera system

transition of the reactive *EEOP* in Fig. 12 respectively. The concrete and more detailed description of this kind of interaction enforced by the user is discussed in (Lüth et al., 2007). The complete planning and execution cycle is continued until the given task is completed or has been aborted, e.g. according to the user's intention or due to a critical and irresolvable system status.

The described approach leads to a system that is able to operate robustly, even if it is not able to execute all steps fully autonomously. Moreover, it supports the evolutionary development of the system with increasing autonomy. This means, it is possible to initially realize a rehabilitation robotic support scenario with a low level of autonomy and to replace missing autonomous functionality with user interactions. Subsequently and in an evolutionary manner, more and more user interactions are replaced by autonomous operations by developing more intelligent skill algorithms or using additional sensors and actuators.

For recapitulating this section it is referred to Fig. 9. After the user's initial task selection, the first system controlled user interaction takes places within the initial monitoring. Initial identification tasks that cannot be solved fully autonomously without unreasonable increase of technical complexity are given back to the user to be solved by him. The same approach is taken into account within the ongoing process of task-execution.

5. Conclusion

This paper gives an overview of the functional capabilities of the rehabilitation robotics systems FRIEND-I and FRIEND-II as well as an introspective view of the used underlying control principles and applied software techniques. It is shown that during the different development steps of FRIEND-I the control of the system evolved from out direct speech control of the manipulator up the fully autonomous execution of a "serve-drink task". It turned out that the control of a rehabilitation robot like FRIEND-I becomes possible for a person without manipulative capabilities in his or her arms and hands. But there is still the need for more robustness and reliability in execution, flexibility in usage and relief for its users, if the overall objective of 1.5 hours of autonomy for daily life activities has to be reached. The most important lesson learned within the FRIEND-I project is the application of the semi-autonomous task execution principle. The motivation is that the involvement of the user's cognitive capabilities has the potential of reducing the system's complexity dramatically, so that a technical manageable system emerges.

The consequent implementation of this principle has been realized within the successor system FRIEND-II. On the one hand FRIEND-II comes up with new sophisticated hardware components like a 7 DOF manipulator or a force-torque controlled gripper. It is evident that the usage of these hardware components leads to more robustness in execution. On the other hand FRIEND-II has been software-technically designed for its application within an intelligent environment, where different daily life as well as working scenarios, as specified within the AMaRob project, will be managed. It is shown that the beforehand mentioned requirements of user involvement and distributed execution, as necessary for its application within the intelligent environment, are satisfied by the multilayer architecture MASSiVE. For user involvement, MASSiVE offers an infrastructure for task-knowledge driven task planning, whereas the user interaction is an integral part within this knowledge. During task execution the system requests the user whenever it has to come to a decision on a cognitive level and the algorithmic realization of this process it too complex and error prone. But even though the user is requested, he or she is relived from tiresome and complicated tasks like remembering the necessary sequence of actions that have to be fulfilled before the execution of the next steps becomes possible. Within MASSiVE's HMI-level the current task context is known, so that a specific and efficient request limited to the current context can be initiated. This has been demonstrated by the detection of a certain cup in a cupboard full of cups, where the application of a well known image segmentation algorithm leads to the same results, from the user's point of view, as a complex image analysis.

The requirement for reactive and distributed execution is satisfied by MASSiVE's reactive layer, which is controlled by the intermediating sequencer. From the sequencer's point of view the reactive layer is as a pool of skills, from which some have to be selected and combined according to a task plan in order to reach the goal of a requested task. Within this context the reactive layer encapsulates the underlying hardware and offers environmental information on a pre-processed symbolic level. Because each hardware component is managed by a separate software server, which itself is managed within the reactive layer according to a well defined server interface concept, the inclusion of new components, like the smart devices within the intelligent environment, has no design consequences for the superior levels like the HMI-layer or the sequencer respectively.

The first task scenario which has been managed with FRIEND-II on basis of MASSiVE is the "serve-drink task" also. Even though the functional improvement with respect to FRIEND-I is not evident, its execution is more flexible from the viewpoint of software technique. Within FRIEND-I subtasks of the "serve-drink task", like grasping, replacing or putting down objects, were programmed in a pre-determined sequence. Within FRIEND-II these subtasks evolve from out a planning process performed on basis of task-knowledge. Due to this flexibility the next development steps concentrate on the implementation of new subtasks, i.e. skills that will be integrated into the already existing task-knowledge data-basis. On the basis of this extended task-knowledge new tasks, as specified within the AMaRob project, will be managed within the next two years.

