

# Recent Advances in the Development of the Humanoid Service Robot *HERMES*

**Rainer Bischoff**

Bundeswehr University Munich  
The Institute of Measurement Science  
85577 Neubiberg, Germany  
e-mail: Rainer.Bischoff@unibw-muenchen.de  
URL: <http://www.rz.unibw-muenchen.de/~l61brai>

## Abstract

The humanoid service robot *HERMES*, first introduced as a concept at the First International Field and Service Robotics Conference FSR'97 in Canberra, was actually built and several times enhanced in the meantime. It is constructed from 25 motor modules with identical mechanical and electrical interfaces, thus yielding a very flexible, extensible and modular design that can be easily modified. With its omnidirectional wheel base, body, head, eyes and two arms it has now 22 degrees of freedom and resembles a human in height and shape. Its main exteroceptive sensor modality is stereo vision. Both camera „eyes“ can be actively and independently controlled in pan and tilt degrees of freedom. A variety of proprioceptive sensors further enhances its perceptual abilities. *HERMES* is used as an experimental platform to advance research in human-friendly man-machine interaction, mobile manipulation, environmental exploration and navigation. To study autonomous human-like cooperative behavior a robot operating system has been developed that allows an easy implementation, coordination and execution of various skills leading to several goal-directed and elaborate robot behaviors. In addition, *HERMES* can be teleoperated via keyboard or voice in all its degrees of freedom, making it ideally suited for entertaining people with a variety of pre-programmed motion sequences. Experiences gained in the development of the robot and in initial experiments designed to allow an assessment of the overall system performance as well as implementation details will be reported.

## 1 Introduction

Although the vast majority of robots today are used in factories, advances in technology are enabling robots to automate many tasks in non-manufacturing industries such as agriculture, construction, health care, retailing and other services. These so-called “field and service robots” aim at the fast growing service sector and promise to be a key product for the next decades [Schraft, Schmierer 1998].

A first generation of service robots already assists or rationalizes manipulation, transportation and processing tasks in

various service domains. Possible application areas include cleaning, transport and handling of goods, surveillance and protection of buildings, construction and maintenance. All robots working in these domains have been designed to provide special solutions for specific problems. They rely on detailed teaching and programming and carefully prepared environments. It is costly to maintain them and it is difficult to adapt their programming to even slightly changed environmental conditions or modified tasks.

To develop the next generation of service robots that could be used in many different environments (domestic, public and industrial) for a variety of tasks (e.g., elderly care, helping handicapped people, assistance in factories) is a challenging task. Much research is still needed to improve considerably design and safety concepts, locomotion and manipulation capabilities, cooperation and communication abilities, reliability, and – probably most importantly – adaptability, learning capabilities and sensing skills. To advance research in these fields we have developed the experimental robot *HERMES* based on our previously gained experiences [Bischoff, Graefe 1998].

In the sequel, we present *HERMES* as it has been designed and realized and give an overview of its control architecture. Examples of skillful behaviors that provide solutions to some of the above-mentioned challenges are described in more detail in section 3. The actual overall system performance can be assessed based on real-world experiments which are described in section 4. Finally, section 5 provides conclusions.

## 2 The Humanoid Service Robot *HERMES*

### 2.1 Design and Realization of *HERMES*

In designing our humanoid experimental robot we placed great emphasis on modularity and extensibility [Bischoff 1997]. All drives are realized as modules with compatible mechanical and electrical interfaces; each drive module consists of two cubes rotating relative to each other and containing a motor-transmission combination, power electronics, sensors, a micro-controller, and a communication interface. A standardized CAN bus connects all drive mod-

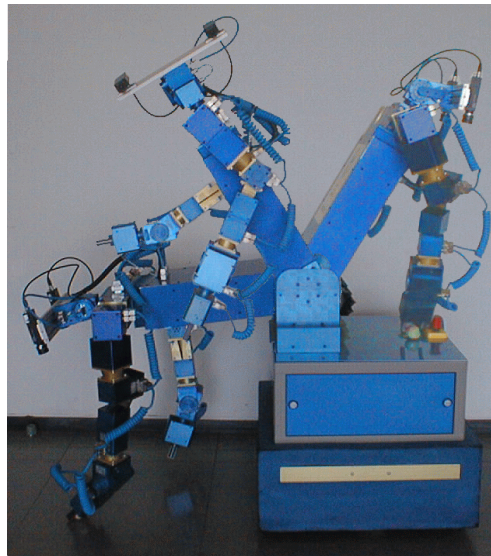
ules with the main computer. *HERMES* runs on 4 wheels, arranged on the centers of the sides of its base. The front and rear wheels are driven and actively steered, the lateral wheels are passive.

