Design Issues for Assistive Robotics for the Elderly

Q. Meng and M.H.Lee
Department of Computer Science
University of Wales, Aberystwyth
Ceredigion, SY23 3DB, Wales, UK
email: {qqm, mhl}@aber.ac.uk

Abstract

The worldwide population of elderly people is growing rapidly and in the coming decades the proportion of older people in the developed countries will change significantly. This demographic shift will create a huge increase in demand for domestic and health-care services and this in turn has the potential to create a major new market for domestic service robots that can assist with the care and support of the elderly and infirm. However, unlike industrial robots, assistive service robots are still under-developed and are not widely deployed. We analyze the nature of the requirements for assistive robotics for the elderly and argue that traditional "industrial" robot engineering approaches are either inappropriate or inadequate to tackle the key problem areas, which we identify as: safety, adaptivity, long-term autonomy of operation, user-friendliness and low costs.

A key issue is *user acceptability* and this paper explores how seemingly difficult and possibly conflicting design requirements can be integrated in a human-centred approach. We develop an approach to the design of autonomous assistive robots for the home, with emphasis on the user and the tasks to be performed. We then introduce some design principles and apply these to a simplified case study. The case study was implemented as a concrete illustration, and a series of experiments are reported.

The demonstration shows, (a) how existing software techniques can be combined in a synthesis that satisfies several key design ideas, (b) how a software architecture can provide a flexible and extensible substrate for the integration of the design, and (c) how this approach can be sensitive to the concept of user "empathy" that is characteristic of these applications.

By highlighting significant design issues and suggesting different approaches, we hope assistive robotics will be better able to address the novel demands of assistive applications in health-care situations.

Keywords: Assistive Devices, Service Robotics, Autonomous Error Recovery, Human-centred Design.

1 Introduction and Motivation

In this paper we discuss some of the special difficulties that face engineers when developing new technologies for enhancing the quality of life of the elderly and infirm. Such technologies can aid both physical and cognitive function and are being intensively researched worldwide. Our aim is not to present a solution to a specific incapacity problem but to explore the special requirements that assistive devices impose, in order to highlight these issues for the benefit of future projects. Early experiments have shown that established industrial robotics techniques are simply not sufficient for transfer directly into the world of domestic and health-care robotics without extensive rethinking. The motivation of this paper is to explore some of this necessary rethinking, to highlight significant design issues, and to suggest how different approaches may be developed. We illustrate our approach through a case study that implements a simple reaching task. However, this assistive robot model is not intended as an actual prototype of a future device; hence we do not cover user trials, usability assessments, psychosocial aspects and many other factors that come into play during full product development.

Rather, our goal is to provide a concrete illustration of how seemingly difficult and possibly conflicting design requirements can be integrated in a human-centred approach. Although somewhat artificial, the example system does show (a) how existing software techniques can be combined in a synthesis that satisfies several key design ideas, (b) how a particular architecture, the behaviors model, can provide a flexible and extensible substrate for the integration of the design, and (c) how this approach can be sensitive to the concept of user "empathy" that is characteristic of these applications.

2 The Current State of the Art

Over the last few decades, industrial robotics has become a significant commercial success. Domestic and service robotics are also now becoming accepted, particularly in the entertainment sectors. In this paper we are interested in assistive devices for the home-care of the elderly and infirm — a field in which robotics should have a potentially major contribution to make. First, we define our terms; we consider *an assistive robot is a device that cooperates with a user through physical activity in the user's environment.* We assume the term "robot" does not properly apply to purely electronic devices that are unable to alter the physical world and "cooperation" implies the need for some physical contact with the user. Thus we find it useful to separate out areas such as medical telematics. It is rewarding to examine these distinctions further and so we now explore different kinds of assistive scenario.

Recent years have witnessed considerable progress with autonomous mobile robots that clean floors, cut grass, or act as guides, couriers or security guards [22]. These robots are usually based on mobile platforms that have to navigate through human environments and communicate with humans during operation. They can carry objects or perform some function with a tool or accessory but their *physical interactions* with the world are bounded or controlled by the task specification. The significant feature of these devices is that the user interactions are not normally physical but purely communicative [18]. Those systems that do have requirements for *physical contact with users*, even if indirect, entail additional major challenges for assistive technologies. We can distinguish the different types in terms of *levels of physical interaction with users* as follows:

Level 0 This level has no physical interactions other than communications. Examples include timers, medication reminders, monitors, alarms and tele-links. Such devices have no robotic features, i.e. no spatial machinery, and there are no special safety concerns, other than those of any electrical or computerized home appliance. This level has recently been a popular application area for intelligent agent technology.

Level 1 Devices at this level are able to move within the user's environment but should generally avoid physical contact with users and even avoid the user's personal space where possible. Examples are autonomous vacuum cleaners, lawn mowers, tour guides and couriers. Extra safety considerations must be given to events such as the user accidently colliding with or falling over the robot.

Level 2 This level covers devices that can intimately share the same operating space as the user. Examples would be robot devices that fetch or retrieve objects, assist with domestic tasks such as cooking or ironing, or perform specific manipulation tasks on a desk top or work area. Some form of cooperation between user and robot is necessary for normal operation and extra safety issues include the specific hazards caused by powered kinematic devices operating near human users.

Level 3 The highest level of physical interaction occurs when the device is in frequent contact with the user during operation. Examples include rehabilitation devices that exercise the user's limbs, powered walking aids, and motorized wheelchairs. Safety is very important because of the close-coupling with the user and failures could easily lead to injury.

Table 1 lists some characteristics of these levels. On this scale, service devices are those at levels 0 and 1, and assistive robotics proper only exist at levels 2 and 3. It is interesting to note that levels 0 and 1

level	Example	Physical interactions	Degree of contact	Spatial world	Workspace	Safety
0	Medication monitor	Only communication	None	None	Not relevant	As for normal (static) domestic appliances
1	Floor cleaning robot	Accidental collisions	Slight	2D	Shares access areas with user but contact normally avoided	As for powered tools and vehicles
2	Object retrieval robot	Transfer of objects, Holding tools or objects, Providing reactive forces	Frequent	3D	Shares user's working envelope and contact expected	Additional user protection required
3	Wheelchair robot	Active movement of user's whole body or parts of body	Almost constant	3D	Shares user's body space and maintains contact for long periods	Safety critical design essential

Table 1: Levels of interaction in assistive robotics

are distinguished by less control coupling as well as less physical coupling with their users, i.e. they can usefully operate without the user for long periods. Level 0 has received the most research attention but this has tended to concentrate on patient monitoring, tele-diagnosis and remote care delivery. While this focus on social and organizational management is welcome there has been relatively little work on direct user quality of life [4]. Research on level 1 devices is also very active, especially on mobile robotics, but most of this does not address any particular needs of the elderly [19]. We note that mobile robots usually operate at floor level and their operating space can often be considered as two dimensional, thus greatly simplifying their design problems. Level 3 is a very active research area, covering mobility issues, especially autonomous wheelchairs, and also rehabilitation and exercise robots [21, 6]. In this paper we address issues concerning devices on level 2, which has seen relatively less research activity than the other levels.

3 The Demand for Assistive Technology for the Elderly

Demographic change in most developed countries is showing a burgeoning population of older people. The continued rapid growth in the population aged 80 and over is set to continue over the coming decades [8]. Since frailty and disability are increasingly common among the most elderly, this shift of distribution is resulting in greater demands for high-quality supportive care provision, particularly home-based or community care [26]. When this trend is combined with the declining numbers of younger people, who represent the future source of care workers, an alarming scenario is presented of a very acute mismatch of supply and demand.

Consequently, the value of augmenting community care with technological support has been widely recognized [26]. Such technology offers great promise for tools and techniques that can allow elderly people to compensate for physical and/or cognitive impairment and thereby reduce both handicap and demands on caring services [28]. Importantly, the goal of health-care is to maintain or improve *quality of life*, whether implemented by human *or* technical solutions [5].

