From: AAAI Technical Report FS-97-02. Compilation copyright © 1997, AAAI (www.aaai.org). All rights reserved.

HuDL, A Design Philosophy for Socially Intelligent Service Robots

D.M. Wilkes, RT. Pack, A. Alford, and K. Kawamura
Intelligent Robotics Laboratory
Vanderbilt University
Department of Electrical and ComputerEngineering
Nashville, Tennessee, 37235 USA
{wilkes,pacman,alford,kawamura}@vuse.vanderbilt.edu

Abstract

The motivation for integrating humans into a service robot system is to take advantage of human intelligence and skill. Human intelligence can be used to interpret robot sensor data, eliminating computationally expensive and possibly error-prone automated analyses. Human skill is a valuable resource for trajectory and path planning as well as for simplifying the search process. In this paper we present our plans for integrating humans into a service robot system. We present our paradigm for human/robot interaction, called Human Directed Local Autonomy (HuDL). Our testbed system is described, and some example applications of HuDL for an aid to the physically disabled are given.

Introduction

Service robots are socially intelligent agents by the very nature of what they are supposed to do. They are to be useful robots that interact closely with humans through as natural interfaces as possible. The relationship can be thought of as symbiotic in the sense that both the human and the robot work together to achieve goals, for example as aids to the elderly or disabled and even in manufacturing.

Since a service robot is a truly concrete physical realization of a socially intelligent agent, its development represents an extremely challenging task. Many difficult problems in planning, learning, control, communication, signal processing, vision, etc. must be solved to produce a robustly functioning system. Attempting to achieve such a system and simultaneously require that it exhibit a high level of autonomy is very difficult, very expensive, and typically impractical at the present time. We use a guiding philosophy for research and design that explicitly supports the evolution of a robot from a system with limited abilities and autonomy to one with a high degree of both. The philosophy we propose is Human Directed Local Autonomy, or HuDL.

HuDL is based on exploiting the symbiotic relationship between the human user and the robot. In essence, the idea is to make maximum use of the things that humans do well and the things that robots can do well. A good example is a robot to aid a physically disabled person, perhaps suffering from some degree of paralysis. The human is intelligent, but has physical limitations in mobility and dexterity. The robot is mobile and/or able to manipulate objects, but perhaps lacks the intelligence to solve many real-world problems. A symbiotic combination of human and robot can improve the ability of the human to control his environment and enhances the usefulness of the robot by significantly augmenting its intelligence with that of the user. One key feature is flexible human integration. The background for this is clear: Human intelligence is superior whenever unexpected or complicated situations are met. Roles and methods for integrating humans into different types of activities must be defined and developed.

For example, the user may request the robot to go into the kitchen and bring him a drink. The robot may have sufficient intelligence to navigate to the kitchen, but the planning and object recognition problems may prove too difficult. If appropriate visual feedback is supplied to the user (perhaps a monitor lets him "see" what the robot sees) he can narrow the search space for the robot by giving high level instructions (perhaps through a speech recognition system that provides a natural and convenient interface) such as "Pick up the red glass on the counter to your right." The robot may see several red objects that could be the glass in question, moves forward, points to the most likely object, and sends a message to the user, "Do you mean this one?" The user responds "No, that is a bottle of catsup, the glass is further to your right." In this way the user guides the robot through the difficult object recognition task, greatly enhancing the likelihood of success. Similarly, the robot has enabled the human to have more control over his environment.

HuDL allows the achievement today of a service robot of useful but limited intelligence and ability. Over time the system will be improved by the addition of better user interfaces, object recognition, reasoning, planning, navigation, etc. as well as improved hardware such as better manipulators, cameras, sensors, computers, etc. Thus a "glide path" toward much greater autonomy is established in the near term and yet a high degree of *interactivity* (one of our main goals) is achieved as well. In particular, newer user interfaces need to be developed that enable the robot to detect and understand the users feelings

Copyright © 1997, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

and emotional state such as joy, anger, fatigue, etc., and to act in an appropriate manner for that emotional state. Also, the robot should have a learning capability to allow it to adapt its behavior to the particular needs and personality of the user. Thus, over time the service robot becomes suited to a particular individual or group of individuals.

