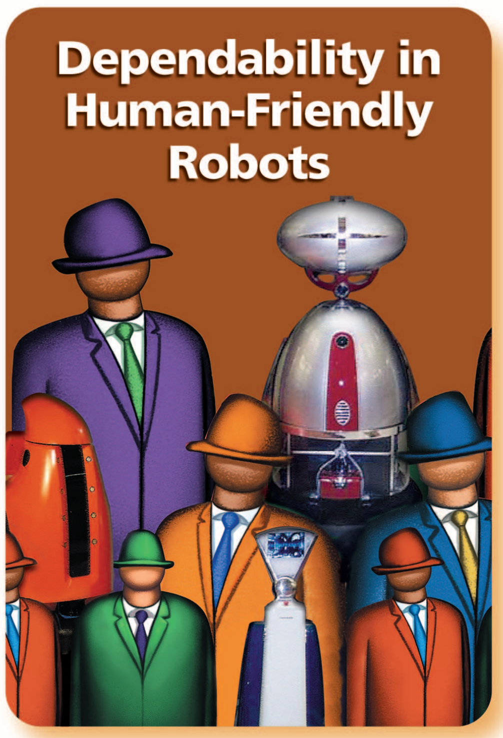
Issues for Dependable Operation in Direct Cooperation with Humans

Mobile Robot Assistants

BY BIRGIT GRAF, MATTHIAS HANS, and ROLF DIETER SCHRAFT



Mobile robots equipped with mechanisms for communication, interaction, and behaviors are employed more and more outside traditional manufacturing applications, such as in museums or exposition areas. Most installations, however, are only temporary and not product ready yet (see the overview in [1]). In the near future, service robots, able to cooperate with and assist humans, will be available for sale to everyone. Operating robots in unmodified natural environments imposes requirements on the robots that are incomparably higher than demands made on the capabilities of industrial robots [2]. Dependability is one of the main issues in the development of such robot assistants.

Dependability involves physical safety on one side and operating robustness—consisting of availability, reliability, and maintainability—on the other. Integrity as a prerequisite for operation robustness has been mentioned further [3] and [4]. Dependability must be obtained for each single component of a robot and for the whole system, which—designed to fulfill a certain task—might be more than just a sum of its components. In order to prevent faults efficiently, modular and structured systems are mostly preferable. They enable the testing of each component separately. However, fault prevention alone does not lead to dependability; fault tolerance and error recovery should be considered as additional means.

This article is structured as follows: First, we will discuss dependable navigation as a basic feature for operating mobile robots among humans. One of the most common accidents caused by industrial robots is a person being hit by the robot [5]. For stationary robots the responsibility lies partly with the user; safety measures, such as keeping a certain distance from the robot, must be obeyed. For mobile robots, however, all responsibility lies with the vehicle; therefore, dependable navigation is a major issue. The importance of each navigation module according to the previously mentioned dependability components and resulting specifications for the development of such modules will be analyzed. Technical implementations of dependable localization, path planning, and obstacle avoidance systems, developed at the Fraunhofer Institute of Manufacturing, Engineering, and Automation (IPA), will be presented.

Second, we will discuss the dependable execution of fetch-and-carry tasks among humans. The crucial components for accomplishing this task will be specified and requirements for the implementation developed. A modular control architecture for task planning and execution will be described. Furthermore, a new method for sensor-based object manipulation—including object detection, path

1070-9932/04/$20.00©2004 IEEE

planning of the manipulator, and sensor-based grasping—will be introduced.

Finally, experimental results, including the long-term installation of three entertainment robots in a museum in Berlin and several short-term installations of Fraunhofer IPA's robots Care-O-bot I and II using the previously described software components, will be presented and evaluated. Here, the discussion is not limited to scientific test cases. In fact, some of the installations presented describe mobile robots that have been sold as a product and that must run and execute their task dependably for many years.

Dependable Navigation

The Concept of Dependable Navigation

In order to dependably navigate in environments populated by people, the navigation system of an interactive robot assistant must be capable of dealing with dynamic environments. It must guarantee that people will not be hurt nor objects damaged at any time. In order to specify the requirements for a dependable navigation system, the different possibilities of implementing a mobile robot's navigation system must be analyzed. Furthermore, the degree of dependability must be evaluated for each solution.

Sensors

Mobile robot navigation is often based on laser range finders, sonar, and/or vision sensors. In order to prevent navigation based on erroneous sensor data, the correct operation of such safety-critical sensors must be ensured at all times. The simplest system could measure the sensor input data and produce an error if no sensor data is available. More sophisticated systems integrate error recovery methods such as stopping the robot and trying to reconnect to the sensor. These types of error recovery, however, reduce the availability and reliability of the robot because the robot would be blocked until the problem was solved. To avoid this reduction, an ideal system would be equipped with redundant sensors that would enable constant operation without pause. Redundant sensor systems might not only be useful to recover hardware errors immediately but also increase the fault tolerance of the system.

Localization and Mapping

For large mobile robots, one of the major safety hazards is the possibility of the robot losing its position. This could lead to the robot leaving its assigned operation area, falling down stairs and hurting people, or damaging its environment. Physical safety can only be guaranteed if the robot knows its position (fault prevention) and is able to recover position loss as quickly as possible (fault recovery).

Local and global localization systems can be distinguished: local localization only works if the robot is approximately at the position where it "thinks" it is and only compensates small position errors. Global localization allows a robot to always find its current position by analyzing and comparing unique features in the environment. In this respect, global localization systems—or at least local methods with integrated position recovery technique—are more dependable than pure local systems. In our opinion, global localization methods should be implemented for increased localization dependability. The integration of simultaneous localization and mapping (SLAM) is an ideal configuration because it enables the robot not only to localize itself in a static environment but—by continuously updating its environment map—also enables the machine to adapt itself to changing environments. Since the personnel operating the robot must no longer be involved in reprogramming the environment, SLAM leads to an increase not only in the safety but also in the integrity and maintainability of the robot. With respect to availability and reliability, correct localization is also an issue. A robot that is not localized correctly will not be available to assist its user. It will not succeed in driving to any specified target and will therefore fail in executing a given task.

Based on these considerations, we propose to classify a mobile robot's localization system in terms of dependability as illustrated in Figure 1.