From the published projections and our own analysis it is clear that the future health-care of the elderly will be dominated by several factors:

Premise 1 *There is a vital need to extend or supplement the available human resources required for the care of frail elderly people at home. Future hospital and home-care services will not be able to match the projected demand.*

Premise 2 *There is an urgent corresponding need to find technological solutions to help provide services that make up the projected shortfall in human resources.*

Premise 3 *To meet the predicted global demand, any successful devices or solutions must be widely available and be relatively low-cost. For any hardware-based aids this implies large volume mass production.*

Premise 4 *Assistive devices have additional challenges over those of level 0 and 1 robots, as they can not avoid complex spatial and/or physical interactions.*

4 Key Design Issues

Although a great deal of knowledge has been gained in industrial robotics, this technology has notably failed to transfer into the assistive field. This is mainly because industrial systems concentrate on precision, accuracy and repetition, and also assume a high degree of structure in the working environment. Together with high cost and relatively low task flexibility, these features are almost *an exact inverse* of the requirements for an assistive device for the home. Domestic environments are highly unstructured and very variable but the degree of accuracy and precision demanded can be quite low. Also, a high degree of adaptability is essential for both task execution and for flexibility across domestic tasks and environments.

This mismatch may explain why the early commercial offerings met with low acceptance and little demand. Despite early work showing the positive feasibility of assistive robotics and the high estimates of future markets, such systems seemed to be under-developed and have not sold well [13]. Considering the wealth of robotic technology available, this lack of penetration into the user market seems paradoxical, as it appears to offer great promise for solutions in the assistive field. However, there are many new factors and additional requirements that need to be taken into account.

Firstly, the needs of assistive users are often much more complex than those for industrial tasks. This is not always recognized by technologists and engineers who concentrate on product design but tend to ignore social and human aspects [23]. Secondly, it is quite common for new hi-tech products to be rejected or abandoned by users, for a wide variety of reasons. For every successful product there are many apparently attractive and desirable devices that are never taken up. This phenomenon is well known and the failure of early assistive robotics to penetrate the commercial health-care market is being examined [14]. However it is clear that many failures occurred because their designers attempted to adapt existing engineering solutions, designed for other purposes, to narrowly defined health-care problems [7]. Another factor is that the aim is not to *replace* an existing human or a human task, as is often the case with industrial robotics, but rather to support and enhance the user's abilities [9]. This is a much more complex situation and means that careful balances must be struck between the roles of users and devices [16].

Current research has shown that acceptance of assistive technology will require much more attention to user preferences than previously thought. Perhaps this concept of user *empathy* is the significant difference from industrial robotics [20]. Models of acceptability and socio-dynamic factors are now being developed [5] and one respected acceptability model argues that users' "felt needs" for aid or enhancement must be matched to the "quality" of the solution or device offered [16]. In this context, quality is defined in terms of efficiency, reliability, simplicity, safety, and cost. In addition, user variability is very high: "The heterogeneity of older people and the diversity of their living circumstances mean that individual preferences will play a strong part in people's attitudes" [16].

From such research we can observe further:

Premise 5 *Top design priority must be given to user preferences with emphasis on perceived quality and accurate understanding of needs. All technology issues are of secondary importance.*

Premise 6 *This implies that level 2 devices for the elderly must be customizable to accommodate different levels of task requirement and user abilities, and widely different personal environments. Such customization may involve the care givers whose needs must also be included.*

Premise 7 *Notwithstanding premise 6, such devices will also need to autonomously adapt* **in situ** *to environmental and task changes, including error events.*

5 Needs and Requirements

Consideration of assistive device scenarios show several very significant differences from other robotics applications. These include the following:

Safety – Assistive robots work in close proximity to humans, even sharing their operating envelopes. Thus, safety features must include collision avoidance and detection, and suitable analysis of user hazards and failure modes. System designs must cover issues such as: structural measures, e.g. support strength and low inertia components; safe operating regimes and fail-safe behavior; and dynamic sensing for proximity detection and environmental changes. We believe that software has a major role to play by contributing to overall system safety in all areas.

Costs – Acceptable costs are essential for global markets and low costs can be achieved by (a) using readily available components, (b) trading precision for robustness, and (c) reducing the complexity of mechanical design by transferring more functionality to sensors and software. Cost reduction is a design priority, and must be reconciled with any hi-tech solutions.

Autonomy – Robustness and high reliability are very important in systems that must run without technical support for long periods. However, assistive applications have so many uncertainties and variables that any given action can not be guaranteed to achieve its expected outcome or effect. Consequently, some errors and action failures are unavoidable. Hence robustness implies that errors should be accepted as events to be accommodated and so some form of automatic error recovery is essential for autonomous operation. This also introduces a need for the system to self-improve, so that repeated errors are detected and eventually anticipated and avoided.

Flexibility – Mass produced systems can be built to provide tools to help with configuration for each user's environment, but some customization by the user and/or carer will have to take place in the user's home after delivery. This means that such systems must be designed so that they can be personalized for each user's needs.

Usability – User requirements for the elderly are very demanding with very constrained modes of communication [2]. Any complexity must be hidden and systems must have very acceptable controls and operational modes. Any "user interfaces" must be minimal and inputs should be intuitive and responsive.

These requirements are very severe and more demanding than those found in previous robotics application areas. Much more research and innovation is needed before all the issues are solved and effective products are available and widely accepted. We can expect future advances in many areas but we believe the fundamental reconciliation of the engineering and economic requirements for high-volume production with the essential need for personal, customizable systems will be best achieved through intelligent software techniques, and this will be a vital factor for success in the high-volume markets of the future.

6 Addressing the Requirements

In order to investigate ways of tackling some of the above requirements we have experimented with robot control architectures and learning algorithms. Following the above observations, our approach is based on trying to satisfy the following:

Design Principle 1 *To keep costs low, much of the functionality of assistive robotics should be achieved through advanced software rather than hardware or mechanical systems. Software can compensate for low accuracy, coarse grained components, and other mechanical limitations, thus greatly simplifying the mechanical and hardware design of a system.*

Design Principle 2 *Software can also provide the added functionality demanded by these applications, particularly ease of use, robustness and adaptability.*

Design Principle 3 *Learning during use is essential. This will be necessary for customization, self-improvement during task execution, learning new tasks and learning about errors.*

Design Principle 4 *Robust operation in terms of tolerance of both environmental variations and action variability is essential. This means effective error treatment methods must be incorporated to allow autonomous operation and long periods of service without breakdown.*

Design Principle 5 *User communications for control or training must be very simple and direct. For example, very few keywords should be allowed and any tokens should closely relate to the objects in the task environment.*

Design Principle 6 *Any configuration and set-up stages may be performed by carers or users, and so any form of conventional programming input must be avoided. The input of data for configuration and/or new task variation could be by "teaching", by showing examples of what is required.*

We envisage a successful device that satisfies the needs of its elderly users will gradually accumulate experience over its life-time through incremental learning, and will adapt to the individuality of its user and to a variety of tasks to some degree. Assuming that there may be long intervals between carers' home visits implies that the system must not fail in an abrupt or catastrophic manner but should degrade gracefully. On error, it should try repeated attempts as the inherent variance in domestic environments means that key parameters are often not fixed and a second attempt may find more favorable conditions. Indeed, the experience of repeating a task gives valuable information for learning. Thus, error recovery is not an extra feature but a basic requirement and the experience captured, during *all* operations, should be utilized and made available for transfer to other tasks if at all possible.

6.1 Methodology

We have argued that future assistive devices, of level 2, will often have demanding general requirements, including long-term viability, low costs, ease of use, and flexible performance. Many such systems will be sufficiently complex that software will be the means to efficiently implement much of the functionality. We also argue that the human aspects are the overriding design concern (and a key distinction with conventional robotics) and so we propose a user-first, technology-next approach. In order to guide our case-study we used the following tentative methodology, summarized as a series of stages:

User situation analysis First, an analysis of the class of of tasks to be covered is needed, together with cost limits, the level of task support needed, and the level and nature of appropriate communication.

Task analysis This covers the design and analysis of a basic task set, defining the objects and/or actions involved, the variables to be sensed, any significant conditions, and maybe some indications of the main control functionality required.

Initial hardware design The identification of low-cost but effective sensors, actuators and other components then follows, taking account of the design requirements and associated error costs, performance and safety limitations. The software functionality will be broadly determined by the processing required by the sensors and other hardware.