In the Intelligent Robotics Laboratory (IRL) at Vanderbilt University, we are using HuDL to guide the development of a cooperative service robot team consisting of a dual armed stationary humanoid, called ISAC, and a modified Yaskawa Helpmate mobile robot, simply called Helpmate. The user interfaces currently under development include speech recognition (for verbal input), text to speech (for verbal output), vision (for tracking of faces and hands as well as many other tasks), gesture (a vision based interface), and sound localization. These interfaces are being used to make the overall interaction with ISAC and Helpmate into a natural "multimedia" experience that is comfortable for non-technical users. ISAC is even able to play an electronic musical instrument, a theremin, with perfect pitch. Indeed, this may be one of the most interesting social skills of a service robot to date.

Human Directed Local Autonomy

For the past ten years, our lab has focused on service robotics research. Specifically, we have developed robotic aid systems for the disabled (Bagchi and Kawamura 1994), (Pack and Iskarous 1994), (Kawamura et al. 1996a). We have continually observed that a key research issue in service robotics is the integration of humans into the system. This has led to our development of guiding principles for service robot system design (Kawamura et al. 1996b). Our paradigm for human/robot interaction is HuDL.

Pook and Ballard (Pook and Ballard 1995) have shown that both full robot autonomy and human teleoperation of robots have disadvantages. In their deictic teleassistance model, a human uses hand signals to initiate and situate a robot's autonomous behaviors. This deictic strategy has also been used to control a mobile robot used (Cleary and Crisman 1996). HuDL also maintains the concept of autonomous robot behaviors directed by a human user. The philosophy of HuDL is that the human is "in the loop." The human does not teleoperate the robot; but rather commands and guides the robot at a higher level.

Integrating humans into a robot system at a level above teleoperation has several advantages. First, is the use of the human's intelligence and decision making abilities. For example, the human can interpret the robot's sensor data (e.g., indicating a target object in a camera scene), thereby simultaneously reducing the computational burden on the robot and increasing the robustness of the overall system. The human can also detect an exceptional or error situation, and assist the robot in recovering (Frohlich and Dillmann 1993). However, the human is not directly or explicitly driving the robot's actuators. This relieves the human of the often tedious and frustrating, or in the case of the physically disabled user, impossible task of manual teleoperation.

Our interaction model is *dialog-based, rhythmic*, and *adaptive*. Dialog-based robot human interaction has been explored in areas such as robot programming (Friedrich and Dillmann 1995) and as an interface to a mobile manipulator (Lueth et al. 1994). A *dialog-based* interaction provides an active mechanism for the robot to receive information from the human; that is, the robot can direct the human's input in order to satisfy the robot's information requirements.

The iterative nature of the dialog leads us to the idea of *rhythmic interaction*. Rhythmic responses from computer systems have become important indicators of correct function: animated cursors for windowing systems, animated icons for web browsers, even the sound of disk access—all these things let us know that our computers are working. The same idea can apply to a robot. As long as it can provide a rhythmic response, the human is more confident that it is functioning.

If the robot/human interaction is rhythmic, exceptional or error states can be detected by the simple technique of the *timeout*. The human expects the robot to respond within a certain time interval. If it does not, the human may suspect the robot has failed in some way. Likewise, the robot also expects some input from the human. If the human does not provide it, the robot may ask if the human is confused, busy, or perhaps even in need of medical attention.

As the human becomes more familiar with the robot, and proficient in interaction, his interaction patterns will change. Consider the user of a word processor. At first, as he discovers the capabilities of the software, he probably does things "the hard way," by searching through menus, etc. As he becomes more familiar with the program, he begins to learn shortcuts. Likewise, the user of a robot is likely to desire shortcuts in using the robot. Besides the obvious customizations of shortcuts, macros and preferences, the *adaptive* aspect of the human/robot interaction provides richer avenues for "personalizing" the robot.

For example, the robot may begin to anticipate what the human will request next. Although it would be annoying and possibly dangerous to have the robot execute tasks before it is asked, the robot could prepare to execute the task by, for example, searching for target objects or positioning its manipulators.

We want the human and robot to interact using as many different media as possible. Ideally, these media include those natural to human interaction, such as speech (Frohlich and Dillmann 1993), (Stopp et al. 1994), gestures (Pook and Ballard 1995), and touch (Muench and Dillmann

1997). The motivation for using these natural modes is that they are familiar to the user, are more comfortable, and require less training for the user. Computer-based GUIs with pointing devices can also be used effectively (Cleary and Crisman 1996), and because of our robot control hardware and software architecture, are the "natural" interface of our robots.