The manipulator system consists of two articulated arms with 6 degrees of freedom each on a body that can bend forward ( $130^\circ$ ) and backward ( $-90^\circ$ ). The work space extends up to 120 cm in front of the robot. The heavy base guarantees that the robot will not lose its balance even when the body and the arms are fully extended to the front. Currently each arm is equipped with a two-finger gripper that is sufficient for basic manipulation experiments (Figure 1).

Main sensors are two video cameras mounted on independent pan/tilt drive units in addition to the pan/tilt unit that controls the common „head“ platform. The cameras can be moved very fast with accelerations up to  $10.000 \text{ }^\circ/\text{s}^2$  and velocities up to  $800 \text{ }^\circ/\text{s}$ . The common pan/tilt unit achieves accelerations of  $860 \text{ }^\circ/\text{s}^2$  and velocities of  $215 \text{ }^\circ/\text{s}$  to Numerous proprioceptors such as angle encoders, current converters and temperature sensors are integrated in the motor modules, additional sensors may be connected via available interfaces. A radio Ethernet interface allows to control the robot remotely. A wireless keyboard can be used to teleoperate the robot up to distances of 7 m. Separate batteries for the motors and the information processing system allow a continuous operation of the robot for several hours without recharging.

## 2.2 Behavior-Based Control of a Humanoid Robot

Seamless integration of many – partly redundant – degrees of freedom and various sensor modalities in a complex robot calls for a unifying approach. We have developed a system architecture that allows integration of multiple sensor modalities and numerous actuators, as well as knowledge bases and a human-friendly interface. In its core, the system is behavior-based, which is now generally accepted as an efficient basis for autonomous robots [Arkin 1998]. However, to be able to select behaviors intelligently and to pursue long-term goals in addition to purely reactive behaviors, we have introduced a situation-oriented deliberative component that is responsible for situation assessment and behavior selection.



**Figure 1:** Motion sequence to illustrate the enlarged work space gained by a bendable body. The two arms have 6 degrees of freedom each.

### 2.2.1 System Overview

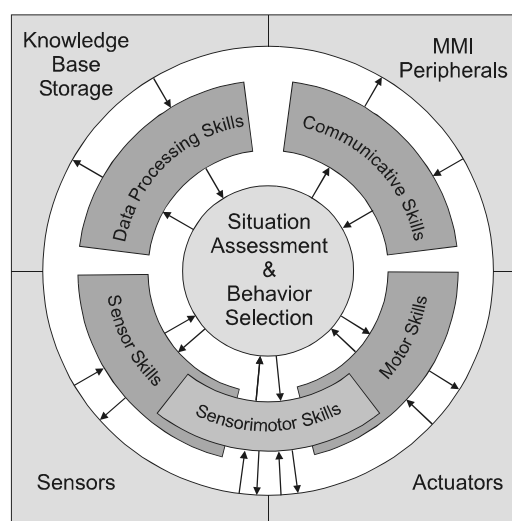
Figure 2 shows the essence of the situation-oriented behavior-based robot architecture as we have implemented it. The situation module (situation assessment & behavior selection) acts as the core of the whole system and is interfaced via “skills” in a bidirectional way with all other hardware components – sensors, actuators, knowledge base storage and MMI peripherals (man-machine and machine-machine interface peripherals).

These skills have direct access to the hardware components and, thus, actually realize behavior primitives. They obtain certain information, e.g., sensor readings, generate specific outputs, e.g., arm movements or speech, or plan a route based on map knowl-

edge. Skills report to the situation module via events and messages on a cyclic or interruptive basis to enable a continuous and timely situation update and error handling.

The situation module fuses via skills data and information from all system components to make situation assessment and behavior selection possible. Moreover, it provides general system management (cognitive skills). Therefore, it is responsible for planning an appropriate behavior sequence to reach a given goal, i.e., it has to coordinate and initialize the in-built skills. By activating and deactivating skills, a management process within the situation module realizes the situation-dependent concatenation of elementary skills that lead to complex and elaborate robot behavior.

In general, most skills involve the entire information processing system. However, at a gross level, they can be classified into five categories besides the cognitive skills: **Motor skills** are simple movements of the robot’s actuators. They can be arbitrarily combined to yield a basis for more complex control commands. Encapsulating the access to groups of actuators, that form robot parts, such as wheel-base, arms, body and head, leads to a simple interface structure, and allows an easy generation of pre-programmed motion patterns. **Sensor skills** encapsulate the access to one or more sensors, and provide the situation module with proprioceptive or exteroceptive data. **Sensorimotor skills** combine both sensor and motor skills to yield sensor-guided robot motions, e.g., vision-guided or tactile



**Figure 2:** System architecture of a personal robot based on the concepts of situation, behavior and skills.

and force/torque-guided motion skills. **Communicative skills** pre-process user input and generate a valuable feedback for the user according to the current situation and the given application scenario. The system's knowledge bases are organized and accessed via **data processing skills**. They return specific information upon request and add newly gained knowledge (e.g., map attributes) to the robot's data bases, or provide means of more complex data processing,