Control function engineering This includes the analysis and design of the software functionality needed to control the sensors and hardware. The overall system architecture is designed, together with any special compensation software. Such compensation functions, e.g. self-calibration, should be self-contained, fast, and have any-time availability, so that they can be used on-demand and as frequently as necessary.

System design This stage covers iterations and walk-throughs with emphasis on finer detail and refinement. Experiments and prototypes help to evaluate designs and lead to further iterations around these stages.

Of course, to produce fully developed products for the future, all design stages will need multidisciplinary cooperation and input, combining expertise from across fields such as robotics, ergonomics, therapy and gerontology.

7 An Experimental Case-Study

In order to further examine and test the above design approach, a simulated case-study scenario was investigated and implemented. It is important to note that this was not a complete prototype implementation, with extensive user trials and full performance analysis. It is intended as an illustration of our approach and shows how it may be developed.

7.1 The acquisition task

First, a particular type of assistive home-care task has to be selected. Various studies of human analysis have dealt with different aspects of impairment but these tend to fall into broad categories as indicated in table 2. The area most demanding of assistance is stair climbing and descending. This is a level 3 task and stair-lifts and other such aids are now well developed. The next most pressing area is in help for reaching and bending activities [27]. This is mainly for accessing shelves and floors and has been rated highly by surveys [24]. We decided to implement an assistive task from this

1	Stairs
2	Reaching and bending
3	Sitting and walking
4	General mobility
5	Aids and adaptations
6	Domestic appliances
7	Care management

Table 2: Task areas in the home

general area and selected personal object localization and retrieval. The task consisted of transferring small domestic items between tables or other horizontal storage places as requested by the user. This task has wide applicability, it can be extended in scope and applicability, and it embodies typical human-centred design issues.

7.2 Design factors

The next step is to analyze the task and look for simplifications and extract the essential core requirements. The objects are to be small personal items, unidentified beforehand and of various shapes and sizes. The user wishes to identify desired objects by some means and the system should then find the desired item from various possible storage places, pick it up and transport it to a target location near the user. We notice that only one storage region need be implemented, as any other surfaces will only differ in their location. Also, we can assume, again without loss of generality, that the user will accept retrieved objects at a single prescribed target location. The required spatial movements of objects is thus the familiar "pick and place" operation with the user selecting the object. However, there are two key problems: *which* object is to be retrieved, and *where* is it located. Thus, we need a means of distinguishing objects and a means of determining their locations.

For the first problem, the user must communicate with the robot but there are many methods and modalities for verbal and non-verbal communication with the elderly. This is a cross-disciplinary area with much valuable input from human factors, ergonomics and health-care experts. However, it is clear that human-robot communication must be simple, natural and effective. Any tokens used in communication must not place any cognitive burdens on users and need to be unambiguously grounded in the application task. Following design principle 5, that user commands should be simple and minimal, a basic design could consist of tokens for <action, object> with optional <adjective> tokens for describing the object. For example, a user might request "*Fetch Red Pen*". The choice of action is simply either "*Fetch*" or "*Save*", corresponding to retrieval from or transport to the storage place. The object names are identifiers associated with particular objects by the user. For the adjectives we have selected color and shape as widely-used and accepted means of differentiating objects.

All the tokens in these commands could be input by many means, for example they might be spoken into a speech input device, or they could be entered by pointing at a touch sensitive screen.

For the second problem, finding where objects are located, we note that their general storage region is known but not their exact positions. As the objects may be of novel size and shape, there is no prior knowledge to simplify this task and so all the required object parameters must be sensed by the system. However, as the objects lie on horizontal 2D surfaces, we do not need six degrees of freedom to specify their position, but 3 variables, X, Y, θ, will suffice. These define the 2D position of an object's centeroid and its angular orientation, and need to be sensed for each object. These three variables will also be suitable to define locations for the start and end of robot actions. In fact, this spatial representation should include the height of objects (above the surface), i.e. a 2.5D space, but this is covered by a sensor as described in section 7.3.

Another design issue is safety. The robot needs to be protected from collisions with (a) other objects, and (b) with users. We do not address the full safety case for home-care robotics here but we made provision for a sensor to detect the height of objects to avoid collisions when approaching objects, and an emergency stop interrupt to prevent operation when users were too near the moving parts.

Table 3 summarizes the basic task specification in general terms.

7.3 Hardware aspects

The next step is to select some potential sensors and other devices that may satisfy the task needs. From the task description it is clear that vision sensing would be appropriate to determine the location, color, and size or shape of objects. Also, the required physical actions to be performed on the objects consists of acquisition and transport, to move the objects through space from one location to another. We used existing laboratory equipment to make up a testbed assistive robot to achieve these goals. All the hardware components were arranged to be coarse grained or of low precision. This is to test design principle 1 and to ensure that our results properly reflect the benefits of intelligent software compensating for the mechanical limitations of inexpensive (mass produced) hardware.

A Puma 500 manipulator arm was fitted with a two-fingered electrical gripper mounted via a flexibly

Type	Requirement	Comments
Objects	color, Size, Shape,	$\}$ All these must be sensed,
	Location(X, Y, θ)	(and vision can sense them all)
	Name	Given by user
Actions	Grasp object at (X_1, Y_1, θ_1)	Vertical approach to surfaces
	Move to (X_2, Y_2, θ_2)	Assume free space above surfaces
	Ungrasp	Low snatch required
Conditions	Avoid collisions — with objects:	Height sensing
	and people:	User interrupt
	Objects held fast during move	Sense object in gripper

Table 3: Main task elements

compliant wrist. A small, cheap, color CCD camera was mounted between the two gripper fingers, as can be seen in figure 1. In order to acquire objects the camera was arranged so that it viewed

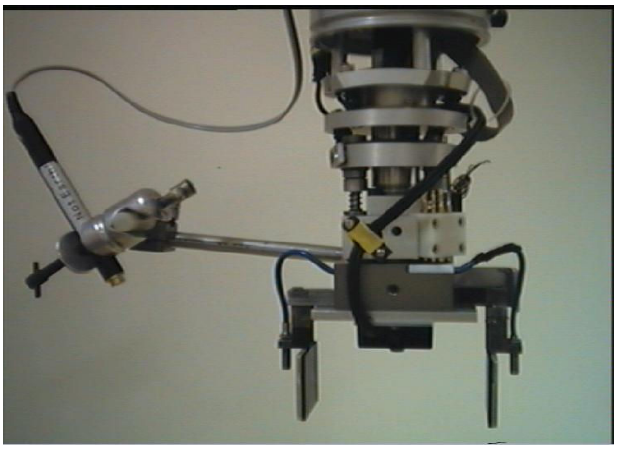


Figure 1: The gripper and sensors used in experiments

downwards towards the target region approached by the fingers when the gripper was lowered onto a table surface. Image analysis software then separated any small object in view from the background and computed the object centroid and the principal and minor axis directions. The center of the two fingers can then be aligned with the object centroid and rotated so that the finger axis is aligned with the object's minor axis. Because the camera focal point was coaxial with the gripper centre, this simplifies the grasping process enormously: when the object is centred in the image it is also centred in the gripper.

Also visible in figure 1 is a laser pointer which was used to determine the heights of objects above the supporting surfaces. The laser shines a spot of red light, at an angle, down onto the table. The position of the red spot in the camera image is related to the height at which it is reflected and this parallax displacement can be easily measured by a simple calibration routine. This additional sensor, which is necessary to detect large objects and obstructions, is thus created by gaining additional functionality from the existing vision sensor. Figure 2 shows this principle of height sensing

Any particular sensors or actuators will have software implications, for example, using images to measure object properties requires image processing software. We employed a commercial package for this purpose, and the robot and gripper were controlled by their respective propriety controllers. Other software demands are made by the task requirements and the design principles. These stipulate that the system must be driven by the user's requests, it must be adaptive in that it can recover from errors and learn from experience.

A summary of the decisions regarding the sensory-motor aspects and the consequent control and software requirements is shown in table 4.