For example, suppose a robot is requested by the user to retrieve a certain item, but the robot's visual object classification system is not robust enough to identify correctly the object in the current environment. The robot can indicate its best estimate of what it "thinks" is the object to the user. The user responds either by saying "Yes, that is the item," or "No, the item I want is the large item *to the right*." Use of symbolic terms such as *to the right* is more natural and convenient for human/robot interaction. Using natural language for describing spatial relationships is explored further in (Stopp et al. 1994) and (Maaß 1995).

Testbed System

This section describes our testbed system, which consists of a dual-arm humanoid robot and a mobile robot, and a human user (Figure 1). We also describe our control software architecture, the Intelligent Machine Architecture (IMA). The IMA is described in more detail in (Kawamura and Pack 1997) and (Pack et al. 1997).

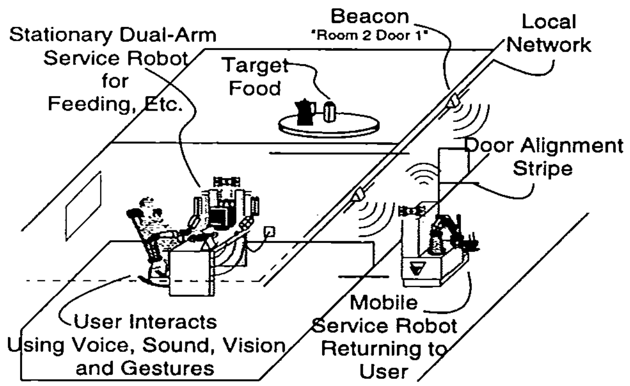

Figure 1: Service Robot System Testbed

Intelligent Machine Architecture (IMA)

The Intelligent Machine Architecture (IMA) is a new approach for the design of the control software for intelligent machines that are principally limited by difficulty in integrating existing algorithms, models, and subsystems. The IMA differs from traditional software systems in two aspects. First, the IMA uses a system level model that is based on *primitive agents*. Thus, each resource, task or domain element is modeled in software as a primitive agent. Figure 2 shows a primitive agent decomposition of a dual-arm humanoid service robot. Second, the IMA uses a primitive agent level model that is component-object based. This two level approach to the problem of complex software design for intelligent systems addresses both software engineering issues such as reuse, extensibility, and management of complexity as well as system engineering issues like parallelism, scalability, reactivity, and robustness. The result of the approach is a system of concurrently executing software primitive agents, formed from reusable component objects, that comprise the control software for an intelligent machine.

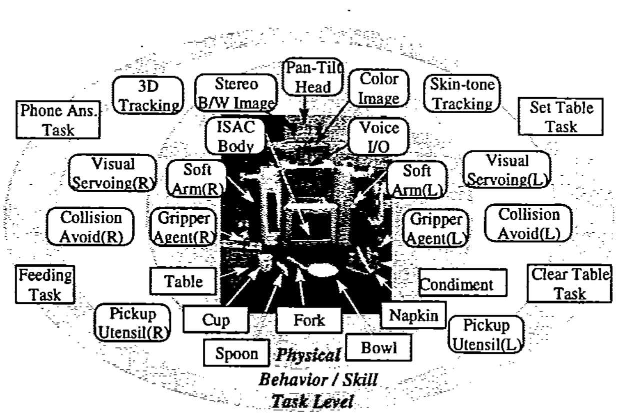

Figure 2: IMA Primitive Agent Decomposition for Dual-Armed Humanoid Service Robot

A primitive agent within IMA is assumed to have certain properties that separate it from the currently popular term "Agent" and focus on what is essential to being an agent in the context of software system development. The properties of a primitive agent assumed in IMA are:

Autonomy - Primitive agents own their components. They are separate and distinct concepts or objects in the domain for which the software is written. Thus, they have a strong conceptual encapsulation boundary at the system level.

Proactivity - Primitive agents act locally, based on their own internal state, resources and observations of other primitive agents through specific relationships. The core of the primitive agent's operation is a decision or action selection process.

Reactivity - Primitive agents are reactive because they respond to changes in their external environment.