	Safety	Integrity	Availability	Reliability	Maintainability
Local Localization System	Low	Low	Low	Low	Medium
Global Localization System	Low	Low	Medium	Medium	Medium
SLAM System	High	High	High	High	High

Figure 1. *Dependability of different localization systems.*

Motion Planning and Obstacle Avoidance

Mobile robots navigating among humans should be able to reach a specified target without bumping into people. Therefore, a dependable motion-planning and obstacle-avoidance system is required. In the simplest case, a robot would turn to a given target, drive towards it in a straight line, and stop if obstacles are detected in its path. Such a system, although safe, would lack in terms of reliability and availability because the robot might be blocked by an obstacle for a long time or get trapped in "dead ends." More elaborate navigation systems enable a robot not only to stop in front of obstacles but to surround them (e.g., by specifying an intermediate target). Based on previously acquired global environmental data, a more intelligent robot might be able to plan an optimal path that leads it around stationary obstacles. The optimal solution consists of a planner that is able to generate a path based on an environmental map and modify it when detecting new obstacles. Integrity is given by a path planning and modification system that works quickly and efficiently, i.e., that keeps the nature of an initially planned path and follows it smoothly without any stops or detours.

We propose to classify a mobile robot's motion planning and obstacle avoidance system in terms of dependability as illustrated in Figure 2.

Implementation of Navigation Modules

Sensors

Each mobile robot developed at Fraunhofer IPA is equipped with a laser scanner which, in our opinion, is essential for safe and reliable navigation. Furthermore, all robots are also equipped with a rubber bumper as a failsafe safety system. Additional sensors are included depending on the application and will be described in the corresponding sections.

Localization

Localization is based on data acquired from the wheel encoders and the gyroscope. However, while using odometric sensors, small errors are unavoidable and add up over time. Therefore, the robot's surroundings are modeled in an environment map and matched with laser scanner data. By comparing features detected in the natural environment of the robot with the given map, the robot corrects its position. Information acquired by this method is merged with odometric data using a Kalman filter. The landmarks of the environment are represented by a few simple features types such as straight wall segments or reflector marks.

PolarBug Obstacle Avoidance Module

The PolarBug obstacle avoidance module, based on the VisBug [6] method, can be applied in simple environments where path planning is not required. It enables the robot to quickly surround dynamic obstacles. This algorithm has been developed especially for obstacle detection with a laser scanner, as well as for fast reaction and navigation in unsteady environments. Laser scanner data is processed directly in polar coordinates, which increases the efficiency and speed of the algorithm.

All of the measurements of the laser scanner are evaluated. If obstacles have been detected between the current position of the robot and the given target, an intermediate position is calculated to lead the robot around the obstacles as quickly as possible (Figure 3). Considering several parameters, such as width and depth of passage, deviation of passage from direct line to target, and distance of intermediate position to robot, and final target position, an optimal passage is chosen. All relevant factors are joined using a fuzzy logic approach.

Path Planning and Elastic Band

Path planning allows a mobile robot to find a continuous trajectory from the start configuration to the goal configuration in complex environments. For global planning, an environmental map is used together with the current and desired positions of the robot. Local replanning that considers sensor information is required when some point of the calculated path is not reachable by the robot, e.g., due to a dynamic obstacle.

Several algorithms have been developed for global and local path planning [7]. Figure 4 displays path planning based on rapidly exploring random trees (RRT) [8] and on potential grids with wavefront expansion. Additionally, path planning based on quad trees and on a visibility graph [9] have been implemented. Depending on the nature of the operation environment of the robot, the best method can be selected.

The path generated through a visibility graph is close to optimum in terms of distance to move. Once the visibility graph has been built up, the algorithm runs pretty quickly; however, the process for building the graph may take a long time. Therefore, the algorithm is suitable for static environments where the graph is built up only once, but not for dynamic environments where the graph must be recalculated during runtime.

In narrow corridors, the random tree method takes a long time because in these cases just a few points are allowed and they are pretty difficult to locate randomly.

	Safety	Integrity	Availability	Reliability	Maintain-ability
Approach Target Directly/ Stop in Front of Obstacle	High	Low	Low	Low	Medium
Approach Target Directly/ Surround Obstables	High	Medium	Medium	Medium	Medium
Plan Path/Stop in Front of Obstacles	High	Medium	Low	Medium	Medium
Plan Path/Modify Path	High	High	High	High	High

Figure 2. *Dependability of different motion-planning and obstacle-avoidance systems.*

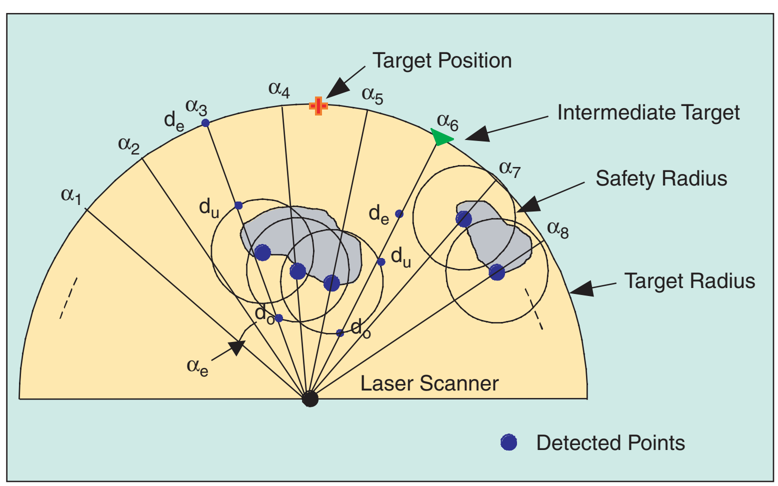


Figure 3. *Reactive obstacle avoidance using the "PolarBug" algorithm.*

In order to lead the robot around unexpected dynamic obstacles and smooth the planned path, a method for dynamic path modification is required. The elastic band algorithm [10]–[12] has been adapted for this purpose. An elastic band is a deformable collision-free path. Subjected to artificial forces, the elastic band deforms in real time to a short and smooth path that maintains clearance from all obstacles (Figure 5). The elastic band enables the robot to accommodate uncertainties and react to unexpected and moving obstacles. Given an initial planned path, the algorithm preserves the global nature of the path and the robot keeps close to it. Only when the path of the robot is blocked completely will the elastic band be broken and local path recalculation required.

All path planning and modification methods have been developed either for holonomic or nonholonomic robots [13]. Robot- or application-specific parameters, such as robot size and maximum turning angle, can be set in a configuration file.