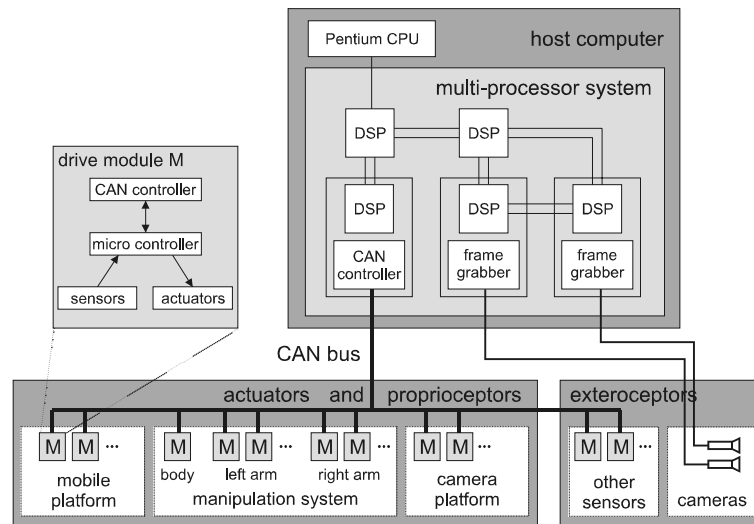
e.g., path planning. For a more profound theoretical discussion on our system architecture which bases upon the concepts of situation, behavior and skill see [Graefe, Bischoff 1997] and [Bischoff, Graefe 1999].

### 2.2.2 Implementation

A hierarchical multi-processor system is used for information processing and robot control. The control and monitoring of the individual drive modules is performed by the sensors and controllers embedded in each module. The main computer is a network of digital signal processors (DSP, TMS 320C40) embedded in a standard industrial PC. Sensor data processing (including vision), situation recognition, behavior selection and high-level motion control are performed by the DSPs, while the PC provides data storage and the human interface (Fig. 3).

A robot operating system has been developed that allows sending and receiving messages via different channels among the different processors and microcontrollers. All tasks and threads run asynchronously, but can be synchronized via messages or events. The left and the right cameras are synchronized via hardware, but software needs to make sure that images are digitized at the same time when stereo vision is to be performed. However, image acquisition can as well run asynchronously, e.g., to provide two different image capture rates for two independent image processing tasks.

Overall control is realized as a finite state machine capable of responding to prioritized interrupts and messages. After powering up the robot finds itself in the state "Waiting for next mission description". A mission description is provided as a text file that may be either loaded from a disk or received via e-mail or entered via keyboard. It consists of an arbitrary number of single commands or embedded mission descriptions that let the robot perform a required task. All commands are written in natural language and passed to a parser and an interpreter. If a command cannot be under-



**Figure 3:** Modular and adaptable hardware architecture for information processing and robot control.

stood, is under-specified or ambiguous the situation module tries to complement missing information from its situated knowledge or asks the user via its communicative skills to provide it.

In the current implementation commands have to be typed and the robot's responses are written to a display or sent via e-mail. Integration of speech recognition and speech output is well under way so that eventually both written and spoken instructions will be understood.

Motion skills are mostly implemented at the microcontroller level within the actuator modules. High-level motor skills, such as coordinated smooth arm movements are realized by a dedicated DSP interfaced to the microcontrollers via a CAN bus. Sensor skills are implemented on those DSPs that have direct access to digitized sensor data, especially digitized images.

## 3 Realization of Skillful Behavior

In this section we show how skillful goal-directed behavior can be achieved by combining simple motor and sensing skills to more complex sensorimotor skills, and describe how communicative and data processing skills are realized.

### 3.1 Motor Skills

Motor skills are simple, yet fundamental, movements of the robot's joints, e.g., moving a single module to a certain position or changing its current velocity. High-level motor skills provide access to groups of modules that form specific robot parts (e.g., wheelbase, arms, or head), and generate more complex motion patterns, e.g., they move the arms to a certain position relative to their actual position or set new velocities for all the modules at the same time. Moving an arm requires the definition of ramp parameters (end position, maximum velocity and acceleration) for each joint to reach a given end position. To generate smooth arm movements at each positioning request, it is furthermore required that all modules start and finish their movements at the same time. Therefore, the ramp parameters are individually computed for each module, considering the motion capabilities of the slowest one at a given moment. The governing DSP finally transmits them to the microcontrollers that actually provide accurate ramp control at a rate of 1 kHz.

### 3.2 Sensor Skills

Sensor skills encapsulate in their simplest form access to the proprioceptive (joint angles, motor currents, battery voltage,

etc.) or exteroceptive sensor data (visual, tactile, hearing, etc.).