Connectivity - Primitive agents are cooperative because they give and receive information from other primitive agents to achieve their tasks.

Resource Bounded - Primitive agents are resource driven and may be competitive because they represent only a single conceptual element and depend on other primitive agents as their resources. In a properly structured system, the competition between primitive agents for a resource should represent a natural property of the system, not a bottleneck in the software architecture. In these cases, arbitration between various actions becomes important.

An additional feature of this system design is that the primitive agent-based decomposition facilitates parallel processing and can take advantage of both distributed and SMP computer systems. Finally, each primitive agent acts locally based on internal state and provides a set of services to other primitive agents through various relationships. The semantics of a relationship between primitive agents is that one agent provides another with resources for action.

Dual-arm Humanoid

Our dual-arm humanoid robot hardware consists of a pair of 6-degree-of-freedom Softarm manipulators and a pan/tilt camera head. The Softarms are actuated by artificial pneumatic muscles called Rubbertuators. These arms have several advantages, e.g., they are low-power, lightweight, and naturally compliant. They are ideal for use in situations where contact with humans is necessary. On each of the Softarms we have installed a 6-axis force/torque sensor at the wrist. The Softarms are controlled by a PC expansion card designed in-house. The camera head is a Directed Perceptics pan/tilt unit on which we have mounted a stereo vergence platform designed in-house. This platform holds two color CCD cameras. The robot's software runs on a network of PCs.

Mobile Robot

Our platform for mobile robotics and mobile manipulation research is an augmented Yaskawa HelpMate. The HelpMate is a differentially steered mobile base with a bank of forward- and side-looking SONAR sensors mounted in a vertical panel. To this robot we have added a laser ranger (LIDAR), a vision system based on a pan/tilt camera head, a manipulator, and two Pentium computers.

The manipulator is a 5 degree-of-freedom Softarm mounted on the left side of the robot, directly behind the sonar panel. The entire workspace of the arm is on the left side of the robot. The Cost-effective Active Camera Head (CATCH) is a 4 degree-of-freedom stereo camera head with pan, tilt, and independent verge. CATCH has two color cameras and is mounted on a platform near the front of the robot. The camera head is offset to the left, so that the Softarm's workspace can be viewed without occlusion by the body of the robot.

The robot has two onboard computers. A 150MHz Pentium handles the drive, SONAR, LIDAR, and Softarm, while a 166MHz Pentium with MMX handles vision processing. The two computers are connected by a 10-BaseT Ethernet network. In addition, the 150MHz Pentium is connected to our lab's LAN by radio Ethernet. This allows the HelpMate to be part of a distributed multi-robot system.

Application of HuDL to Service Robotics

This section shows how the concept of HuDL can be used to integrate humans into service robot system for aiding the physically disabled. We are currently developing demonstrations for our service robot system using the IMA. What follows is a description of a pair of related scenarios we have chosen to help us develop and integrate service robot technology, including human interfaces. Thus, in the words of the recent NSF Workshop on Human-Centered Systems, our research can be characterized at least in part as a "Testbed-Style Research Project within a Situated Context," (Woods and Winograd 1997). Our system uses ISAC, a dual-arm humanoid robot, and Helpmate, a mobile robot with one arm, as a service robot team. This pair of benchmarks was selected because there is direct human interaction, a relatively unstructured environment, and simplifying assumptions can be gradually relaxed to increase problem complexity as our development proceeds.

Humanoid Service Robot Aid Task

ISAC is activated and the benchmark demonstration is started. ISAC begins scanning its environment for utensils, food, telephones, and other items relevant to its task. Once a certain level of certainty about this environment is achieved, ISAC begins scanning in front of itself for a user. Strange or badly classified items are noted for later use. If too much time passes, ISAC re-scans the environment and then comes back to looking for a user. When a user is detected (skin-tone, face detection, voice command and combinations of these) ISAC begins with a greeting using voice and gestures. ISAC will then try to determine the identity of the user, either through a detection algorithm or by asking for confirmation of identity. Identification is not necessary to proceed, but could be used to customize responses or subsequent behavior. ISAC may then ask the user for help (through voice, gestures and visual feedback) in identifying things that were not automatically classified in the initial pass and introduce the user to some options.