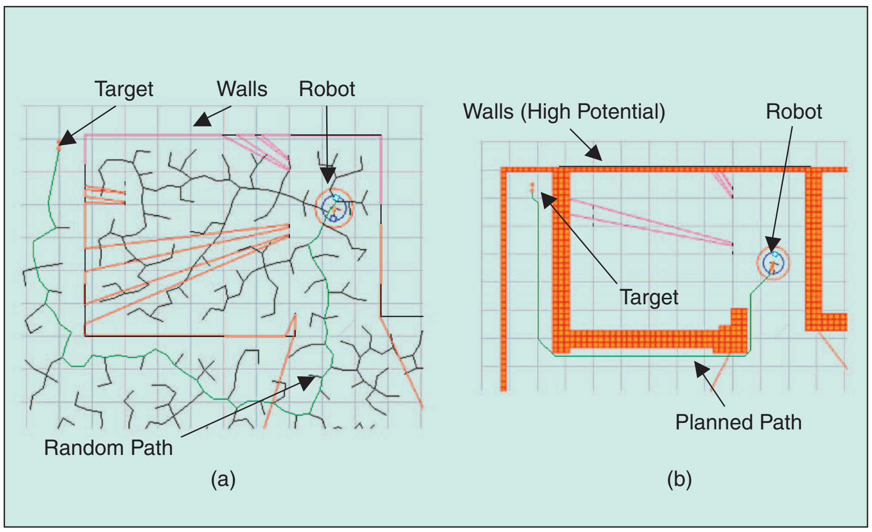


Figure 4. *(a) Path planning with random trees and (b) potential grids with wavefront expansion.*

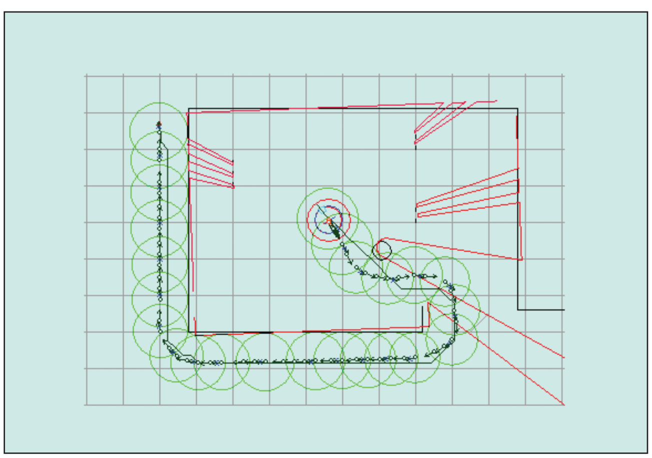


Figure 5. *Smooth path leading around static and dynamic obstacles.*

Dependable Execution of Fetch-and-Carry Tasks

Requirements for Dependable Fetch-and-Carry Task Execution

Unlike the dependable navigation issue that has mostly been solved, the dependable execution of fetch-and-carry tasks in a workspace shared by a human and a mobile robot with a manipulator is an extremely crucial task. The task of the robot is divided into a symbolic planning phase (compute single-execution steps to solve the given task) followed by an execution phase (a sequence of navigating and manipulating actions).

Task Execution

A robot assistant executing manipulation tasks in direct cooperation with humans will most likely operate in unstructured dynamic environments that are constantly modified by humans. A robot equipped with only on-board sensors is unable to supervise all of its operation environment all of the time. The position of objects might change without the robot noticing; therefore, its control architecture must be capable of coping with uncertainties and incomplete environment information in order to guarantee reliability and availability.

Manipulation

For a robot assistant interacting with humans, moving its manipulator along static trajectories is not sufficient. For dependable object manipulation in dynamic environments, environmental data must be acquired and considered throughout the whole manipulation process. An adequate level of maintainability and availability can only be given if all sensors are available on-board the mobile vehicle. Further considerations will therefore be restricted to mobile manipulators that use on-board sensing systems.

The degree of dependability for different manipulation methods can be classified similarly to motion planning and obstacle avoidance for a mobile platform (Figure 2). We can differ between simple collision avoidance (stop in front of an obstacle) and complex obstacle surrounding methods, as well as simple motion patterns (use shortest path) or planned motion (consider environment information). However, the complexity of the dependable manipulation problem is significantly higher because all calculations take place in multidimensional space: the manipulator acts in three-dimensional (3-D) space and collision detection and avoidance must not only be considered for the end effector but for each single axis.

The manipulation of objects can be classified by the number of objects to manipulate (including information on how to handle a specific object) and by the information that is available on the location of the objects (exact/approximate/no position information, Figure 6). The higher the complexity, the more difficult is it to obtain operation robustness.

Implementation of Fetch-and-Carry Task Execution

Task Execution

In contrast to the low-level modules (navigation, arm control, etc.) that are implemented in C++, the task execution module has been implemented in Python [14]. Python is a scripting language that provides interfaces for a wide range of operating systems and programming languages. It can be used to easily implement homogeneous interfaces for different applications. Python additionally supports network communication. By that, it is possible to run the main execution program on one computer and the different control modules on separate remote computers connected by Ethernet.

The execution module is implemented as a framework that enables the creation of parallel or sequential—as well as cyclic or discrete—activities (Figure 7). Activities can create and call other activities hierarchically, and any number of arguments can be transferred.

Special mechanisms for the synchronization of parallel activities have been implemented (Figure 8). An activity is able and wait for the end of one, multiple, finite, or cyclical activities. When waiting for multiple activities, the end of the first activity in the given list of activities will be indicated. When waiting for a cyclic activity, the end of each execution cycle will be indicated. Additionally, cyclic activities can indicate that a given condition has been fulfilled.

Furthermore, the following methods have been implemented:

- *stop*-command: exit given activity and clean up
- *sleep*-command: pause given activity. It can still be stopped by stop-command.
- *getState*-command: read state of given activity: running, ended, stopping, stopped, timeout, or failed
- *returnValue*-command: read last return value of given activity.