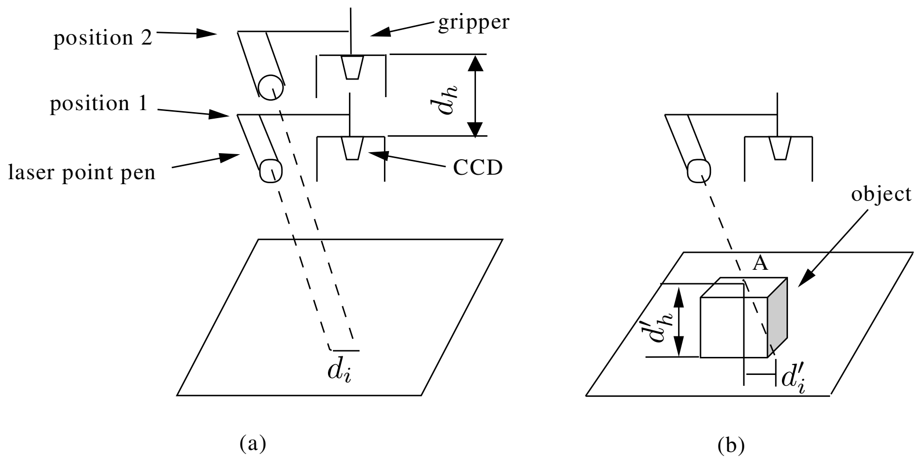


Figure 2: Object height measurement by laser pointer

Variable/Action	Sensor/Actuator/Features	Processing
color	Vision	color patch extraction
Shape	Vision	Extract axes, features and moments
Location	Vision	Extract centeroid and angle (X, Y, θ)
Name	Given by user	Database
Grasp object at (X_1, Y_1, θ_1)	2-fingered gripper	Vision aligned
Move to (X_2, Y_2, θ_2)	Manipulator arm	Low accuracy: ±1mm
Ungrasp	Electric gripper for low snatch	Conventional controller
Avoid collisions	Height sensor	Vision via laser spot
	User interrupt	Simulated, by experimenter interrupt
Detect dropped objects	Gripper sensor	Simulated, by experimenter interrupt
Control	Demand driven	Depends on: user request, current state, relevant experience
	Error recovery (repeat actions)	Requires experiential learning
	User tolerance	Allow variance in commands
	Environmental tolerance	Allow variation in objects
	Extensions to new tasks	Teaching by showing

Table 4: Proposed hardware and associated processing

7.4 Control features

The effect of design principles 1 and 2 on the analysis is now seen in the simplicity of the hardware and the demands made in the software requirements. Further analysis of the software identified several separate areas of functionality, as follows: color processing, the integration of user and vision data to locate target objects, the design of the main control architecture, the use of virtual movement simulation in image space, and the means of action selection and error recovery. We now explain each of these in turn.

7.4.1 color processing

Unfortunately, humans can only provide naive color names rather than precise color specifications, so some form of conversion is needed to convert human perceived object color names into a color space which robots can understand. Although a form of color histogram may be the most accurate way of describing a color distribution, it is not practical to ask human users to input this kind of information in real applications. A mapping between natural color names and a specific color space is therefore needed. There are many possible color spaces, for example, RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), Munsell, YUV, etc. We selected the HSV color space since this space is nearly perceptually uniform, i.e. the similarity of two colors is closely related to their proximity in the HSV color space.

The image captured from our camera was coded in the RGB color space, so the first step was to implement a conversion algorithm to map RGB into HSV. Next, we derived color names from the ISCC-NBS color system [11] which uses a standard set of base hue names together with a set of hue modifiers. Our implementation mapped the incoming color space into a large number of bins, each of which were assigned a color name in the ISCC-NBS color system according to a nearest distance criteria in the space. After creating the mapping relationship, this can then be used as a look-up table to convert new data.

During object recognition, we use both the color names provided by users and their variants in the ISCC-NBS color system. For example, if a user wants a green object, the system will search not only green, but also light green, moderate green, deep green, vivid green, etc. Each image pixel is converted and compared with the color bins of the target color and its variants to see whether this pixel belongs to the target object. If the percentage of the number of pixels with the target color, or its variants, in the overall number of object pixels is greater than a threshold, then this object is regarded as the target object by the color checking process.

7.4.2 Integration of user and vision data to locate target objects

Vision can be a very powerful sensing modality, but often incurs high costs associated with high bandwidth data and complex computations. In line with our design philosophy, we employed a low performance sensor and must therefore allow for low quality in our visual data. Consider the typical user request example "*Fetch Red Pen*", where *Pen* is an arbitrary name, that may have been previously assigned to an object or object class. Note that if the pen is already known to the system its location is also likely to be known (recorded) and the robot can proceed immediately to retrieve the object. However, if the object's location is unknown then we require some search and recognition algorithms to find it. Furthermore, if a red pen is unknown but other pens have been seen then the system can use any existing shape descriptors for *Pen* together with the specified color to guide the search for a matching object. Finally, if the word *Pen* is unknown then the system searches for items whose color matches *Red*; the closest matching item is presented to the user, which, if accepted, allows the name *Pen* to be associated and stored with the shape and location data. A simple database was used to record the known data on objects with entries for <object name, color, shape features, location coordinates>.

In designing an appropriate visual object recognition system we examined various feature extraction methods. We rejected local salient point based methods in favor of global feature methods, mainly because global features are more useful as general descriptors of new objects. We combined simple object shape features with user color information to provide a fast and effective object recognition scheme. The shape features used are, eccentricity, compactness and moment invariants. We examined two image segmentation techniques, an edge-based method and a histogram-based method. All segmentation methods have their own various advantages and disadvantages, and we found the histogram-based method to be superior in its tolerance to lighting variation and this was used in all our experiments.

A specific behavior module (section 7.4.3 describes behaviors) was developed to perform the data acquisition and object recognition process described here. This method relates the shape information derived from observation with the user's declared color and name for an object. Consider the "*Fetch Red Pen*" scenario again. The first time an object is seen the system will save its shape descriptors in the database. If a color can be associated with the object then this will be saved too, as will a user given name, e.g. "*Pen*", if possible. Then in later searches for a given object, if the object is known, the name, color and shape entries can be used to match and locate the desired item. When the robot is asked to search for a target object, if the object's position is known from previous operations then this positional information is used directly to locate the target object, otherwise, a search process is triggered. For each image taken at each camera position, objects are first separated from the background, and then, for each object, its color information is checked to see whether this object satisfies the color criteria of the target. In this way, user provided information is given a higher priority than information obtained from the vision system. Thus, if there are no objects in the current scene that match the features provided by the user, then other feature processing will be inhibited and attention will move to other areas to search. Figure 3 illustrates this scheme.

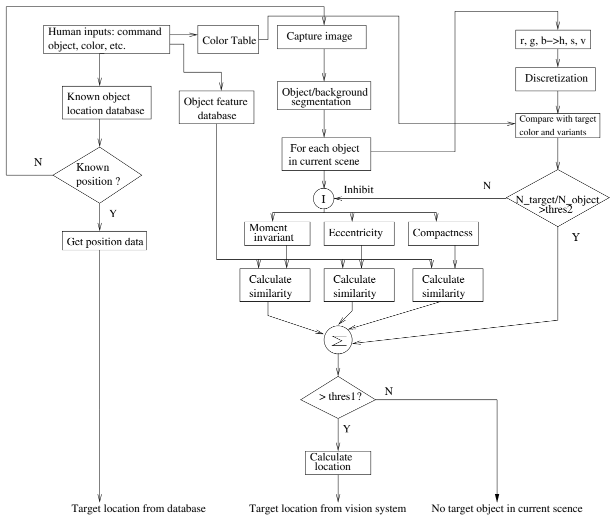

Human inputs: command object, color, etc.
Color Table
Capture image
r, g, b–>h, s, v
Discretization
Known object location database
Object feature database
Object/background segmentation
Compare with target color and variants
For each object in current scene
N
Known position ?
Y
Get position data
I
Inhibit
N
N_target/N_object >thres2
Y
Moment invariant
Eccentricity
Compactness
Calculate similarity
Calculate similarity
Calculate similarity
> thres1 ?
N
Y
Calculate location
Target location from database
Target location from vision system
No target object in current scence

Figure 3: Part of the **Search_Object** behavior

When searching for objects on a surface the robot coordinates need to be correlated with the image coordinates. In the general case, this is a difficult issue because robot space is 3D and image space is not only 2D but usually has perspective and other distortion effects. We simplified this problem by noting that the camera is only used to take images vertically downwards onto the work surface. This means we can map the 2D image space onto the 2D coordinates of the surface as known to the robot. This was done by implementing an auto-calibration process which operated by tracking certain points in the image and observing the changes produced by small movements of the camera (robot). A coordination coefficient matrix is then produced from this data by a standard algorithm. This process produces only an approximate image-robot mapping matrix but it has various advantages: the method is simple, efficient and fast, and this allows it to be executed at any stage during a task.