### 3.2.1 Visual sensing

One of the most needed sensor skills is to detect objects in the robot's surroundings. Among the objects that a mobile robot needs to detect while navigating in an office environment are corridors, junctions, doors, work places (e.g., tables) and information signs (e.g., door plates). Obstacles need to be detected as well but since they may have arbitrary appearance in terms of shape, texture and rigidity, a method that would be suitable for the detection of all kinds of obstacles cannot be given. Instead, we make some basic assumptions about the appearance of the background when it is obstacle-free (see, e.g., [Horswill 1994]). Thus, by identifying these obstacle-free areas, the robot will automatically get hints about where obstacles, but as well objects of interest, might be located. Although we have to restrict our robot to working environments where the floor does neither have bright reflections nor shadows nor texture (big patterns), this method is very reliable, fast to compute (on single 2-D images) and rather conservative, preferring false positives to false negatives.

Robust segmentation is provided by a multilevel image processing algorithm that self-initializes to the floor color in front of the robot and adapts during operation, so that changes in brightness can be compensated to some extent. Built upon this sensor skill of segmenting the image and yielding a polygon of the contour, other skills have been established that enable the robot to detect and recognize, or at least to derive hypotheses about the presence of, objects in the scene relevant for navigation, e.g., junctions, doors and docking stations (e.g., tables shown in Figure 4, right). Upon object recognition, reference points and lines are identified (based on procedural and object data knowledge) and subsequently used for tracking.

### 3.2.2 Tactile sensing

To enable a robot to perceive objects by a sense of touch, in general, tactile sensors are required. Although we are working towards the development of highly integrated and high-resolution tactile sensors to be placed on the gripper surfaces and around the wheelbase, we have found other means of detecting touch events that are helpful in guiding human-



**Figure 4:** Gradient filtering along vertical search paths yields contour points that mark the transition between the floor and other objects. Left: typical corridor image with an open door as a possible obstacle. Right: two tables as possible docking objects.

robot or robot-environment interaction. By intelligently processing angle encoder values and motor currents we are able to provide kinesthetic sensing. Kinesthesia is a “sense mediated by end organs located in muscles, tendons, and joints and stimulated by bodily movements and tensions” [Babcock 1976]. Transferring kinesthetic sensing to the robot for detecting touch events means to detect tensions on the robot structure or torques at the joints that do not result from internal motion requests but most probably from external circumstances.

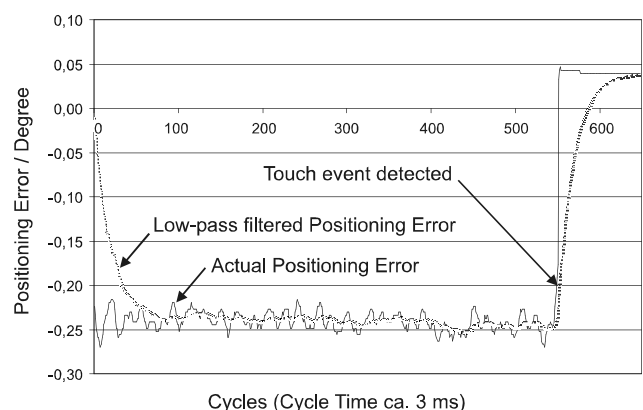
Two kinesthetic sensing skills have been developed: one, for detecting touch events or vibrations that occur on any part of the robot structure; two, for detecting unusual external forces during pre-defined robot motions that are, however, unknown to the sensing skill. While the first skill is being used to interact with people in order to shake hands and to hand over or take objects, the second skill allows to smoothly place grasped objects onto other objects. In both cases angle encoder values are sampled at a rate of 1 kHz and low-pass filtered to yield a prediction for the next cycle. If a new angle value deviates significantly from the predicted one, a touch event is signaled to the software module that has requested to detect this touch event (Figure 5).

## 3.3 Sensorimotor Skills

Combining a few of the above-mentioned motor and sensor skills yields sensor-guided robot motion that leads to goal-directed robot behavior.

### 3.3.1 Fixating

Visually fixating an environmental point of interest requires, first, a sensor skill that continuously delivers the image coordinates of this point with respect to a predefined fixation point in the image, and second, a motor skill that computes motion control words for the camera head motors in order to minimize the difference between reference and fixation point. A simple proportional control law is used to derive the velocities of the camera head motors: The difference in the y-coordinate between reference and fixation



**Figure 5:** Positioning error (commanded position - actual position) of the wrist pitch joint during the grasped object's downward movement. As soon as the object touches the table, the positioning error increases, leading to the detection of a touch event.

point is used to control the velocity of the tilt axis (elevation angle of the camera) and the difference in the x-coordinate is used to compute the required velocity for the pan-axis.