The execution framework uses exception handling methods provided by Python. Once an error has occurred, it is

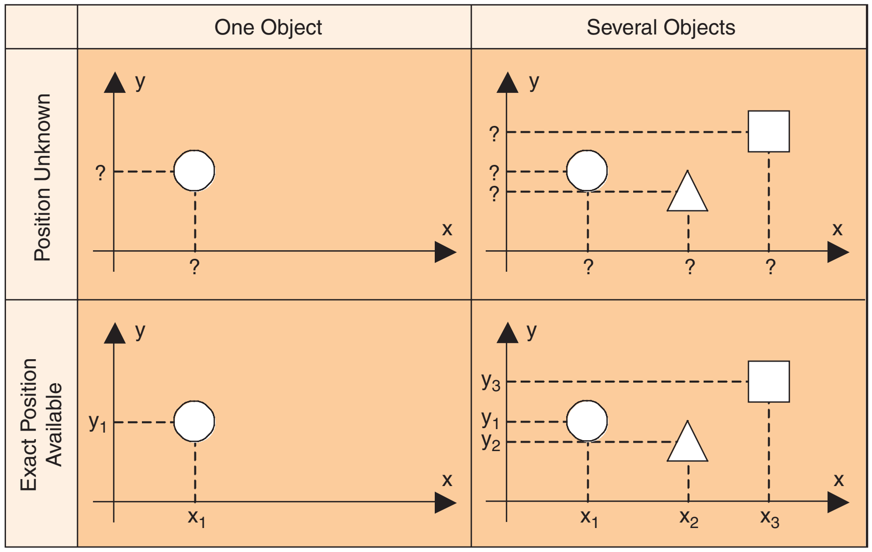


Figure 6. *Complexity of different manipulation tasks.*

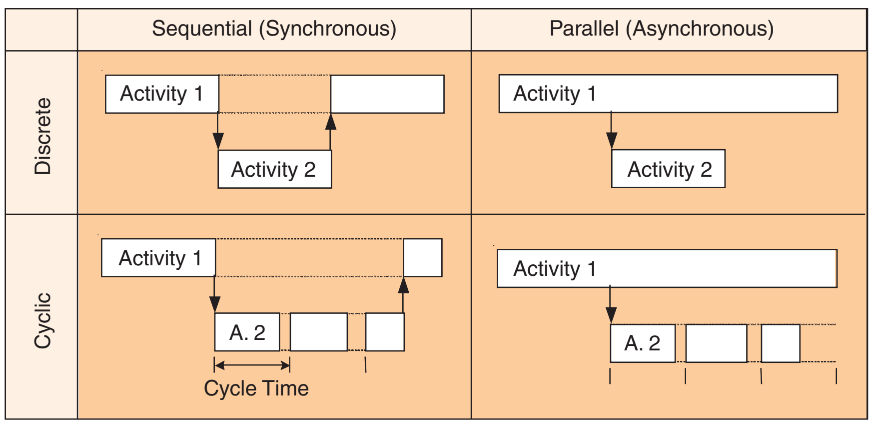


Figure 7. *Types of activities supported from the task execution module.*

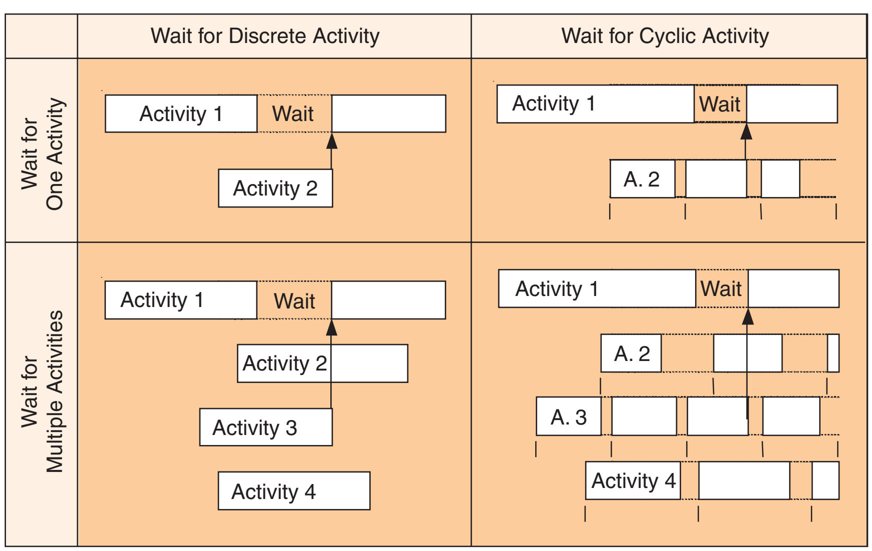


Figure 8. *Synchronizing parallel activities.*

caught in the calling activity, and a method to solve the problem will be executed. The error will be transferred hierarchically until an activity can solve the problem.

Grasping Objects from Flexible Positions

A new method for grasping different objects autonomously, based on camera and 3-D laser scanner information, has been implemented. With a camera, objects are initially taught to the robot. In order to grasp these objects, the corresponding reference image is loaded and compared to the current image of the camera. Object detection is based on object detection modules provided in the Matrox imaging library [15]. By fusing the data from the camera-based object detection and the 3-D distance information provided by the scanner, the exact location of the object is computed. This involves compensating inaccuracies of the camera and scanner position. The process of detecting a bottle is displayed in Figure 9.

Based on the laser scanner data, a collision-free trajectory for moving the end effector of the manipulator to the detected object without collisions is computed [16]. The implemented method is based on path planning with rapidly exploring random trees [8] and smoothing the calculated path. Figure 10 shows the calculated path for the scene of Figure 9. Although the manipulator its mounted on the right-hand side of the robot, the end effector approximates the object from the front in order to avoid collisions with the other objects detected in the scan.

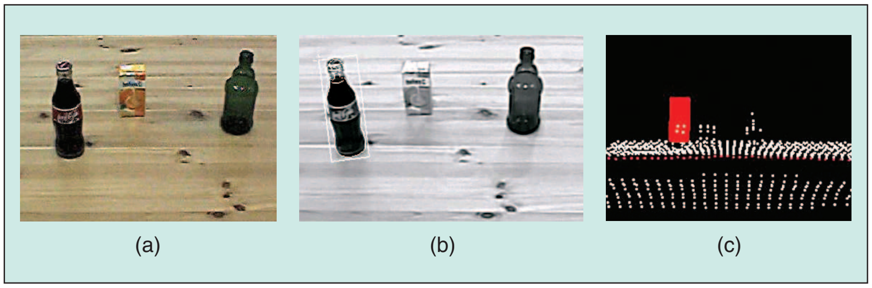


Figure 9. *Raw camera image, detected bottle, object to grasp in the 3-D scan.*

A method to locally adapt the gripper position precisely to the location of an object has further been implemented. The gripper is equipped with several distance sensors. An object is being approached and the position of the arm adapted until the sensors indicate the correct distance and position for grasping.