Another area of active research, which is also important with respect to further improvements of rehabilitation robots, is the system control via non-invasive brain-computer interfaces (BCIs). In (Lüth et al., 2007) first results are presented that show the feasibility of user interactions in MASSiVE solely on the basis of brain signal interpretation. The applied principle is the detection of either the characteristic P300 waveform or steady state visual evoked potentials (SSVEP) in the visual cortex of the user, who is concentrating on a continuously blinking light source. The bandwidth of this kind of input media is still very low and also it is currently not yet possible to realize user interactions with fast reaction times. However, due to ongoing improvements of this technique, a very promising opportunity to command a robotic system arises for highly handicapped users.

6. References

Bien, Z. Z.; Song, W.-K; Kwon, D.-S.; Chung, M.-J.; Park, H.-S.; Kim, D.-J.; Kim, J.-H. & Lee, K. (2001). A Wheelchair Robot System and its Various Interface Methods for the Disabled Persons, *Proceedings of the 1st Workshop on Technical Challenge for Dependable Robots in Human Environments,* Seoul, Korea, May 21-22, 2001

Bolmsjö, G. & Neveryd, H. (1995). WALKY, an Ultrasonic Navigating Mobile Robot for the Disabled, *Proceedings of the 2nd TIDE Congress*, pp. 366-370, ISBN 978-90-5199-220-5, Paris, France, April 1995, IOS Press, Amsterdam

Bonasso, R.; Kortenkamp, D.; Schreckenghost, D. & Ryan, D. (1998). Three tier architecture for controlling space life support systems, *Proceedings of IEEE International Joint Symposium on Intelligence and Systems (SIS) 1998,* pp. 195-201, ISBN 0-8186-8548-4, Rockville, MD, USA, May 1998

Borgerding, B.; Ivlev, O.; Martens, C.; Ruchel, N. & Gräser, A. (1999). FRIEND - Functional Robot Arm with User Friendly Interface for Disabled People, In:, *Assistive Technology on the Threshold of the New Millennium: AAATE'99, Proceedings of the 5th European Conference for the Advancement of Assistive Technology,* C. Bühler, H. Knops (Ed.), pp. 286-290, , IOS Press, ISBN 1-58603-001-9, 1st-4th November 1999, Amsterdam

Busnel, M.; Coulon-Lauture, F.; Détriché, J.-M.; Le Claire, G. & Lesigne, B. (1999). The robotized workstation –MASTER- for users with tetraplegia: Description and evaluation. *Journal of Rehabilitation Research and Development*, Vol. 36, No. 3, July 1999, pp. 217-229, ISSN 0748-7711

Cao, T. & Sanderson, A. C. (1998). AND/OR net representation for robotic task sequence planning, *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 28, Issue 2, May 1998, pp. 204–218, ISSN 1094-6977

Casals, A.; Merchan, R.; Portell, E.; Cufi; X. & Contijoch, J. (1999). CAPDI: A Robotized Kitchen for the Disabled and Elderly, *Proceedings of the 5th European Conference for the Advancement of Assistive Technology (AAATE)*, pp. 346-351, ISBN 1-58603-001-9, Düsseldorf, Germany, November 1-4, 1999, IOS Press, Amsterdam

Colle, E.; Rybawzyk, Y. & Hoppenot, P. (2002). Arph: an assistant robot for disabled people, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics,* pp. 176–181, ISBN 0-7803-7437-1, Yasmine Hammamet, Tunisia., 6-9 October 2002unKhoc cho nho thuong voi trong long, khoc cho noi sau nhe nhu khong. Bao nhieu yeu thuong nhung ngay qua da tan theo khoi may bay that xa... http://nhatquanglan1.0catch.com

Coste-Maniere, E. & Simmons, R. (2000). Architecture, the backbone of robotic systems, *Proceedings of International Conference on Robotics and Automation,* pp. 67–72, ISBN 0-7803-5886-4, San Francisco, CA, USA, April 2000