### 3.3.2 Wandering around with obstacle avoidance

Wandering around is a basic behavior that can be used by a mobile robot to navigate, explore and map unknown working environments. Our implementation requires the above-mentioned segmentation and fixating skills as well as basic motor skills for controlling the wheelbase. We further assume that the cameras cannot be rotated around their longitudinal axis, i.e., the top of the world is represented in the top of the images, and all objects to be detected rest on the ground plane. After having segmented the image in obstacle-free and occupied areas, the robot can pan its camera head towards those obstacle-free regions that appear to be largest. If such a region is classified as large enough, the steering angle  $\lambda$  of both wheels is set to half of the camera head's pan angle  $\alpha$  and the robot moves forward while continuously detecting, tracking and fixating this area. The robot will continue its smooth wandering locomotion until it reaches a dead end. Then, the robot stops and invokes a search pattern that scans the floor around the robot. At least in the robot's back (where it has come from) an obstacle-free path should be found. In this case the camera points backwards (pan angle  $\alpha = 180^\circ$ ) and the steering angles for both driven wheels are set to  $\lambda = 90^\circ$ , according to the given control law ( $\lambda = \alpha/2$ ). Thus, while the robot tries to move forward, it turns on the spot, and, automatically leaves the dead end situation.

### 3.3.3 Docking (approaching objects)

The main task of a mobile service robot with manipulator arms is to manipulate various objects at different locations. Prerequisite for manipulating objects is to bring them into the working range of arms and grippers, i.e., to navigate the robot sufficiently close to the object to be manipulated. We propose a visual servoing method enabling the robot to approach objects that are in its field of view and to stop in front of them at a pose (position and orientation) that is suitable for subsequent manipulation. It is based on the continuous tracking of a predefined reference line that corresponds to a physical edge of the docking object (e.g., a table's front edge) and the fixation of a reference point of the object (e.g., a table corner).

Prerequisites for showing this docking behavior are the segmentation skill (as a basis for object detection), the fixation skill (for keeping the object's reference point at the fixation point in the center of the image), motor skills (for wheelbase control) and cognitive skills (for deriving control words depending on the robot's perceived situation).

The main idea of the docking behavior is to derive the steering angle  $\lambda$  at each moment from the values of the pan angle  $\alpha$ , the slope of the reference line  $m = \tan \beta$  in the image and the tilt angle  $\gamma$  in order to maneuver the robot into its predefined final docking pose (e.g.,  $\alpha = 0^\circ$ ,  $\beta = 0^\circ$ ,  $\gamma = \gamma_{end}$ ). More

details of the docking behavior are beyond the scope of this paper and may be found in [Bischoff, Graefe 1998].

### 3.3.4 Taking, giving and placing objects

Interaction with people and objects requires tactile sensor skills. In combination with motor skills, such as gross arm positioning, objects can be received from or given to people, or placed onto other objects. Since the robot is not yet skilled enough to visually perceive the current pose of a human hand in order to conform to it, it brings its arm into a configuration where the human user could easily hand over objects or receive them. A touch event will signal that a human has closed the kinematic chain and is willing to receive or give an object. The robot then closes or opens its gripper, respectively.

To place objects onto other objects, the arm with the grasped object has to be grossly positioned first. Since the perceptual abilities are still limited and do not allow to visually guide the manipulator tip with the grasped object to the required location, the arm is fully stretched out first, and, then, commanded to move the first elbow joint down and the wrist joint up with the same velocities, to yield a downward movement of the gripper and, at the same time, to keep it aligned with an assumed horizontal surface (e.g., a table). Supervising all arm modules for a touch event will indicate when either the robot arm or the grasped object have touched something. Figure 5 shows an example of the positioning error of the wrist joint during the arm's downward (and the wrist joint's upward) movement. As soon as a touch event is detected, the arm is halted and the gripper is opened, thus relieving immediately the minimal structural stress upon the robot and the object that occurs when the kinematic chain closes.

## 3.4 Communicative Skills

The communicative skills of the robots are mostly based on natural language. Natural language is used both to instruct the robot and to generate easy-to-understand messages for the user. Commands may be input via voice (via a wireless microphone) or keyboard (directly via a wireless keyboard or indirectly via e-mail messages that may contain multiple commands). The robot displays its messages either on a screen, sends e-mails or generates speech from text.

To enable natural language processing with limited computational resources, a simple grammar has been designed. It is able to cover all the commands, statements and questions that might be given to the robot. Command sentences have a simple structure allowing them to be classified after the first word, thus facilitating the interpretation of the following words. Examples for command sentences are object and action-oriented instructions such as "Go to the kitchen!", or "Grasp the small ball!". Directive instructions such as "Turn around!" or more complex commands like "Turn left at the next intersection!" are supported as well. Intervening commands that do not contain a command verb are partly supported, e.g., "faster" (instead of "move faster"). In this case

these adverbs are treated like single command words. Questions are allowed as well, e.g., “What”, “Where” and “How”. Only a few questions can be answered by the robot so far, e.g., “What is your status?”, “Where are you?”, “How do I

get to the kitchen?” etc. The fixed syntax obviously does not allow an arbitrary reordering of parts of the sentences, e.g., “Take the glass, the big one” or “The glass over there, please take it”. Those sentences could however, be admitted, if they had been defined in advance (e.g., using the macro language described below).