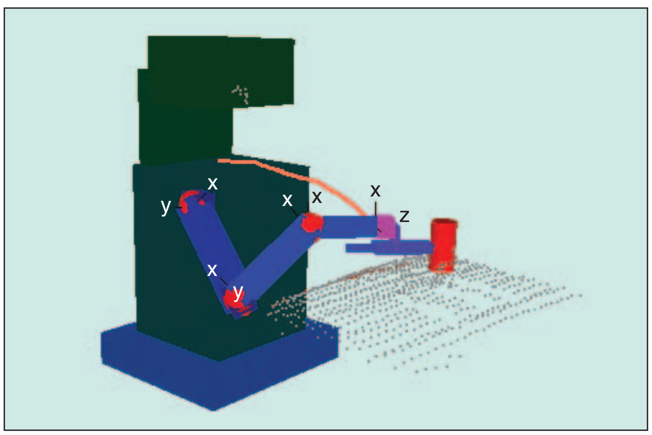


Figure 10. *Planning a collision-free path for grasping an object.*

Figure 11. *Entertainment robots in the Museum für Kommunikation, Berlin.*

Experimental Results and Verification

Museum für Kommunikation Berlin

In order to entertain visitors in the Museum für Kommunikation Berlin with a new technical attraction, three mobile robots (Figure 11) [17]–[19]—built and programmed by Fraunhofer IPA—were installed in the museum in March 2000. This is to our knowledge the first and only occasion where commercial entertainment robots have been installed for permanent operation, and, therefore, represents a milestone in mobile robotics history.

Robot Hardware

The hardware architecture of the museum robots is shown in Figure 12. The core of each robot is an industrial PC running on Windows NT. Drives and sensors are connected through serial ports or by CAN bus. Each vehicle is equipped with two driven wheels (differential drive), including shaft encoders for motion tracking. The robots are able to move at a speed of up to 1.2 m/s. Four castor wheels are further used for keeping the robots upright. A gyroscope is integrated in the robot platforms to track their current orientation. Two of the museum robots are additionally using an upper axis to rotate or tilt their head. A 2-D laser scanner is attached to the front of each robot. Its range is 180°, and its scan distance reaches up to 15 m. It provides a scan approximately every 30 ms at about 15 cm above the ground with a resolution of 0.5°.

Safety systems include a rubber bumper around the vehicles. Activating the bumper results in an immediate stop (power supply for all driven axes is cut). The operation speed of the robot is initially restricted depending on the size of the bumper, so that the robot can always stop before the bumper is crushed completely. In order to secure the area above the laser scanner, several infrared sensors have been integrated in the bumper facing upwards. Third, each robot is equipped with magnetic sensors facing to the ground. They are used as a secondary system to ensure that no robot ever leaves its operation area. In the unlikely case of a software failure, by leaving the given operation area and, therefore, crossing a magnetic band lowered in the ground, an emergency stop will be activated. In addition, each robot is equipped with two emergency stop buttons to deactivate the robot manually.

Control Software

The robots drive around autonomously in a big hall in the museum (Figure 13) and interact with the visitors. Each robot has a specific character, expressed through its look and appearance (driving speed, voice, etc.). The robots also differ in the way they interact with the museum visitors. The "Inciting" (KOMM-REIN) acts as an entertainer. His main concern is to welcome as many visitors as possible. The "Instructive" (ALSO-GUT) is a little introvert. She gives a guided tour of the museum and its history. The "Twiddling" (MACH-WAS) represents a child in its actions and expressions. It plays with a ball and does not bother to do anything serious.

The motion control of the Instructive is based on program-controlled navigation: the mobile vehicle command language (MVCL) allows the writing of operation programs as simple ASCII files. Operation programs provide the possibility to easily synchronize motion, multimedia, and upper-axis control commands. The other two robots use reactive navigation: the target position is constantly recalculated in reaction to their environment. Selected objects of a given shape are detected with the laser scanner (e.g., the ball or museum visitors). The robots then drive to a computed intercepting position. Due to the simple nature of the environment, targets are always approached directly without path planning. For obstacle avoidance, the museum robots use the PolarBug method as described previously and slow down their speed when getting close to obstacles. Localization uses local segment features (straight wall segments) and a heuristic scheme for matching and position estimation. If a robot loses its localization, a recovery method (drive to a corner and try to localize again) has been implemented.

Before the robots were set into operation, they were tested in the museum for two months. Due to the extensive tests performed during this time, the robots' software was thoroughly debugged and ran without any trouble from the start. Some problems occurred only in the first year of operation. The only software difficulty was caused by the multimedia player on one robot, which kept saving log files on the hard disc drive until it was full. This problem has been solved by deleting all temporary files regularly. More difficulties occurred due to the malfunction of hardware parts. A manufacturing fault of the motors required them to be replaced in all three robots. After months of daily operation, a shaft/grain connection became loose on the ball-playing robot. This incident occurred in this particular robot because it accelerates