Once the user is in place and the robot has begun interacting, the user might ask (via a voice command) ISAC to feed him soup. ISAC should confirm this with the user using voice and gesturing to the soup. If soup is not on the table, ISAC should set the table from a side cart. If there is no soup, ISAC should say so and list the food items it knows it actually has as alternatives for the user. Assuming that soup is requested and is available, then the demonstration should proceed.

ISAC should place the soup close to the user and pick up the spoon. The active vision system will then alternate between tracking the user and fixating the bowl to guide both parts of the feeding motion. ISAC will enter a cycle of dipping up soup (dip confirmation might use color or force information, for example) and bringing it to the user's mouth. A force transient will signal that the soup is taken and the cycle will continue. Then, perhaps the telephone starts ringing when the robot has soup in the spoon. It should start taking the spoon back to the bowl and begin locating the telephone. As soon as it is located, the robot should pick up the telephone with its free hand while shaking off soup from the spoon with its dipping hand. When the conversation is over, signaled by force on the telephone, ISAC should hang up and ask if it should resume feeding.

Mobile Manipulator Fetch Task

In this example, the user requests that HelpMate, the mobile manipulator, fetch an object such as a soda can, from another room and bring it back to ISAC. In such a scenario, the robot must complete a series of subtasks in the proper order:

1. Go to the correct room
2. Pick up the soda can
3. Return to the original room
4. Hand the soda can to ISAC

We assume that in steps 1 and 3 the robot uses a map-based navigation program. The granularity of the map is fine enough to place the robot close to the target object, i.e., in the same room. In steps 2 and 4, the robot uses vision, both to move toward the target destination and to guide its manipulator for pickup and deposit. We will call steps 1 and 3 the *map mode* of operation, and steps 2 and 4 the *visual servoing mode*.

One problem that occurs in map mode is that the robot may become trapped or lost. In this case, the human assists the robot; the problem now becomes a question of how the human will know when to intervene. Our solution is twofold: (1) allow the human to monitor the robot's progress using sensor data and (2) have the robot monitor its own progress and decide when to ask the human for help.

Sensor data, such as the robot's dead reckoning, range data from SONAR or LIDAR, or a video stream can be used to give the human a graphical representation of the robot's surroundings. From this the human can identify the robot's position, decide if the robot is lost and, if so, how to correct the situation. Likewise, monitoring the robot's speed can indicate whether the robot has malfunctioned or become trapped.

For the robot to determine if it needs help, we will use supervisor primitive agents to monitor the robot's progress. For example, the navigation supervisor will make an initial estimate of total spatial displacement and an initial estimate of the time required to finish the navigation phase. It will also monitor the robot's speed. Should either the navigation take too long or the robot's speed fall below a threshold for too long, the robot will request help from the human.

In the visual servoing mode, the robot tracks the target object in a video image. To initialize the tracking, the robot interacts with the human to locate the object in a video image. The robot presents an image of the scene, highlighting its estimate of the object's location. The human can agree with the robot, or can direct the robot to look at another location. This process is repeated until the robot has found the object. This reduces the complexity of the problem of initializing a tracking algorithm by restricting the search space of the initialization. The interaction between the human and the robot can include many modalities such as speech, mouse, touch sensitive screen, and gesture.

A supervisor primitive agent for tracking can be used to ask the user to reinitialize the tracking algorithm. This agent can make assumptions about the velocity of the target object, based on information from the human; e.g., if the target is on a table, it can be assumed to be stationary. The supervisor can compute the object's apparent velocity based on the robot's velocity and the velocity of the active camera head, and compare this with the object's assumed velocity. The tracking algorithm itself may report a confidence value to the supervisor agent, providing additional information about the success of the tracking.

An IMA primitive agent decomposition of this system is shown in Figure 3. The robot's physical resources and skills (e.g., visual servoing, navigation), the human, and the fetch task itself are all represented by primitive agents.

Conclusion

In this paper, we have described HuDL as our design philosophy and guiding principle for building service robot systems and for integrating humans into these systems. We have also described the IMA as a software architecture for facilitating the implementation of robot systems. We have described two example applications of how we will use IMA and HuDL to achieve a practical service robot system for aiding the physically disabled. We are currently implementing our ideas on the testbed systems described in this paper.