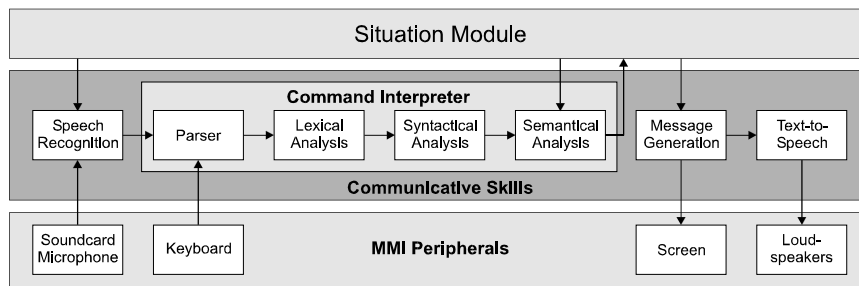
### 3.4.1 Command Interpreter

A command interpreter handles all user input. It consists of a parser, a lexical analysis, a syntactical analysis and a semantical analysis (Figure 6). The parser is fed by a text string provided by the speech recognition module or a keyboard. It separates the character string into a sequence of words and numbers using space, tabulator and punctuation characters as delimiters. This list is given to the lexical analysis where each word is looked up in a dictionary to obtain its type. Possible types are command verbs (e.g., go, take, place), locations (e.g., office, kitchen, workshop), prepositions (e.g., to, on, onto, in, into), objects (e.g., ball, table, pen) and fill words (e.g., please), just to name a few. Character strings enclosed in quotation marks are treated as one part of a sentence of type “text string”. The following syntactical analysis tries to identify the structure of the sentence by comparing the list of types with a list of prototype command sentences that includes all the commands the robot is able to understand. If the comparison is successful the semantical analysis will eventually provide missing words or place holders (such as “it”) from the robot’s situated knowledge in order to make the command complete.

### 3.4.2 Speech Recognition

Speech recognition is, in many regards, an unsolved problem. It is especially difficult if speech has to be recognized in the presence of a high level of ambient noise, e.g., inside a moving car or in office environments where the ambient noise includes various machinery sounds, telephones, moving persons or background conversations. Many methods have been proposed to increase the robustness of speech recognition systems, e.g., training in noisy environments or using multiple microphones, e.g., [Matsui et al. 1997].

Since commercial systems able to cope with these problems are not yet available we require, for the time being, the human to use an ordinary wireless microphone to send his commands to the robot. The used speech recognition engine



**Figure 6:** Visualization of the data flow between the peripherals of the man-machine interface, the communicative skills and the situation module. The robot’s current situation influences the speech recognition and the semantical analysis of the command interpreter.

is a commercial product enabling speaker-independent recognition of continuous speech, which means that users may speak to the system naturally, without pauses between words. This is a very important feature because it allows anybody to

communicate with the robot without needing any training with the system. The speech recognition engine generates text strings equivalent to the ones that may be entered via the keyboard.

To render the speech recognition more robust, larger word classes such as [object] have been split into several classes, e.g., [object\_to\_be\_manipulated] and [object\_used\_for\_navigation] which are now used as specific arguments of the command words TAKE or GRASP and MOVE or GO. The fewer the number of words per class and the stricter the syntax, the better the results of the speech recognition will be because fewer hypotheses have to be verified. Also, meaningful results are produced even under noisy conditions.

Another important way to increase the robustness of the speech recognition system has been the usage of so-called contexts that contain only those grammatical rules and word lists that are needed for a particular situation. Most parts of robot-human dialogues are situated and built around robot-environment or robot-human interactions, a fact which may be exploited to enhance the reliability and speed of the recognition process. When the robot knows what kind of answers it may expect from the user at a given moment it can switch to a context and disable or enable word lists that are appropriate for the current situation. For example, when the robot asks for confirmation, whether it should execute a certain task or not, the answers will be most likely “yes” or “no” and it would make no sense to expect, and to test, other words. By limiting the set of recognizable words or phrases that can actually be expected, the risk of recognition mistakes is reduced considerably.

However, at any stage in the dialogue a few words and sentences not related to the current context must be available to the user, too. These words are needed to “reset” or bootstrap a dialogue, to trigger the robot’s emergency stop and to make the robot execute a few other important commands at any time. For example, “Hello, *HERMES*” is used to begin a new dialogue, “Stop” and “Halt” are used for disrupting the robot from its current task, and “Stop listening” and “Continue listening” are used for disabling and enabling the speech recognition engine.

