



**KTH Numerical Analysis
and Computer Science**

Department of Numerical Analysis and Computing Science
TRITA-NA-P0110 • IPLab-186

Användaranpassad utformning av en "fetch-and-carry"-robot för funktionshindrade i arbetslivet

**Kerstin Severinson Eklundh, Erik Espmark, Anders Green,
Helge Hüttenrauch, Mikael Norman and Lars Oestreicher**

Interaction and Presentation Laboratory (IPLab)



NADA is a co-operating department between the
Royal Institute of Technology and Stockholm University.

Kerstin Severinson Eklundh, Erik Espmark, Anders Green,
Helge Hüttenrauch, Mikael Norman and Lars Oestreicher
*Användaranpassad utformning av en "fetch-and-carry"-robot för
funktionshindrade i arbetslivet*

Report number: TRITA-NA-P0110, IPLab-186

Publication date: May 2001

E-mail of author: {kse, green, hehu, pnorman, larsoe}@nada.kth.se

Reports can be ordered from:

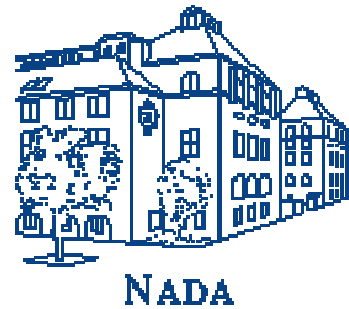
Interaction and Presentation Laboratory (IPLab)
Numerical Analysis and Computing Science (Nada)
Royal Institute of Technology (KTH)
S-100 44 STOCKHOLM, Sweden

telephone; + 46 8 790 6280

fax: + 46 8 10 2477

e-mail: reports-iplab@nada.kth.se

URL: <http://www.nada.kth.se/nada/iplab/>



Användaranpassad utveckling
av en "fetch-and-carry"-robot
för funktionshindrade i arbetslivet

Kerstin Severinson Eklundh
Erik Espmark
Anders Green
Helge Hüttenrauch
Mikael Norman
Lars Oestreicher

Rapport IPLab-186, NADA

Kungliga Tekniska Högskolan

Inneh 11

1.	Introduktion och bakgrund	2
2.	Relaterad forskning.....	4
3.	Uppgiftsanalys för intelligenta servicerobotar	6
3.1.	Servicerobotar som stöd för handikappade personer	6
3.1.1.	Intervjuer.....	6
3.1.2.	Resultat från intervjuerna	7
3.2.	Uppgiftsanalys för intelligenta servicerobotar.....	9
3.2.1.	Bakgrund till ansatsen i projektet.....	10
3.2.2.	Projektets ansats till uppgiftsanalys	10
3.3.	Idéer för fortsatt arbete	11
4.	”Wizard av Oz”-studie	12
5.	Robotens fysiska utformning	17
6.	Systembeskrivning.....	22
6.1.	Distribuerade system.....	22
6.2.	CORBA	22
6.3.	Kunskapsrepresentation	23
Operationer		23
Objekt		23
6.4.	Prototypsystemet.....	24
6.5.	Informationsflöden.....	26
6.6.	Kontroll	26
6.7.	Navigation	27
6.8.	Grafiskt användargränssnitt.....	28
6.9.	Interaktion.....	28
6.10.	CERO	29
6.11.	Talsyntes.....	29
6.12.	Taligenkänning	29
7.	Robotplattform	30
7.1.	Odometer	30
7.2.	Ultraljudssensorer	30
7.3.	Framdrivning	31
8.	Robotkontroll.....	32