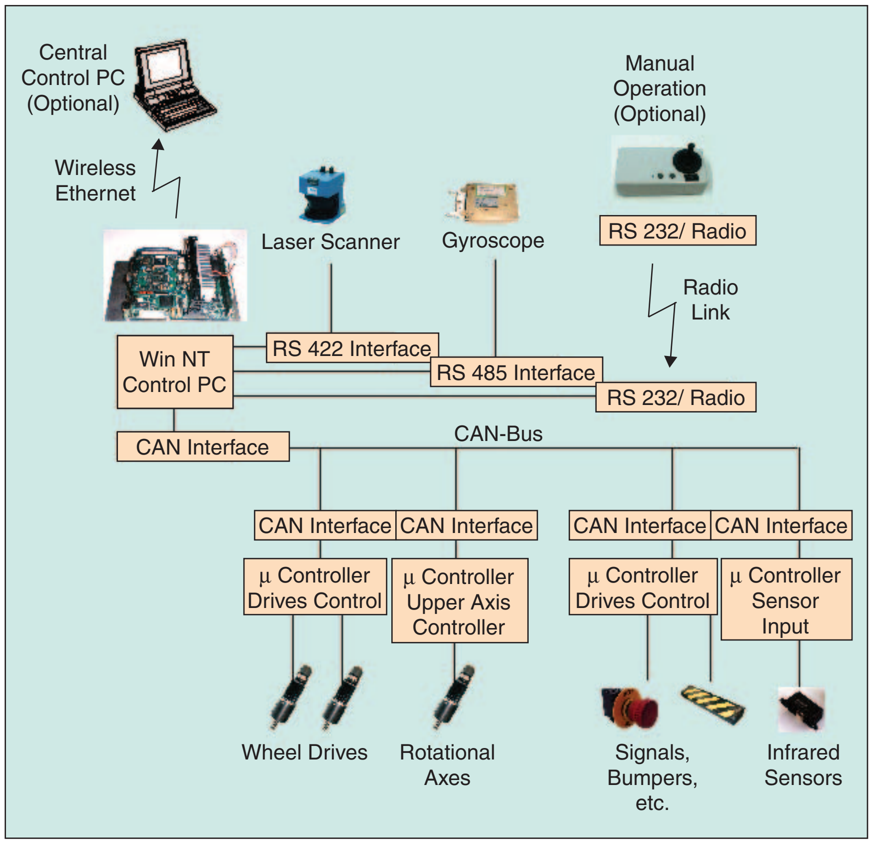


Figure 12. *Hardware architecture of museum robots.*

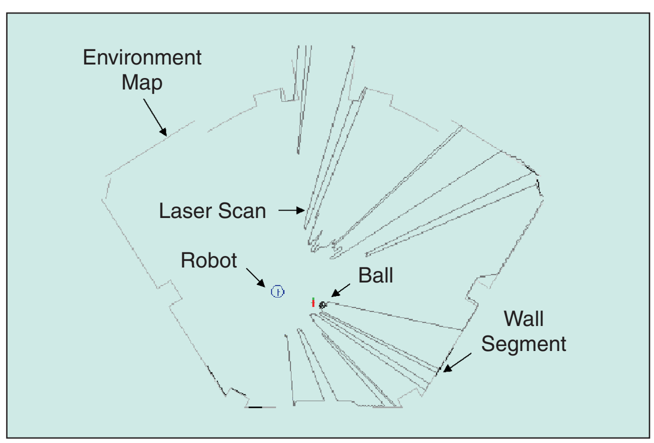


Figure 13. *Environment map of the Museum für Kommunikation, Berlin.*

and decelerates most frequently. The affected connection was modified on all three robots.

By the end of April 2004, after more than four years (more than 1,250 days) of operation, the three museum robots have accumulated approximately

- 10,000 h of operation
- 12,700 km travel distance.

During all this time, no collisions with either visitors or inventory of the museum was reported. In spite of the huge numbers of visitors that populated the operation area of the robots every day, they never left their operating area and never lost their localization. Thus, at no time did the robots present any danger to the visitors of the museum. They fulfilled their assigned tasks dependably.

Regular maintenance is done once a year. Worn parts, like batteries and worn-down wheels, are exchanged if necessary. Apart from that, no direct contact to the museum or the operating personnel is required. Due to the simple intuitive user interface, even inexperienced personnel can operate the robots without difficulty.

After the initial hardware problems mentioned above, no major faults were reported. Smaller difficulties are solved by the museum personnel themselves (e.g., restart robot if it does not work correctly), so it is impossible for us to give precise numbers on failures or fault-free operation time. However, in the last year we actually have not been contacted for assistance by the museum personnel even once, so obviously the robots run very dependably and no major problems have occurred.

The robots have been well accepted by the visitors of the museum. Children especially like the ball-playing robot. Even children of about three years of age enjoy playing with the robot, which at 1.2 m tall is substantially taller than them. This proves that an intuitive interaction with the robots was achieved by IPA's implementation.

An inconvenient observation has been made concerning the way visitors of the museum are using the emergency buttons: They tend to press the emergency buttons of the robots for fun. After an emergency stop button has been pressed, a member of the museum staff has to put the robot back in operation since a key is needed to release the emergency stop. Due to safety regulations, the staff members could not be relieved from this duty.

FESTO Anniversary Exhibition

Path planning based on a geometric planner and dynamic path modification based on the elastic band method were first proved during a two-week installation of Fraunhofer IPA's Care-O-bot at the anniversary exhibition of the company FESTO in October/November 2001 (Figure 14). Care-O-bot is a mobile service robot originally designed to assist people in their private homes. The first prototype, a mobile platform with moveable touch screen, was built in 1997 [20]. The hardware of Care-O-bot I is identical to that of the museum robots.

Care-O-bot drove around the exhibition area autonomously. When visitors were detected, the robot approached them, turned its touch screen toward them, and offered to guide them to a specific exposition area. The target could be selected by touching an image displayed on the touch screen of the robot. Due to the complexity of the environment, which was also sometimes crowded by visitors, Care-O-bot often had to decide between different paths and modify a planned path in order to surround dynamic obstacles.

Care-O-bot fulfilled its task to our satisfaction. It never compromised the safety of the visitors. It always found a feasible path to its target quickly (below 1 s), unless all possible paths were blocked by visitors. In that case, visitors were asked to move out of the way by speech output. After they had moved, Care-O-bot created a new optimal path based on updated sensor information and continued its way.

Minor difficulties occurred due to malfunctions of a motor controller and the touch screen controller. However, these difficulties were fixed quickly and considered for all future robot developments. Based on these and other experiences, the latest models available from Neobotix [21]—as, e.g., Care-O-bot II and the Opel robots—use different and more efficient hardware components than Care-O-bot I and the museum robots.

Care-O-bot II

Care-O-bot II, built in 2002, is a mobile service robot

Figure 14. *Care-O-bot I at the FESTO anniversary exhibition.*

Figure 15. *Care-O-bot II in sample home environment.*

equipped with a manipulator arm, adjustable walking supporters, a tilting sensor head containing two cameras and a laser scanner, and a hand-held control panel. For a complete hardware description see [22]. The manipulator arm—developed specifically for mobile service robots—provides the possibility of handling typical objects in a home environment. The flexible gripper attached to the manipulator is suitable for grasping different objects such as mugs, plates, and bottles. The hand-held control panel can be used for instructing and supervising the robot.

In order to demonstrate the fetch-and-carry capabilities of Care-O-bot II, a sample home environment at the size of 5 × 5 m, containing a sofa, coffee table, and two bookshelves has been set up (Figure 15). The execution of fetch-and-carry tasks was first presented successfully to international scientists and visitors at the Human Computer Interaction Status Conference, Berlin, Germany, 3–4 June 2003. Since then, presentations of the robot at SPS/IPC/DRIVES fair in Nurnberg, Germany, in November 2003, at the CeBIT fair in Hannover, Germany, in March 2003, and several additional presentations of the robot in our lab have taken place.

Several objects of different types (e.g., a bottle, juice pack, or chips box) can be placed in random positions in the different locations. A user can sit on the sofa and command Care-O-bot II through its hand-held control panel (touch screen or speech input). The control panel is connected to the two control computers of Care-O-bot II by a wireless local area network (LAN). Two commands have been implemented: "bring to" and "tidy up." The first command is used to order the robot to bring a specific object to a selected position, whereas the second command orders the robot to clean up all objects located at the selected location. The commands are used as goal statements for a symbolic planner, which generates a list of actions. Whenever a solution to the problem exists, the symbolic planner generates a suitable list of actions to fulfill the given task. The list is executed step by step from the execution framework mentioned above. For a complete description of the system architecture concept, see [23].