Depending on the prevailing situation and the type of dialogue conducted, various contexts are activated that can be

very simple, e.g., allowing only “Yes” or “No” when *HERMES* expects such an answer, or as complex as a navigation context in which multiple phrases and many words exist to allow complex robot control, especially during supervised learning of environmental features. To teach the robot object and place names, a spelling context has been defined that mainly consists of the international spelling alphabet. Since the spelling alphabet has been optimized for ease of use by humans in noisy environments, such as aircrafts, it should be well suited for robotic applications, too.

### 3.5 Data Processing Skills

Data processing skills organize and access the system’s knowledge bases. Three types of knowledge bases are being used: an attributed topological map for storing the static characteristics of the environment (for details see [Graefe, Bischoff 1997]), an object data base and a list of missions to accomplish. Depending on his preferences and on the abilities of the robot, the user may define the robot’s mission in more or less detail. A mission description may either consist of a detailed list of actions (e.g., elementary behaviors) that are to be executed sequentially by the robot, or only of a single command if the user has enough confidence in the robot’s planning abilities. For instance, a route is planned by a data processing skill based on Dijkstra’s shortest path algorithm in terms of vision-guided navigation behaviors, e.g., leave home base, turn right, stop at the second door to the right (no coordinates are used).

## 4 Experiments and Results

An experiment designed to allow an assessment of *HERMES*’ actual overall performance based on its currently implemented skills has been conducted (Figure 8). First, a mission description containing a list of commands (Figure 7) has been sent to *HERMES*’ e-mail address ([hermes@unibw-muenchen.de](mailto:hermes@unibw-muenchen.de)).

To acknowledge the reception of this valid user request, *HERMES* sends back an e-mail message stating that it will execute the commanded mission. One could say that its situation has changed the moment the message arrived (by

```

go to the secretariat
take over tray
place tray onto the table in the kitchen
go home

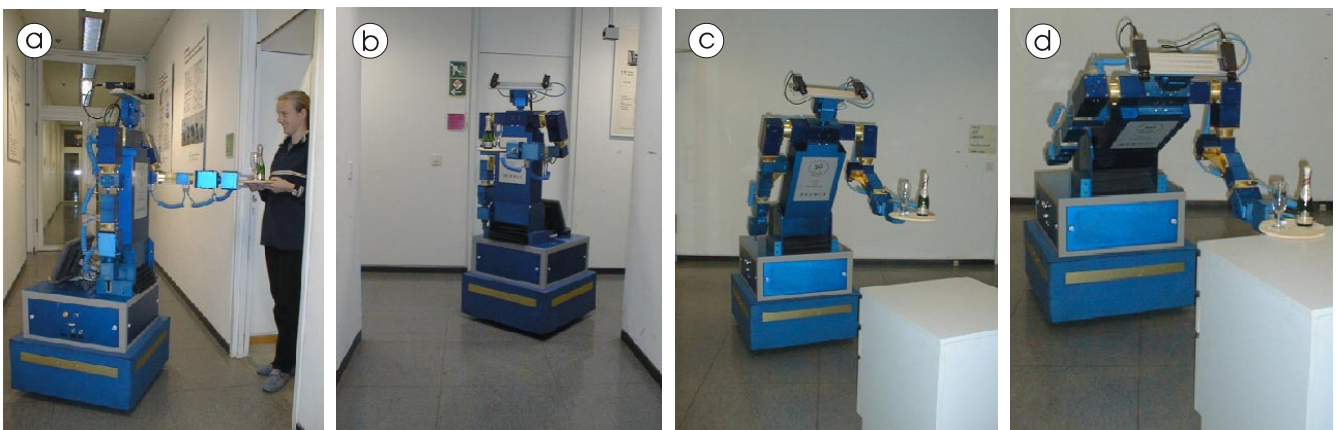
```

**Figure 7:** Simple mission description provided as a text file.

setting a new transient goal). The required paths to the secretariat, to the kitchen and to its home base are planned in terms of elementary navigation behaviors (see section 3.5). After having arrived at the secretariat *HERMES* brings its arm and gripper into an appropriate pose. From its knowledge base it knows how it has to grasp a tray so that objects on it will not fall down. When the user places the tray between the two fingers of the gripper and thereby slightly moves one of the fingers, *HERMES* perceives a touch event, and closes the gripper. *HERMES* is able to check for the presence of an object between its fingers by identifying their stop positions. If an object is present, *HERMES* must assume that the right object has been handed over, as it is not yet able to visually verify the presence of the tray. Thereafter, *HERMES* proceeds on its way to the kitchen. Once arrived, it searches for the table and initializes the vision-guided docking behavior, as the table may have been moved since the last mission to this location. After having successfully approached the table, *HERMES* places the tray onto it. Finally, *HERMES* continues with the next command in the list and returns to its home base. From there it sends an e-mail message to notify the user about the successful completion of the task.