If the robot moves to a new surface or other non-local changes occur then the auto-calibration routine can be quickly executed again to refresh the coordination mapping.

7.4.3 The behavior-based architecture

Reactive systems give some attractive advantages, including robustness and real-time performance. In particular, the behavior-based approach [1] maps sensor information directly to actions and so avoids the overheads and complexity concomitant with planning, world models and representations. But pure behavior-based systems have inherent limitations that rule out learning and planning and so such systems are usually not able to accomplish complex tasks, find it difficult to conduct tasks different from those in-built, and can not assimilate data learned from experience. A different approach, which we favor, is to embed representation into a behavior-based system, as in the method presented by Matarić [15]. Our system builds on this successful approach for integrating representation into behavior-based systems.

Rather than apply learning to the lower level of behaviors, i.e. perform modification of the condition/action relationships, we focus on a higher level, that of learning how to use existing behaviors. The aim is to allow behaviors to be used in new ways, to reuse the knowledge inherent in one task in other later tasks, and to allow tasks to be completed more efficiently by learning different action sequences. We provide behaviors with high level descriptions modeled in a *preconditions-behavior-effects* form. This is to support reasoning about the results of behaviors without their actual execution, and to facilitate the features described above. Figure 4 illustrates this structure. In addition to the basic behavior module, which couples sensory inputs to action, information is held about the behavior's preconditions and expected results, its previous experience, and a pointer to an automatic error recovery module. The error recovery module is triggered when the behavior has not achieved the expected effects.

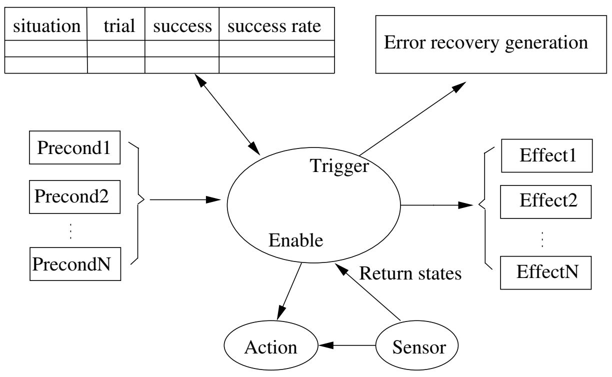


Figure 4: The structure of behaviors

Whenever a behavior is used or reused the experience embedded within it will be accessed and updated according to the result of execution. Experience records are designed to be context related, so the same behavior may have different experiences for different situations. This distributed embedding of previous experience is a key feature for action selection and learning in our system. Table 5 shows an example of some experience records used in our experiments. An experience entry has five elements: the behavior name, the conditions under which the experience was gained, the number of times it has been tried, the number of successes, S, and the success rates. In this example the conditions refer to possible obstructions near an object about to be moved.

The method of computing success rates from the raw data needs careful design — simple success/failure ratios will not suffice. We selected a tried and tested confidence interval estimation algorithm from the literature [10] that produces a success rate measure, given a set of trials and their successes. This method can give a behavior a high success rate either because it is known to be a good behavior to take under the current situation *or* because very few trials have been executed under the

behavior	Constraint conditions	Trials	S	S rate
Put_Object	Clear	85	85	100%
	Not clear	5	0	43.45%
Side_Push	Clear	0	0	100%
	One-sided constraint	17	17	100%
	Bilateral constraints	24	2	25.85%
	Two-sided constraints (not bilateral)	17	15	96.71%
	Three-sided constraints	13	1	33.31%
	Four-sided constraints	0	0	100%

Table 5: Examples of experience records

context, (before any trials have been performed the success rate = 1). In this way, the system can rule out actions with very low likelihood of success and try other behaviors. A parameter, α, affects how many trials are needed before estimating whether a behavior is good or not in the current situation. Intuitively, the smaller α is chosen, the more trials are needed to explore, since more experiences are required to drive the success rate down. In the limit, when $\alpha \to 0$, the system becomes completely exploratory and all behaviors are treated as equally good. At the other extreme, when $\alpha \to 1$, the system becomes completely exploitational, since the success rate approaches the raw empirical probability of the behavior success. In our experiments, we chose $\alpha = 0.05$.

We now list the behavior set used in the experiments. Some of the behaviors are basic behaviors, while others are compound behaviors which encapsulate other basic behaviors and their dependencies. Table 6 gives the name and index number of each behavior together with the preconditions and effects used for higher level control.

Index	Name	Preconditions	Behavior	Effects
0	Center_Object	Object is in the current image	Align center of object with center of image	Object centered
1	Approach_Object	Object centered	Descend towards object taking account of height from laser	Object is between the gripper fingers
2	Move_Object	Object in gripper and at free height	Move object to defined location, error recovery procedure triggered if object leaves gripper	Object is above the target location
3	Open_Gripper	Gripper is closed	Open gripper	Gripper is open and no object in gripper
4	Close_Gripper	Gripper is open	Close gripper	Gripper is closed
5	To_Free_Height	Not at free height	Robot moves to the table clearance height	Gripper at free height
6	Move_To	None	Robot moves to a given location	The gripper is at the target location
7	Search_Last	At free height	Search for similar object in last used area	Object is in the current image
8	Search_Object	At free height	Search for object with desired color and shape	Object is in the current image
9	Lift_Object	Object in the gripper	Lift the grasped object to free height	Object at the free height
10	Put_Object	Object in the gripper	Place object at the target location	Object at the target location
11	Grasp_Object	Object is between the fingers	Close the gripper fingers to grasp an object	Object in the gripper
12	Side_Push	Object in the gripper	Place object *beside* other(s), put fingers *near* one side and push towards target location	Object at the target location
13	Observe_Pattern	None	Observe user's object pattern, extract and store relevant information	Example observed
14	Check_Result	Example observed	Checks the effects of Put_Object and Side_Push for success	Obtain discrepancy between action result and desired
15	Imitation	Example observed	Imitates the object pattern shown by the user or previously stored	Demonstrated example is reproduced

Table 6: The Behaviors Repertoire

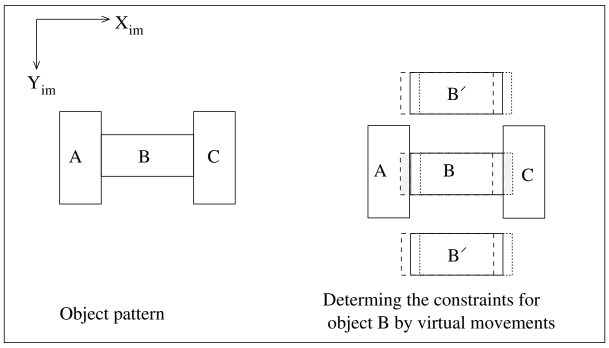


Figure 5: Virtual movements find proximity constraints

7.4.4 Object constraints from virtual movement in image space

When the robot gripper is to approach an object it is important that there is enough clearance to avoid collision with other nearby objects. We devised a method based on virtual movement in the image space to extract data on any action constraints imposed by the proximity of nearby objects. Figure 5 illustrates this process. For each object to be moved, we shift the object, in image space, along its principal short and long axes while keeping other objects fixed, and check whether this object collides with other objects. The existence of a constraint is decided according to how many pixels are in the overlay area of two objects after the virtual movement. Allowing for noise at the edges, a threshold is used to judge whether a constraint really exists. Virtual movements in image space usefully identify objects that may constrain a behavior and determine the direction in which the constraint operates. This algorithm provides conditional data that is used in planning and selecting appropriate actions.