Dallaway, J.L. & Jackson, R.D. (1993). CURL - today and tomorrow, *Proceedings of the 2nd Nordic Workshop on Rehabilitation Robotics,* Orup Rehab Center, Sweden (HADAR, Malmö),, September 1993

Dallaway, J. L.; Jackson, R. D. & Timmer, P.H.A. (1995). Rehabilitation Robotics in Europe. *IEEE Transactions on Rehabilitation Engineering 1995*, Vol. 3, No. 1, March, 1995 pp. 33-45, ISSN 1063-6528

Dario, P.; Dillman, R. & Christensen, H. I. (2004). EURON research roadmaps, *Key Area 1 on Research Coordination*, http://www.euron.org

DLR. (2007). Key research program "Leitinnovation Servicereobotik", *www.service-robotik.de*, DLR - Project executing organization for German Federal Ministry of Education and Research ("Bundesministerium für Bildung und Forschung")

Eftring, H. (1994). Robot control methods and results from user trials on the RAID workstation, *Proceedings of the 4th International Conference on Rehabilitation Robotics (ICORR)*, pp. 97-101, Wilmington, Delaware, USA, June 1994

Engelberger, J. F. (1989). *Robotics in Service,* MIT Press, ISBN 0-2620-5042-0, Cambridge, MA, USA

Fikes, R. & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence, Vol. 2*, pp. 189-208. Reprinted in Readings in Planning, J. Allen, J. Hendler, and A. Tate (Ed.), 1990, Morgan Kaufmann Publishers, San Mateo, California

Gamma, E.; Helm, R.; Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison-Wesley Professional, 1st edition ed., 1995, ISBN 0-2016-3361-2

Hillman, M. R. et al. (1999). Design of a Wheelchair Mounted Robot, *Proceedings of the 5th European Conference for the Advancement of Assistive Technology (AAATE)*, pp. 316-321, ISBN 1-58603-001-9, Düsseldorf, Germany, November 1-4, 1999, IOS Press, Amsterdam

IAT. (2007). AMaRob – Autonomous Manipulator Control for Rehabilitation Robots, *Project-webpage, www.amarob.de*

Innocenti, C.; Mondino, G.; Regis, P. & Sandini, G. (1994). Trajectory Planning and Real-time Control of an Autonomous Mobile Robot Equipped with Vision and Ultrasonic Sensors, *Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 1994*, pp. 1861-1866, ISBN 0-7803-1933-8, Munich, Germany, September 12th-16th, 1994

Kawamura, K.; Bagchi, S.; Iskarous, M. & Bishay, M. (1995). Intelligent robotic systems in service of the disabled, *IEEE Transactions on Rehabilitation Engineering*, Vol. 3, No. 1 , Mar 1995, pp. 14 –21

Kazi, Z. (1996). Multimodally Controlled Intelligent Assistive Robot*, Proceedings of 16th Annual RESNA Conferenc*e, pp. 348-350 Salt Palace Convention Center, Salt Lake City, Utah, USA, June 1996

Kazi, Z.; Chen, S.; Beitler, M.; Chester, D. & Foulds, R. (1997). Grasping at Straws: An Intelligent Multimodal Assistive Robot, *Proceedings of the 5th ICORR'97,* pp. 87-90, Bath Institute of Medical Engineering, University of Bath, UK, April 1997

Korondi, P. & Hashimoto, H. (2003). Intelligent Space as an Integrated Intelligent System, *Keynote Paper in Proceedings of the International Conference on Electrical Drives and Power Electronics,* pp. 24-31, 2003

Lang, O.; Martens, C & Gräser, A. (2000). Realisation of a Semiautonomous Grippingskill for the Support of Disabled People, *Proceedings of Robotik 2000*, pp. 273-278, ISBN 3-18-091552-8, Berlin, 29./30. June 2000, VDI Berichte 1552, VDI Verlag GmbH, Düsseldorf

Laschi, C.; Teti, G.; Tamburrini, G.; Datteri, E. & Dario, P. (2001). Adaptable semi-autonomy in personal robots, *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication,* pp. 152–157, ISBN 0-7803-7222-0, Bordeaux, Paris, France, September 2001