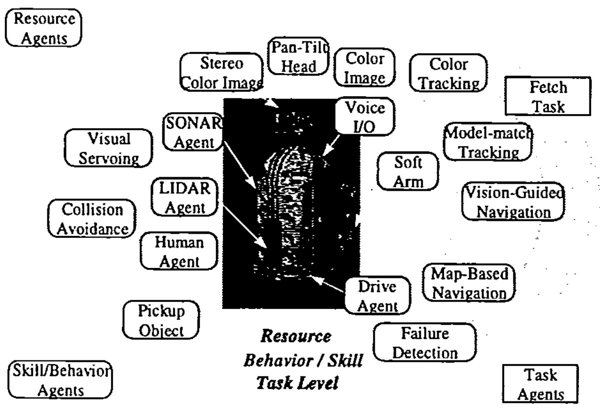

Figure 3: IMA Primitive Agent Decomposition of Fetch Task

References

Bagchi, S., and Kawamura, K.1994. ISAC: A Robotic Aid System for Feeding the Disabled. In Proceedings of the AAAI Spring Symposium on Physical Interaction and Manipulation. Stanford University.

Cleary, M.E., and Crisman, J.D. 1996. Canonical Targets for Mobile Robot Control by Deictic Visual Servoing. In Proceedings of the International Conference on Robotics and Automation, 3093-3098. Minneapolis, Minn.

Friedrich, H., and Dillmann, R.. 1995. Robot programming based on a single demonstration and user intentions. In Proceedings of the 3rd European Workshop on Learning Robots at ECML'95, M. Kaiser, editor, Heraklion. Crete, Greece.

Fröhlich, J., and Dillmann, R. 1993. Interactive robot control system for teleoperation. In Proceedings of the International Symposium on Industrial Robots (ISIR '93). Tokyo, Japan.

Kawamura, K., Wilkes, D.M., Pack, T., Bishay, M., and Barile, J. 1996a. Humanoids: Future Robots for Home and Factory. In Proceedings of the International Symposium on Humanoid Robots, 53-62. Waseda University, Tokyo, Japan.

Kawamura, K.,. Pack, R, Bishay, M., and Iskarous, M. 1996b. Design Philosophy for Service Robots. *Robotics and Autonomous Systems*, 18:109-116.

Kawamura, K., and Pack, R. T. 1997. Object-Based Software Architecture for Service Robot Development. In Proceedings of the International Conference on Advanced Robotics.

Lueth, T. C., Laengle, T., Herzog, G., Stopp, E., and Rembold, U. 1994. KANTRA - human-machine interaction for intelligent robots using natural language. In Proceedings of the ROMAN '94.

Maaß, W. 1995. How Spatial Information Connects Visual Perception and Natural Language Generation in Dynamic Environments: Towards a Computational Model. In: A.U. Frank, W. Kuhn (eds.), Spatial Information Theory: A Theoretical Basis for GIS. Proc. of the Int. Conference COSIT'95, Semmering, Austria, 223-240. Berlin, Heidelberg: Springer.

Muench, S., and Dillmann, R. 1997. Haptic Output in Multimodal User Interfaces. In Proceedings of the 1997 International Conference On Intelligent User Interfaces (IUI97), 105-112. Orlando, FL.

Pack, R.T., and Iskarous, M 1994. The Use of the Soft Arm for Rehabilitation and Prosthetics. In Proceedings of the RESNA 1994 Annual Conference, 472-475. Nashville, TN.

Pack, R.T., Wilkes, M., Biswas, G., and Kawamura, K. 1997. Intelligent Machine Architecture for Object-Based System Integration. In Proceedings of the International Conference on Advanced Intelligent Mechatronics. Waseda University, Tokyo, Japan.

Pook P.K., and Ballard, D.H., 1995. Deictic Human/Robot Interaction. In Proceedings of the International Workshop on Biorobotics: Human-Robot Symbiosis, 259-269. Tsukuba, Japan.

Stopp, E., Gapp, K. P., Herzog, G., Laengle, T., and Lueth T. 1994. Utilizing spatial relations for natural language access to an autonomous mobile robot. In Deutsche Jahrestagung für Künstliche Intelligenz (KI '94).

Woods, D., and Winograd, T. 1997. Breakout Group 3 Report: Human-Centered Design, NSF Workshop on Human-Centered Systems. Arlington, VA.