This simple fetch and carry experiment is a representative example of some of the capabilities of the robot. Speech recognition was not needed during this experiment. However, it could have been used as well, e.g., to give further instructions after the robot’s arrival at the secretariat.

Although the commands of the mission description (Fig. 7) could have been given via voice, the number of problems that the robot could solve by conversing dialogues is still limited. For a more natural dialogue-oriented task instruction and supervision more situations and contexts have to be defined.



**Figure 8:** *HERMES* performing a simple service task upon user request: a) receiving a tray (with a bottle and a glass) from a user; b) navigating in a network of hallways towards the commanded goal location; c) approaching a table at the goal location and d) placing the tray onto it.

Although visual sensing skills are based on algorithms that only work under quite restrictive assumptions, they are well suited for studying various control algorithms and validating the proposed situation-oriented behavior-based system architecture. In combination with some basic motor skills more powerful sensorimotor skills can be created and lead together with communicative and data processing skills to goal-directed behavior.

Vision-guided docking yields good results. The robot may start its approach from arbitrary poses and stops sufficiently close in front of the docking station and in parallel to it. It is important to notice that both the robot's trajectory and the final docking pose are directly derived from sensor data (image features and encoder readings). They are not calculated from distance measurements, kinematic models and inverse perspective transforms with respect to a fixed reference frame (world coordinate system). The method is generic and suitable for all kinds of docking or goal objects where a reference line and a specific point can be defined. In addition, it may be implemented in a calibration-free or self-calibrating way.

Kinesthetic sensing is sufficiently accurate to provide the robot with as a sense of touch. Arbitrary objects can be placed onto other objects without using visual feedback. It remains to be validated if this sense can be used for minimizing damaging effects caused by collisions between *HERMES'* manipulators and other objects. Nevertheless, high resolution tactile sensors for both grippers and around the wheelbase are being developed to enhance the robot's perceptual abilities.

Many more experiments have been carried out and other behaviors have been created based on the elementary skills presented here, such as filling a glass with water (including recognition of the glass and the water level) and following persons. Many pre-programmed control sequences and taught motion patterns can be modified and executed via a wireless keyboard, thus allowing *HERMES* to be teleoperated in all its degrees of freedom for entertaining purposes.

## 5 Conclusions and Outlook

Still much research is needed to endow future personal or service robots with skills enabling their deployment in massive numbers in environments cohabited by humans. Since users of such robots will not be robotic experts, the robot design has to be the more human-like and its control has to be the more human-friendly, the closer the contact with humans will be.

If a robot's sensor modalities include vision, touch and hearing they allow an understanding of the actual situation being perceived by both the robot and the user on a similar level of abstraction. This is prerequisite for instructing the robot in a natural way and for all kinds of cooperative tasks to be performed.

Due to the innate characteristics of the situation-oriented behavior-based approach that we have proposed for control, our humanoid service robot *HERMES* is able to cooperate with a human and to accept orders that would be given to a human in a similar way. To improve the human-friendliness of the interface further, current speech recognition and speech output capabilities have to be enhanced, and to improve manipulation and navigation skills both grippers and wheelbase will be equipped with tactile sensors.

## References

- Arkin, R. C. (1998).** Behavior-Based Robotics. MIT Press, Cambridge, MA, 1998.
- Babcock, P. (1976):** Webster's Third New International Dictionary of the English Language, G. & C. Merriam Company, Springfield, MA, 1976.
- Bischoff, R. (1997).** *HERMES* – A Humanoid Mobile Manipulator for Service Tasks. Proc. of the Int. Conference on Field and Service Robotics. Canberra, Australia, December 1997, pp. 508-515.
- Bischoff, R.; Graefe, V. (1998).** Machine Vision for Intelligent Robots. IAPR Workshop on Machine Vision Applications. Makuhari/Tokyo, November 1998, pp. 167-176.
- Bischoff, R.; Graefe, V. (1999).** Integrating Vision, Touch and Natural Language in the Control of a Situation-Oriented Behavior-Based Humanoid Robot. IEEE Conference on Systems, Man, and Cybernetics, October 1999, pp. II-999 - II-1004.
- Graefe, V.; Bischoff, R. (1997).** A Human Interface for an Intelligent Mobile Robot. 6th IEEE Int. Workshop on Robot and Human Communication, Sendai, Japan, Sept. 1997, pp. 194-197.
- Horswill, I. (1994).** Visual Collision Avoidance. IEEE/RSJ International Conference on Intelligent Robots and Systems, Munich, Sept. 1994, pp. 902-909.
- Schraft, R. D.; Schmierer, G. (1998).** Serviceroboter – Produkte Szenarien, Visionen. Springer-Verlag, Berlin, 1998 (in German).