In the majority of cases, the first action is a move command to go to an object's location. Therefore, Care-O-bot II is equipped with the latest version of our navigation software, including localization with multiple hypotheses, path planning, and elastic-band path modification and is able to navigate within the specified environment without any difficulties.

For picking up objects, the previously described method for grasping objects from flexible positions is used. First, the object is detected by processing the camera image. This is a delicate point during task execution: due to changing lighting conditions, objects sometimes are not detected correctly. If the number of identified objects does not match the number of expected objects provided by the internal symbolic world model, an error is detected. To recover, the current picture is sent to the control panel, and the user is asked to indicate the object he wants to get grasped. During our tests, this problem occurred during approximately one out of five trials and could be solved through user interaction every time. After having detected the object that should be grasped, task execution continues normally with 3-D scanning, matching of camera-image and 3-D data, and calculating a collision-free trajectory. If the alignment of the sensor head with the manipulator is disturbed, small inconsistencies occur when matching the camera and laser scanner information. These inconsistencies are compensated completely by the sensor-based grasping module.

The calculation of valid, collision-free trajectories is the second and last delicate point during task execution. Although the algorithm is very effective, mechanical constraints complicate the search for a valid trajectory. The manipulator has 6° of freedom with altering pivot joints and revolting joints, starting with a pivot joint at axis one. Every axis has limit switches at ±165°. Therefore, the work space of the manipulator has the shape of a vertical torus. Furthermore, the movement of the arm is restricted in order to avoid self-collisions with the body, head, and manipulator of the robot. Because of these constraints of the specific kinematics of Care-O-bot II, low positions like the coffee table can be reached more easily than higher positions like the bookshelves.

The position error of the mobile platform when driving to a target lies within ±5 cm and ±2 °cm. Depending on the final position of the platform, the workspace of the manipulator covers different areas of the furniture surfaces. In bad situations, the object to grasp is outside this area, which results in a timeout failure of the trajectory planner. This error causes a break in the task execution. It could be recovered by correcting the platform position, which, however, has not been implemented yet. During demonstrations, such situations are avoided by allowing only one object on furniture with high surfaces. Whatever position error the platform has, the working area of the arm covers the one object, and the object can be grasped every time without any difficulties.

When placing objects on a table or bookshelf, the location is first analyzed with the 3-D laser scanner. Once a position free of other objects has been detected, a collision-free path is planned, and the arm moves to this position. Data from the force-torque sensor is used to detect the point where the object has come in contact with the table. The gripper is then opened and the object relieved.

Figure 16. *Care-O-bot II serving a drink.*

Figure 17. Opel navigators OSKAR and MONA.

The location "sofa" indicates the direct interaction between user and robot: the object is handed over to or taken from the user (Figure 16). The direct interaction between user and robot is accomplished by using the force-torque sensor to detect the touch of the user. When handing over an object to the robot, the gripper sensors are used to detect the object and to close the gripper.

In order to enable the user to supervise the execution of a task by the robot, the camera picture of the robot is constantly transferred to the hand-held control panel. Furthermore, the currently executed action is indicated so that the user is always in control of the machine and, if necessary, can modify or interrupt an action. In addition, the robot describes its current action by speech output.

The conference presentations of Care-O-bot as well as several presentations of the robot in our lab have proven the dependability of the task-planning and execution system: the execution framework ran to our complete satisfaction. Error detection and error recovery of the framework as mentioned above enables the detection and grasping of objects to function without any failures. If additional failures occur, they are caused by hardware problems that prevent the system from initializing correctly. Low batteries, wireless LAN that does not connect, gripper controllers or laser scanners that do not initialize, or an emergency button that has not been reset are typical failures during start-up. But as soon as the system runs, Care-O-bot II fetches and carries objects without failures from any one location to another.

The safety of the illustrated manipulation method, however, is still low and not suitable for everyday use—especially for inexperienced users. The trajectory for the robot arm is planned only once. Components to supervise the environment while moving the arm—which is absolutely necessary in order to react to dynamic obstacles—are not implemented, yet. Due to the high processing power and time required for taking the 3-D laser scanner (tilting the head takes nearly 1 s) and the scan evaluation, this proves to be extremely difficult. Furthermore, apart from standard emergency stop buttons, no backup safety system has been implemented—yet.

Summary and Outlook

The dependability of mobile robot navigation and execution of fetch and carry tasks with a mobile robot with a manipulator have been analyzed. For dependable navigation, the requirements for localization, motion planning, and obstacle avoidance systems have been analyzed. A dependable localization system and two methods for dependable obstacle avoidance implemented at Fraunhofer IPA have been presented. The successful operation of three entertainment robots in the Museum für Kommunikation in Berlin for nearly three and a half years proves that the navigation system of these robots is absolutely dependable and suited for everyday use. The successful operation of Care-O-bot I at the anniversary exhibition of FESTO shows that dependable navigation capabilities in complex environments, requiring path planning and dynamic path modification, have already been obtained.

Two new entertainment robots (Figure 17), OSKAR (*Opel Systemkreation aus Rüsselsheim*) and MONA (*Multifunktionale Opel Navigatorin*) have been developed for Opel in Berlin, starting daily operation in October 2003. Based on our experiences with the museum robots in Berlin, the hardware and control software for this new generation of entertainment robots has been improved considerably. Due to the high complexity of the operation environment, an improved localization method based on multiple hypotheses has been implemented. The robots lead visitors to specific exposition areas; therefore, a path planner based on potential fields and elastic-band path modification for dynamic obstacle avoidance have been integrated.

Before they were set in operation, the robots were installed in their environment and tested for two weeks. During this time, the robots operated reliably and safely. The multihypotheses localization enabled the robots to recover position loss even in difficult situations, and the improved path planner, elastic band, and motion controller ensured the reliable and smooth motion of the vehicles. The long-term operation of the robots will provide additional information about the dependability and suitability for everyday use of our new hardware and software modules.

Unlike the dependable navigation issue, the dependable execution of fetch-and-carry tasks in a workspace shared by a human and a mobile robot with a manipulator is an extremely crucial task and target for ongoing research. The requirements for dependable execution of fetch-and-carry tasks, which involves dependable manipulation as well as dependable task planning and execution, have been discussed. An execution framework for dependable task planning and execution has been described. These modules were tested by executing fetch-and-carry tasks in a sample home environment with Care-O-bot II, a mobile robot assistant equipped with manipulator arm. A new method for object detection and trajectory planning for a manipulator arm based on vision and 3-D laser scanner data has been presented.