8.1.	Navigation	32
8.2.	Lokalisering.....	34
9.	Grafiskt Gränssnitt.....	36
9.1.	Introduktion	36
9.2.	Designansats: Teori och process.....	36
9.2.1.	Kravspecifikation	37
9.2.2.	Systemarkitektur för robotinteraktion	38
9.3.	Detaljdesign och enhetstester	41
9.4.	Integrerat gränssnitt.....	44
10.	Utformning av talgränssnitt.....	48
10.1.	Dialogmodeller	48
10.2.	Dialogscenarier – användningsfall.....	48
10.3.	Lingvistiska och psykologiska modeller för dialog	49
10.4.	Dialogutformning.....	51
10.5.	Fysisk utformning för talgränssnitt	54
10.6.	Återkoppling med gester	54
10.7.	Jämförelse med andra typer av dialogsystem	55
10.8.	Systemutformning.....	56
10.9.	Resultat.....	58
11.	Utvärdering med användare	59
11.1.	Försöksuppsättning	59
11.2.	Genomförda försök	62
11.3.	Slutsatser, diskussion och rekommenderad fortsättning	65
12.	Slutord.....	67
13.	Referenser	68
13.1.	Böcker och artiklar.....	68
13.2.	Webbplatser	70
14.	Figurer och Tabeller.....	71
14.1.	Figurer	71
14.2.	Tabeller.....	72

1. Introduktion och bakgrund

De senare årens utveckling inom IT-området har inneburit ett ökat intresse för intelligenta hem och miljöer, och en syn på datorn som naturligt inbyggd i många vardagliga föremål. Samtidigt har robottekniken gått framåt, och man börjar mer och mer se robotar som ett specialfall av ”smarta” hjälpmedel, som det är fullt realistiskt att införliva i och anpassa till människors aktiviteter i arbete och fritid. Denna nya utveckling ställer stora krav på gränssnittet mellan människa och maskin, dvs. hur man som användare kan interagera med och instruera tekniska artefakter som har en viss grad av autonomi i sitt funktionssätt.

Robottekniken har speciellt intresse för människor med olika typer av funktionshinder, som kan vara permanenta eller tillfälliga. Vi betraktar här uteslutande rörliga robotar, som kan navigera i användarens naturliga miljö. Tidigare forskning och utveckling har visat att dessa har mycket stor potential som stöd i människors vardagsöromål, både i hemmet och på arbetsplatser. Visserligen innebär dessa miljöers föränderliga karaktär stora utmaningar för att skapa helt autonoma system, men å andra sidan finns många situationer när användarens behov av hjälp är begränsat till sådana aktiviteter som robotar är lämpade för, t.ex. att förflytta olika föremål. Det är också relevant att tänka sig en robot i andra roller för att stödja en grupp av människor, t.ex. som en rörlig kommunikationscentral. Men även här krävs en övergripande modell av hur interaktionen mellan människa och robot ska utformas, ett forskningsområde som ännu är ganska outvecklat.

Denna rapport beskriver ett tvåårigt projekt vid NADA, KTH med inriktning mot en ”fetch-and-carry”-robot för funktionshindrade personer, tänkt att användas i kontorsmiljö. Den ursprungliga idén har varit att skapa ett system som stöder en person som har svårt att gå, genom att hjälpa till att transportera olika kontorsföremål och personliga ting (t.ex. kaffekoppar) inom en arbetsplats. Vi tänkte oss att roboten skulle ha två olika funktionssätt:

- Roboten går från en plats till en annan på användarens uppmaning, för att hämta eller leverera ett föremål.
- Roboten följer användaren kontinuerligt medan han eller hon rör sig i miljön, för att vara till hjälp med transporter.

Projektarbetet har resulterat i en fungerande prototyp, CERO-roboten, som kan navigera i en begränsad del av en kontorsmiljö. Roboten är utrustad med en specialdesignad transportdel, mikrofon och högtalare. Den styrs av användaren via tal eller via ett grafiskt gränssnitt.

Arbetet i projektet har följt en användarcentrerad modell, där stor emfas lagts på studier av användarnas behov för att utröna vilka scenarion som är intressanta att stödja. Hit hör intervjuer med både friska och funktionshindrade personer för att nå fram till relevanta uppgifter, samt användarförsök med en simulerad prototyp i ett tidigt skede av utvecklingen. En industridesigner engagerades tidigt för att analysera de fysiska kraven på systemet och ge roboten en tilltalande utformning, vilket gett mycket viktiga bidrag till slutresultatet. Andra centrala delar av projektet har berört utformningen av gränssnittet mot användaren, samt konstruktion av ett robust system för koordinering och planering av robotens uppgifter. En mindre användarutvärdering av den slutliga prototypen har också utförts.