Lüth, T.; Ojdanić, D.; Friman, O.; Prenzel, O. & Gräser, A. (2007). Direct control in a rehabilitation robotic system via a Brain-Computer Interface, Proceedings of the International Conference on Rehabilitation Robotics ICORR, Noordwijk, Netherlands, 13-15 June 2007

Mahoney, R. M. (2001). The Raptor Wheelchair Robot System, *Proceedings of the 7th International Conference on Rehabilitation Robotics (ICORR) 2001*, Mokhtari, M. (Ed.) Integration of Assistive Technology in the Information Age, pp. 135-141, ISBN 1-58603-171-6, Evry, France, IOS Press, Amsterdam

Martens, C.; Lang, O.; Ruchel, N.; Ivlev, O. & Gräser, A. (2001). A FRIEND for Assisting Handicapped People. *IEEE Robotics and Automation Magazine,* Vol. 7, No. 1, March 2001, pp. 57-65, ISSN 1070-9932

Martens, C.; Kim, D.-J.; Han, J.-S.; Gräser, A. & Bien, Z. Z. (2002). Concept for a modified hybrid multi-layer control-architecture for rehabilitation robots, *Proceedings of the*

3rd International Workshop on Human-friendly Robotic Systems, pp. 49-54 Daejeon, Korea , January 21-22, 2002

Martens, C. (2003a). Generation of parallel executable control sequences for rehabilitation robotic systems on the basis of hierachical petri-net based task representation, In: *Methods and Applications in Automation,* B. Lohmann, A. Gräser, (Ed.), pp. 73–85, Shaker Verlag, ISBN 3-8322-4502-2, Aachen

Martens, C. (2003b). *Teilautonome Aufgabenbearbeitung bei Rehabilitationsrobotern mit Manipulator,* Ph.D. thesis University of Bremen, Shaker-Verlag, ISBN 3-8322-2400-9, Aachen

Martens, C. (2005). Task oriented programming of service-robots on the basis of process-structures, In: *Methods and Applications in Automation,* B. Lohmann, A. Gräser, (Ed.), pp. 45–56, Shaker Verlag, ISBN 3-8322-4502-2, Aachen

May, T. & Schäffer, C. (1999). Care-O-bot: A System for Assisting Elderly and Disabled Persons in Home Environments, *Proceedings of the 5th European Conference for the Advancement of Assistive Technology (AAATE)*, pp. 340-345, ISBN 1-58603-001-9, Düsseldorf, Germany, November 1-4, 1999, IOS Press, Amsterdam

Mokhtari, M. & Amni, C. (2001). Assistive Technology for the Disabled People: Should it Work? The French Approach, *Proceedings of the 2nd International Workshop on Human-friendly Welfare Robotic Systems*, pp. 134-136, KAIST, Daejeon, Korea January 2001, HWRS-ERC, Daejeon

Ojdanic, D.; Ivlev, O. & Gräser, A. (2006). A new fast motion planning approach for dexterous manipulators in 3D-Cartesian space, *Proceedings of ISR-Robotik Joint Conference on Robotics,* p. 49, ISBN 3-18-091956-6, International Congress Centre Munich, Germany, May 15-17 2006, VDI Wissensforum IWB GmbH, Düsseldorf, Germany

Prenzel, O. (2005). Semi-autonomous object anchoring for service-robots, In: *Methods and Applications in Automation,* B. Lohmann, A. Gräser, (Ed.), pp. 57–68, Shaker Verlag, ISBN 3-8322-4502-2, Aachen

Prenzel, O.; Ojdanic, D. & Gräser, A. (2006). Manipulative Robotic Tasks in Smart Home Environments, *Proceedings of 4th International Conference On Smart Homes and Health Telematics (ICOST),* pp. 172-179, ISBN 1-58603-623-8, ISSN 1383-813X, Belfast, Northern-Ireland, UK, June 26-28, 2006, Series: Assistive Techology Research Series, IOS Press, (Ed.) C. Nugent, J.C. Augusto, Vol. 19