7.4.5 Action selection and error recovery

The method of action selection, illustrated in figure 6, is based on Maes's activation spreading network [12] modified by the inclusion of experiential data and a behavior "difficulty" factor. This module generates an action by backtracking from the behavior whose effect is the goal to be achieved to a behavior whose effects match the preconditions. Only one behavior is backtracked at each step, so we can always find the nearest behavior to the goal. By avoiding planning and executing multiple actions as a group, the system is able to handle failures or unexpected events that occur at any stage during the execution process.

```
1: Find the behavior(s) whose effect is the goal,
   and set activation level.
2: If the behavior's preconditions are not met,
   spread activation back to behaviors whose
   effect(s) are the preconditions.
3: Multiply this activation level by the success
   rate and the difficulty factor of the behavior.
4: Check if the current behaviors' preconditions
   are met, and the activation level is higher
   than a threshold, if yes, go to step 5; if not,
   go to step 2.
5: Return the behavior with the maximum
   activation value.
```

Figure 6: The automatic action selection process

In order to incorporate both the history of the success of the behavior and choose a relative low cost action, the spreading activation level is multiplied by the behavior's success rate and its difficulty factor. Difficulty factors for each behavior are entered at design time according to knowledge of the activities involved. For example, in our system, behavior **Side_Push** and behavior **Put_Object** have the same preconditions and effects, but **Side_Push** has a higher cost because it is more complex to perform, see figure 7.

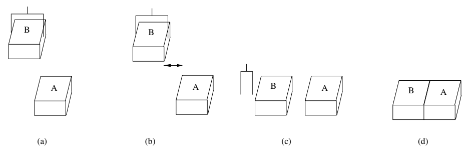


Figure 7: Illustration of behavior *Side_Push*

If one behavior's success rate is low, then successor behaviors on the same activation path will only get a small activation level, so the final action selected may be one on another path which is more reliable. However, in that case, the robot may then need to perform more actions before achieving the final goal than with the lower reliability path. Thus, action selection involves a tradeoff between cost and reliability.

Behavior-based approaches have difficulty with complex error situations, i.e. those where a sequence of several actions are needed for recovery. Such situations are often handled by the planning modules in hybrid deliberative/reactive systems. By embedding error recovery strategies into each behavior, our system generates appropriate recovery action only when required and as governed by current circumstances. The experience gained in error recovery is also stored in relevant behaviors, and thus future error recovery action selection is adjusted by this experience. For example, in executing **Move_Object**, the robot will stop immediately when the object is dropped from the gripper and the error recovery module is triggered. If the dropped object is still in the center of the image, the first error recovery action to be performed is to approach the object, using **Approach_Object**; if the object is in the image, but not centered, then the error recovery action will be **Center_Object**; and if the object is not in the current scene of the camera, then behavior **Search_Object** will be selected.

8 Experiments and Results

8.1 Single object retrieval

The first experiment tests the basic object retrieval process using single items recognized and retrieved by their color and shape. A range of objects were shown to the system in a designated search area, as seen in figure 8.

We asked the robot to find an object by giving the command: "*Fetch Pink Pencil*". The pencil in the center-right of the image in figure 8 is pink. This experiment was implemented as the highest level behavior **Fetch_Object** (not listed) which passes the key words "*Pink*" and "*Pencil*" to behavior **Search_Object** which, if successful in locating a specific object, then calls a sequence of **Move_To**, **Center_Object**, **Approach_Object**, **Grasp_Object**, and **Move_Object** behaviors to deliver the object to a designated user receiving location.

Although several pixels of other objects were also regarded as pink their number was very small and only one object gave a good match (1134 pixels out of 1254, giving 90%). The color matching candidates are then tested against any stored shape measures for "*Pencil*". In the case illustrated the shape match is 88% and the pink pencil is correctly identified and located. Notice that color has a higher priority than shape so that failure to match color prevents any computational effort on shape analysis. Over a series of trials, the success rates averaged 96% for complete retrieval of the selected items.

The object database contains records with entries for <object name, color, shape features, location coordinates>. If a name, e.g. "*Pen*", has not been encountered before then, provided the selected object is accepted by the user, a new record will be created containing the name value, color, shape

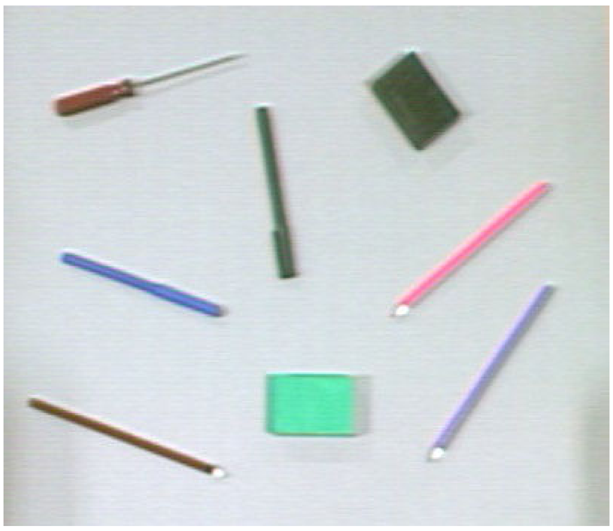


Figure 8: A range of objects in the search area

and location data. Thus, on future occasions the object may be retrieved directly by name. In this way it is easy to teach the system new objects at any stage, even if other objects are visible in the search area.

The system still functions usefully if some of the database records are incomplete because entries are made whenever missing data are discovered. For example, if an object has a distinguishing color then the user need not even mention a name; "*Fetch Pink*" will be sufficient to select the object dominant in that color. The shape and location parameters will be computed and entered, and the user can be asked to supply a name. Alternatively, shape features can also be used to select objects by commands like "*Fetch Pencil*" if the name has already been used. A shape matching threshold is needed (we found that our pencils matched a pen model with a similarity comparison of 70%) and if several candidates are produced then color can be used to select further.

Our system uses a mixture of user provided information, names and colors, with extracted visual properties, color, shape, location. We have seen that color should have a high priority in object description, because (a) this can reduce the computational costs involved in object search and recognition, (b) it is simple and natural for users, and (c) it is a relatively robust measure.

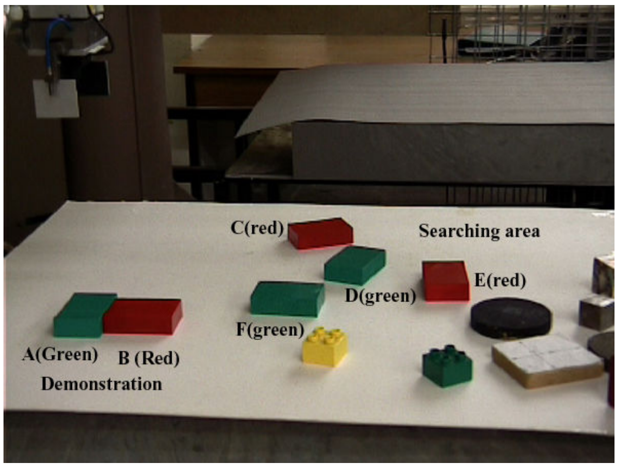
8.2 Error recovery

In the second set of experiments, we tested the ability of the system to automatically recover from errors using existing behaviors and demonstrate how learned experience can influence the activation spreading mechanism to adjust the error recovery process.

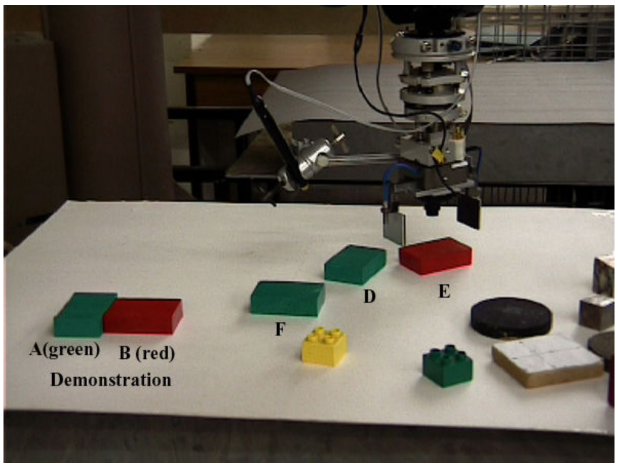
All the following experiments used a common framework in which users communicate with the robot by showing examples of desired objects or locations. This framework subsumes the "*Fetch Red Pen*" task and shows how error recovery was performed in such tasks. It also extends the scope to include problems with local proximity of objects and spatial patterns of object configurations.