Whereas the dependability of the task-planning and execution module as well as the reliability of the manipulation method could be proved, safety issues for manipulating objects among and in direct contact with humans are still an open problem. Therefore, future work will be concerned with developing new safety sensors for real-time collision avoidance of mobile robot manipulators and discussing new safety regulations that enable the application of such robots in everyday environments.

Acknowledgments

Part of this work has been supported by the MORPHA-project [24], funded by the German Ministry of Education and Research (bmb+f) under grant 01IL902G/9.

References

[1] R. Siegwart, K.O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, and M. Meisser, "Robox at expo.02: A large scale installation of personal robots," in *Robotics and Autonomous Syst.,* vol. 42, no. 3, pp. 203–222, 2003.

[2] K. Lay, E. Prassler, R. Dillmann, G. Grunwald, M. Hägele, G. Lawitzky, A. Stopp, and W. von Seelen, "MORPHA: Communication and interaction with intelligent, anthropomorphic robot assistants," in Bundesminister für Bildung und Forschung/Projektträgerschaft Informationstechnik (Lead Projects Human-Computer-Interaction: International Status Conference), Saarbruecken, Germany, Oct. 26-27, 2001. Berlin, 2001, pp. 67-77.

[3] A. Avizienis, J.-C. Laprie, and B. Randell, "Fundamental concepts of computer system dependability," in *Proc. IARP/IEEE-RAS Workshop on Robot Dependability*, Seoul, Korea, 2001, pp. I-1.

[4] C.-W. Lee, Z. Bien, G. Giralt, P. Corke, and M. Kim, "Technical challenge for dependable robots in home environments," in *Rep. 1st IARP/IEEE-RAS Joint Workshop*, Seoul, Korea, 2001. Available: http://www.laas.fr/rdhe/ Post_WS_report.pdf

[5] P. Corke, "Safety of advanced robots in human environments," in *A Discussion Paper for IARP*, 1999. Available: http://www.international-robotics.org/

[6] V.J. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 5, pp. 1058–1068, 1990.

[7] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[8] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Computer Science, Iowa State Univ., Tech. Rep. No. 98-11, 1998.

[9] K. Jiang, L.D. Seneviratne, and S.W.E Earles, "A shortest path based path planning algorithm for nonholonomic mobile robots," *J. Intell. Robotic Syst.*, vol. 24, pp. 347–366, 1999.

[10] S. Quinlan and O. Khatib, "Elastics bands: Connecting path planning and control," in *IEEE Trans. Robot. Automat.*, 1993.

[11] M. Khatib, H. Jaouni, R. Chatila, and J.P. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," in *IEEE Int. Conf. Robotics and Automat.*, Albuquerque, NM, 1997, pp. 2920–2925.

[12] H. Jaouni, M. Khatib, and J.P. Laumond, "Elastic bands for nonholonomic car-like robots: Algorithms and combinatorial issues," in *3rd Int.l Workshop on the Algorithmic Foundations of Robotics (WAFR'98)*, Houston, TX, 1998.

[13] B. Graf and J.M.H. Wandosell, "Flexible path planning for nonholonomic mobile robots," in *Proc. 4th European Workshop on Advanced Mobile Robots (EUROBOT'01)*, Lund, Sweden, Sept. 19–21, 2001, pp. 199–206.

[14] Python Language Homepage. http://www.python.org.

[15] Matrox Imaging Library Homepage: http://www.matrox.com/imaging/products/mil/home.cfm.

[16] B. Rohrmoser and C. Parlitz, "Implementation of a path-planning algorithm for a robot arm," in *Proc. of Robotik,* 2002, Ludwigsburg, Germany, pp. 59-64.

[17] B. Graf, R.D. Schraft and J. Neugebauer, "A mobile robot platform for assistance and entertainment," in *Proc. ISR-2000*, Montreal, Canada, pp. 252–253.

[18] B. Graf and O. Barth, "Entertainment robotics: Examples, key technologies and perspectives," in *Proc. IROS 2002-Workshop: Robots in Exhibitions*.

[19] R.D. Schraft, B. Graf, A. Traub, and D. John, "A mobile robot platform for assistance and entertainment," in *Ind. Robot J.*, vol. 28, pp. 29–34, 2001.

[20] R.D. Schraft, C. Schaeffer, and T. May, "The Concept of a system for assisting elderly or disabled persons in home environments," in *Proc. 24th IEEE IECON*, Aachen, Germany, 1998.

[21] Neobotix Homepage http://www.neobotix.de, 2003.

[22] M. Hans, B. Graf, and R.D. Schraft, "Robotic home assistant care-O-bot: Past-Present-Future," in *Proc. ROMAN-2001*, Berlin, Germany, pp. 407–411.

[23] M. Hans and W. Baum, "Concept of a Hybrid Architecture for Care-O-bot." in *Proc. IEEE Int. Workshop Robot and Human Interactive Communication* (RO-MAN), 2002, Bordeaux, Paris, France, pp. 407-411.

[24] "MORPHA—Interaction, communication, and cooperation between humans and robot assistants." Available: http://www.morpha.de.

Birgit Graf is working as a research assistant at the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) in Stuttgart, Germany. After concluding her Diploma thesis on "Robot Soccer" at the University of Western Australia, Perth, she received a Diploma in Informatics from Stuttgart University in 1999. Since then she has been involved in several mobile robot projects, for example in the "Museum für Kommunikation Berlin," where in March 2000 three mobile platforms have been installed to autonomously move among the visitors, communicate to, and interact with them. For this installation, together with her colleagues, she was honored with the Fraunhofer award 2000, a research award given away annually for outstanding scientific work within the Fraunhofer society. Her main research interest lies in the development of mobile service and entertainment robots.

Matthias Hans is working as a research assistant at the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) in Stuttgart, Germany. He received a Diploma in Electrical Engineering from Kaiserslautern University in 1997. He manages the MORPHA project at Fraunhofer IPA and coordinated the development of the second Care-O-bot prototype. His main research interest lies in the development of the system architecture and interactive task planner.

Rolf Dieter Schraft has been director of the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) in Stuttgart, Germany, since 1993. He graduated from Stuttgart University in 1969. He finished his Ph.D. thesis in 1976 and is lecturing at the Universities of Stuttgart and Dortmund. In 1986 he received the "Golden Robot Award" in 1990 the JIRA Award and in 1992 the "Josef F. Engelberger Award." He is member of the executive committee of the International Federation of Robotics and was chairman in 1998–1999.

***Address for Correspondence*:** Birgit Graf, Fraunhofer Institute Manufacturing Engineering and Automation (IPA), Nobelstrasse 12, 70569 Stuttgart, Germany. E-mail: birgit.graf@ipa.fhg.de. URL: http://www.care-o-bot.de.