Radchenko, O.; Martens, C.; Pape, A.; She, H.; Volosyak, I. & Gräser, A. (2004). FRIEND – An Intelligent Assistant in Daily Life, In: *Advances in Rehabilitation Robotics: Human-friendly Technologies on Movement Assistance and Restoration for People with Disabilities - Part II Rehabilitation Robots for Assistance of Human Movements, II.1 Conceptions and Experimental Design,* Ed.: Z. Zenn Bien, Dimitar Stefanov, pp. 95–126, Springer-Verlag, ISBN 3-540-21986-2, Heidelberg, Germany

She, H.; Martens, C. & Gräser, A. (2003a). Application of Programming by Demonstration in the Rehabilitation Robotic System FRIEND, *Proceedings of the 8th International Conference on Rehabilitation Robotics (ICORR) 2003,* pp. 39-42, ISBN 89-88366-09-3, Daejeon, Korea, 23-25 April 2003, HWRS-ERC

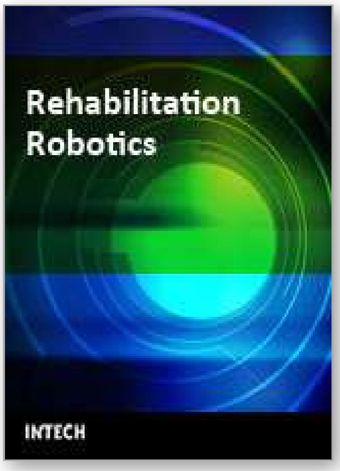
She, H.; Martens, C.; Gräser, A.; Kim, D.-J.; Lee, H.-E. & Bien, Z. Z. (2003b). Soft Computing-based Robust Contact/Non-contact Detection during Serving a Beverage Task, *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2003),* pp. 613-617, Marriott Pavilion Downtown Hotel, St. Louis, MO, May 25-28, 2003

Van der Loos, H.F.M. (1995). Lessons Learned in the Application of Robotics Technology to Field of Rehabilitation. *IEEE Transactions on Rehabilitation Engineering*, Vol. 3, No. 1, March 1995, pp. 46-55, ISSN 1063-6528

Van der Loos, H.F.M.; Wagner, J.J.; Smaby, N.; Chang, K.-S.; Madrigal, O.; Leifer, L.J.; Khatib, O. (1999). ProVAR assistive robot system architecture, *Proceedings of International Conference on Robotics and Automation (ICRA) 1999*, pp. 741-746, ISBN 0-7803-5180-0, Marriott Hotel, Renaissance Center, Detroit, Michigan, May 10-15, 1999, IEEE Robotics and Automation Society

Volosyak, I.; Radchenko, O.; Pape, A., Martens, C.; She, H.; Wendland, E. & Gräser, A. (2003). Smart tray for the support of a wheelchair mounted manipulator, *Proceedings of the International Conference on Economic, Engineering and Manufacturing Systems ICEEMS 2003,* ISBN 973-635-215-3, Brasov, Romania, 23-24 October, 2003

Wagner, J.J; Van der Loos, H.F.M; Leifer, L.J. (1998). Dual-Agent User Interface for an Assistive Robot, *Proceedings of the 21st Annual RESNA Conference*, pp. 286-288, ISSN 0883-4741, ISBN 0-932101-4-2, Minneapolis, MN. Washington DC, June 1998, RESNA Press



Rehabilitation Robotics
Edited by Sashi S Kommu

ISBN 978-3-902613-04-2
Hard cover, 648 pages
Publisher I-Tech Education and Publishing
Published online 01, August, 2007
Published in print edition August, 2007

The coupling of several areas of the medical field with recent advances in robotic systems has seen a paradigm shift in our approach to selected sectors of medical care, especially over the last decade. Rehabilitation medicine is one such area. The development of advanced robotic systems has ushered with it an exponential number of trials and experiments aimed at optimising restoration of quality of life to those who are physically debilitated. Despite these developments, there remains a paucity in the presentation of these advances in the form of a comprehensive tool. This book was written to present the most recent advances in rehabilitation robotics known to date from the perspective of some of the leading experts in the field and presents an interesting array of developments put into 33 comprehensive chapters. The chapters are presented in a way that the reader will get a seamless impression of the current concepts of optimal modes of both experimental and ap- plicable roles of robotic devices.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821