The test scenario is as follows. The workspace had three general work areas which were made known to the robot: a *search area* where a range of objects might be found, a *demonstration area* where arrangements of objects were created by the user and then observed by the robot, and an *imitation area* where the robot was to reproduce the arrangements seen in the demonstration area. To begin, a user arranges a pattern of colored objects in the demonstration area and then the robot moves to this area to observe the pattern. Data is extracted on object shape and color, spatial relations between objects in terms of their relative positions and orientations (using their centroids), and also any constraints on physical proximity as described in section 7.4.4. The robot then moves to the search area to find matching objects of appropriate shape and color. A suitable object is then grasped, moved to the imitation area and placed according to the pattern seen during the observation phase. Finally, the created pattern is checked to see if all the objects have been copied; if not then further objects are found and placed.

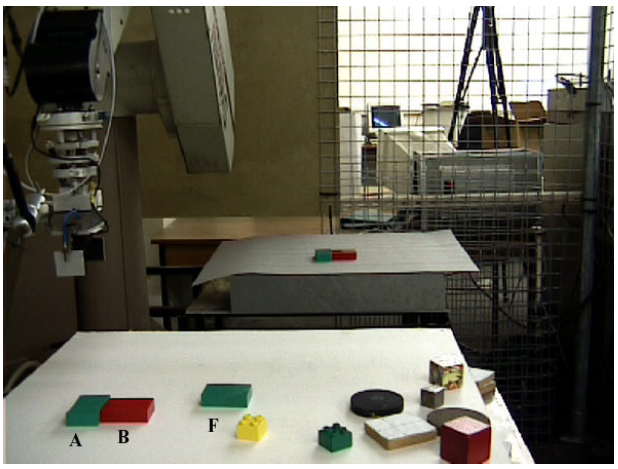
On one table, a human showed the robot a two object pattern as seen in the left corner of figure 9(a), (green object A and red object B). After observing, extracting and analyzing information, the robot imitated the pattern on a second table as seen in figure 9. During the imitation process, the robot first searched on table 1 for a target object with the same color and shape as the object in the demonstrated pattern (object C), then grasped it and moved it towards the other table. During transport, the object dropped from the robot gripper when the gripper was either above one of the tables or between the two tables and an automatic error recovery process was triggered.



(a) Initial state



(b) Re-grasp the dropped object



(c) Final state

Figure 9: Automatic error recovery action generation

Without any relevant experience, the robot searched the area where the loss had occurred and could not successfully re-grasp the dropped object either because it could not find it or it could not approach it for grasping. During the next phase of the error recovery process, behavior **Search_Last** was automatically selected by the activation spreading module, and used to search for a similar object from the place where the robot had acquired the object. The robot found and grasped another red object and then moved this, object E, to the target location. While the robot was moving, it dropped the object again, but this time above table 1 rather than between the two tables. The robot searched again, and retrieved the dropped object — figure 9(b) shows the moment of retrieving object *E*. This was successfully grasped and placed on table 2 and then, finally, actions were selected automatically for the second object, which was not dropped from the gripper. Figure 9(c) shows the final result with the robot's constructed pattern on table 2 being similar to the shown example on table 1.

We see that action selection is not only related to the goal to be achieved, but also depends upon the current situations and the available experience. To be selected, a behavior must both have its preconditions satisfied, and have the highest activation level due to its combination of previous success and relative difficulty or cost.

8.3 Experience learning and reuse

Some further experiments explored how behaviors may reuse experience gleaned from previous tasks. There are two behaviors that can be used to place an object, **Put_Object** and **Side_Push**, which both have the same preconditions and effects. However, in the absence of any experience **Put_Object** will be preferred as it has been given a better difficulty rating. Given enough free space, the robot will successfully place objects using **Put_Object**. But if a collision occurs while using **Put_Object** then an error is raised and triggers the automatic error recovery procedure, which then selects the behaviors **Lift_Object**, **Move_Object** and **Side_Push** to recover from the error state. After the collision, the activation level of **Put_Object** changes to be lower than that of **Side_Push**, which is then selected for the error recovery second attempt. We note that **Move_Object** was selected because one of the preconditions of **Side_Push** is that the object is above the target location, this automatically chosen action ensures that the object is in the correct position before the placement.

The trials immediately following showed the benefits of this learning: when a second object is in close proximity to the first placed object, the activation spreading module selects the **Side_Push** behavior directly instead of trying to repeat the same error as in the first trial. Note that the difficulty factors for competing behaviors, such as **Side_Push** and **Put_Object**, influence how many times the system will try one before choosing the other.

Further experiments explored the reuse of experience to determine action sequencing. In single object retrieval there will often be plenty of clear space available, but in general the sequence in which a series of objects are placed may affect the overall success rate. These experiments investigate how experience can be used to select the best sequence for imitating a task. For full details of all the experiments described here see [17].

9 Discussion

Our work has integrated a number of techniques into a system that directly addresses a specific user task scenario. The reasons for our selection of particular software methods and techniques should now be clear — they emerge as a consequence of the requirements, which in turn, are grounded in the premises and design principles which we argue are important for assistive devices for the elderly. We have demonstrated how task analysis has led to an experimental system with a wide range of features. Perhaps the most notable aspects of the system are that it integrates a number of functions to achieve flexibility, and it uses learning technology to gradually learn from errors and experience and then reuse its experience later in other tasks.

A behavior-based architecture is ideally suited for demand-driven autonomous systems that must respond to the needs of the environment as and when required. Fixed sequence programming is not possible in these situations and deliberative planning has limited scope. Our approach provides support for reasoning about actions while retaining the reactivity of a behavior-based system, but without the sharp separation of function which often causes awkward problems for hybrid architectures. A key feature of our architecture is its distributed, context-based design which supports the reuse of behaviors and their exploitation in new tasks. Being embedded into the behaviors, experiential data becomes related to the actions/goals of the behavior and the prevailing context. Simple planning can also be performed using the representations contained within behaviors.

The learning aspects of our system reside in the adaptation facilities, provided mainly through the activation spreading mechanism. This underpins action selection, using cost and success factors, which also drives both error recovery and the capture of experiential data. Action selection is automatic and changes with experience and the properties of the behaviors available.

We have resused existing techniques wherever possible but these have often been modified as required by the task demands. Apart from the integration aspects, the novel features of our system include: prediction through virtual movement in image space; the use of fast, on-demand, self-calibration of various system components; the adoption of human-centered values, namely color and shape; and the use of similarity measures in imitation and teaching by examples.

Many low accuracy or coarse-grained components were used to illustrate how hardware costs could be reduced and also demonstrate how software techniques can provide compensation. This showed how quality, in terms of flexibility and reliability, can be achieved at the system level even with lower quality, low-cost components.

9.1 Vision

We believe vision is such a powerful modality that it can be used to solve many communication problems. Other modalities are being actively investigated for use by the elderly, e.g. voice, hand gestures, touch controls etc, but the visual channel is particularly rich for developing new user-robot interaction and communication techniques. For example, if an image of a work area is shown to a user on a touch-screen then an object can be named/confirmed/selected by a simple indication, without any use of commands or textual processing. We have shown how low-quality image processing is quite sufficient to extract enough data and can be developed to provide a valuable communication channel. Careful selection and utilization of low bandwidth visual data may have great value if properly integrated into assistive devices.

Another role for vision is in “showing” new tasks or objects to a system - this can be particularly appropriate when configuring, by a user or carer, for individual circumstances. We have previously used such teaching methods successfully in other applications with restricted user input [29]. By highlighting the differences between the desired and the actual results, a learning system can successively correct its errors until the desired result is achieved.

9.2 Managing complexity

If we now review the design decisions that were made for the case-study we see how the approach has tended to reduce or control the complexity inherent in the application task. Table 7 lists some of these decisions and their consequences.

Feature	Comments	Software implications
Camera-in-Gripper	Reduces coordination complexity	Simplifies grasping cycle — no sense-then-act sequence
Laser spot projection	Saves using extra sensor	Via extra software functionality
Auto-correlate image/robot space	Rapid, anytime facility	Fast software procedure
Experience gathering	Enhances performance, reuse and recovery	Requires cumulative learning
Virtual movement testing	Prevents collisions, improves planning	Via image processing
behavior architecture	Robust, demand/goal driven	Additions for error recovery and learning

Table 7: The role of software and other methods in reducing complexity

Some apparently simple choices can have significant implications. For example, the location of a camera is likely to influence the difficulty of the required image processing, as corrections for perspective, distortion, registration etc may be required. But by using the camera-in-hand technique much of this is eliminated as the two spatial frames of reference (camera and robot) are permanently fixed and aligned.

The use of simple and well-tried sensors and other equipment is also beneficial and many performance improvements can be made in software. The trade-off between hardware and software must be decided for individual situations but modern software can be made very powerful, very reliable, and cheap to reproduce. Software can also offer extensive functionality. Thus, when a choice exists between mechanical or software design, the latter now offers a real alternative and deserves serious consideration.

We believe this complexity reduction is one of the main benefits of our approach and will become increasingly important in the design of assistive systems. Many alternative choices could be made in our case-study but the main principles and resultant outputs would still cover the significant task requirements. For example, our grasping technique for acquiring objects is very simple and we could have implemented one of the many complex grasping methods on offer [25]. However, all such methods are prone to fail occasionally (with widely varying personal objects) and so the requirements dictate that computational effort is better spent on error treatment. Thus, there will be many such (technological) variations but these must be driven by the human requirements, not the other way round.

10 Conclusions

This study has argued that conventional robot technology will not be sufficient for new assistive robotic applications unless user *acceptability* is given top design priority. This is different from the usual objective models of requirements seen in, for example, industrial robotics, because human roles or functions are not *replaced* by assistive devices but are enhanced or supplemented. Assistive robots will work in close cooperation with their human users and care givers. This means new design factors such as user preferences and subjective perceptions, in short *empathy*, must be considered as important and highly influential.

Many new social models of acceptability in assistive technology are now being developed [16] and these examine the interactions between user's physical and mental capacities and their environments. The aim of such models is to improve and understand acceptability by matching subjective felt needs to more objective qualities such as efficiency, reliability, simplicity, safety, and cost. These are exactly the attributes that have been raised through our needs analysis.

Clearly, our laboratory testbed does not constitute a full prototype for an assistive device — but that was not its purpose. The experiments demonstrate how the design objectives may be realized in functioning systems that satisfy key criteria, and show the way forward. It should be noted that the behavioral library can be expanded to cater for other tasks and the design will scale up easily as the behaviors are designed for particular actions and contexts. Two main aspects of the implementation are the combination and integration of a range of existing techniques, which did not involve any new technology although several novel modifications were produced; and the use of an architecture that provided the right level of complexity to support the integration in an extensible and flexible substrate. Any future continuation of robotic reaching aids will need to examine further issues including dynamic performance, effect on social dynamics, psychosocial issues, trials with user groups, and full economic assessments.

Autonomous systems, by definition, must have their own goals, but we view the user as the focus of those goals — that is, the user can set, modify and alter the system's goals at the top level. Taking this mixed approach leads to systems that should prove more acceptable to frail or impaired users without compromising on functionality. We have seen that learning is crucial for such robots because they must adapt to different tasks and environments. Consequently, learning technology and other

Artificial Intelligence methods are likely to have a major role to play in the future of assistive robotics.

In response to the impending need, we see that assistive robotics offers real promise of new devices that will support and increase the independence of frail elderly people living at home.

References

[1] Arkin R.C., Behavior Based Robotics, MIT Press, 1998.

[2] Batavia A.I. and Hammer G.S., Toward the Development of Consumer-Based Criteria for the Evaluation of Assistive Devices, J of Rehabilitation Research and Development, 27, pp425-436, 1990.

[3] Buhler C. and Knops H. (Eds.), Assistive Technology on the Threshold of the New Millenium, IOS Press, 1999.

[4] Cortes U. *et al*, Assistive technologies for the disabled and for the new generation of senior citizens: the e-tools architecture, AI Communications, 16(3), pp193-207, 2003.

[5] Cowan D. and Turner-Smith A., The role of assistive technology in alternative models of care for older people, in [26], pp325-46, 1999.

[6] Dallaway J.L., Jackson R.D. and Timmers P.H.A., Rehabilitation Robotics in Europe. IEEE Transactions on Rehabilitation Engineering, 3, pp35-45, 1995.

[7] Fehr L., Langbein W.E. and Skaar S.B., Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey, Rehabilitation Research and Development, 37(3), 2000.

[8] Grundy E., The Epidemiology of Aging, in: Tallis R.C., Fillit H.M. (Eds.) Brocklehursts Textbook of Geriatric Medicine and Gerontology, Churchhill Livingstone, pp3-20, 2003.

[9] Hoenig H., Taylor D.H., and Sloan F.A., Does assistive technology substitute for personal assistance among the disabled elderly? American Journal of Public Health, pp330-37, 2003.

[10] Kaelbling L.P., Learning in Embedded Systems, MIT Press, 1993.

[11] Kelly, K. L. and Judd, D. B., The ISCC-NBS Method of Designating colors and a Dictionary of color Names, National Bureau of Standards Circular, 553, 1955.

[12] Maes P., Situated Agents Can Have Goals, Robotics and Autonomous Systems, 6, pp49-70, 1990.

[13] Mahoney R.M., Robotic Products for Rehabilitation: Status and Strategy, Int. Conf. on Rehabilitation Robotics, University of Bath, UK, pp1-6, 1997.

[14] Martin B. and McCormack L., Issues Surrounding Assistive Technology; use and abandonment in an emerging technological culture, in [3], pp413-20, 1999.

[15] Mataric M. J., Integration of Representation into Goal-Driven Behavior-Based Robots, IEEE Transactions on Robotics and Automation, 8(3), pp304-312, 1992.

[16] McCreadie C. and Tinker A., The acceptability of assistive technology to older people, Ageing and Society, pp91-110, 2005.

[17] Meng Q., Control and Learning in Behaviour-Based Domestic Service Robotics, PhD thesis, University of Wales, Aberystwyth, 2003.

[18] Miskelly F.G., Assistive technology in elderly care, Age and Ageing, pp455-58, 2001.

[19] Mittal V.O., Yanco H.A., Aronis J. and Simpson R. (Eds.), Assistive Technology and Artificial Intelligence, 1458, Lecture Notes in AI, Springer-Verlag, Berlin, 1998.

[20] Pape T.L-B., Kim J. and Weiner B., The shaping of individual meanings assigned to assistive technology: a review of personal factors, Disability and Rehabilitation, pp5-20, 2002.

[21] Rehabilitation Engineering and Assistive Technology Society of North America, http://www.resna.org/

[22] Rofer T., Lankenau A. and Moratz R., Sevice Robotics - Applications and Safety Issues in an Emerging Market, Workshop W20 in proc. ECAI'2000, Berlin, 2000.

[23] Sixsmith A. and Sixsmith J., Smart technologies: meeting whose needs?, Journal of Telemedicine and Telecare, pp190-2, 2000.

[24] Stanger C.A., Anglin C., Harwin W.S. and Romilly D.P., Devices for Assisting Manipulation: A Summary of User Task Priorities, IEEE Trans. Rehabilitation Engineering, 2(4), pp256-65, 1994.

[25] van der Stappen, A.F., Wentink C., and Overmars M.H., Computing immobilizing grasps of polygonal parts, Int. J. Robotics Research, 19(5) pp. 467-479, 2000.

[26] Sutherland S., With Respect to Old Age: A Report by The Royal Commission on Long Term Care, The Stationery Office, UK, 1999.

[27] Tinker A., McCreadie C., Turner-Smith A. and Blake P., Older Users Perspectives on Mobility-Related Assistive Technology Research, Kings College, London, 2000.

[28] Verbrugge L.M., Rennert C. and Madans J.H., The great efficacy of personal and equipment assistance in reducing disability, American Journal of Public Health, 87, 3, pp384-92, 1997.

[29] Williams T. G., Rowland J.J., Lee M.H. and Neal M.J., Teaching by Example in Food Assembly by Robot, Proc. 2000 IEEE Int. Conf. On Robotics and Automation, San Francisco, pp3247-52, 2000.