För forskningsgruppen har projektet utgjort en pilotstudie, där vi för första gången angrep många viktiga frågeställningar inom människa-robotinteraktion. Vi har av naturliga skäl endast kunnat realisera en del av alla ideer som kommit fram. Trots att vi fått viktig hjälp av

CAS (Centrum för autonoma system, KTH), har tekniska svårigheter och begränsningar varit styrande för en stor del av arbetet, och har t.ex. gjort att vi i den version av roboten som realiserats inte kunnat integrera de båda funktionssätten ovan. Även i andra avseenden är den prototyp som projektet resulterat i begränsad till sin funktionalitet. Emellertid har arbetet i många avseenden varit mycket lärorikt och har krävt insikter från en rad olika kunskapsområden, där IPLabs tvärvetenskapliga arbetssätt kommit till stor nytta.

Nedan beskrivs rapportens innehåll i korthet, samt huvudansvarig person för varje kapitel.

Kapitel 2 (ansvarig: HH) tar upp en del relaterad forskning i korthet, framför allt när det gäller robotar för handikappade. I kap 3 (LO) analyseras robotens uppgifter, dels med utgångspunkt från intervjuer med funktionshindrade personer, och dels i form av en föreslagen ansats för uppgiftsanalys som kan ligga till grund för design inom detta område. Kap 4 (HH) behandlar en simuleringsstudie av Wizard-of-Oz-typ, som genomfördes på ett tidigt stadium för att få input till designarbetet.

I Kap 5 (EE) beskrivs de resonemang och avvägningar som lett till robotens fysiska design, samt en del reflektioner kring alternativa utformningar. Den tekniska implementationen av robotsystemet, samt själva robotplattformen, beskrivs i kap 6, 7 och 8 (MN). Utformningen av det grafiska gränssnittet, och de överväganden som det baserats på, beskrivs i kap. 9, och motsvarande för talgränssnittet följer i kap. 10. Kap. 11 beskriver en preliminär utvärdering med användare, och kap. 12 ger en kort sammanfattning och slutsatser.

I projektets tidiga fas ingick även ett examensarbete kring robotnavigering, som utfördes av Henrik Tåquist (Tåqvist 1999) under handledning av Henrik Christensen. Då resultatet av detta arbete inte kunde användas i robotsystemets slutliga implementation av tekniska skäl, har vi valt att utelämna det här.

Vi vill uttrycka vår tacksamhet till Henrik Christensen, Olle Wijk och Anders Orebäck på CAS för teknisk hjälp och stöd med robotikkunskaper under projektets gång. Vi tackar också AMS som finansierat projektet, och speciellt Ulf Keijer, Inger Rundström och Jan Breiding, som deltagit i långa diskussioner och gett många goda råd.

Lars Oestreicher har ansvarat för rapportens grafiska utformning.

Stockholm i januari 2001

Kerstin Severinson Eklundh

Projektleddare

2. Relaterad forskning

Vi ger i detta kapitel en kort översikt av relevanta teknologier och projekt kring människa-robotinteraktion, med speciellt fokus på forskning inom hjälpmedelsrobotik.

Engelhardt (Engelhardt och Edwards 1992) räknar upp ett antal olika typer av robotar, bland andra industrirobotar som t.ex. används på verkstadsgolvet, servicerobotar, personliga robotar, pedagogiska robotar, datorstyrda hushållsmaskiner samt olika apparater som bäst klassificeras som avancerad hemelektronik. Kategoriseringen speglar den fortskridande diversifieringen inom robotikområdet. Idén bakom indelningen i olika robottyper är att kunna identifiera de olika särdrag som kännetecknar robotar avsedda för olika användningsscenarion. Indelningen måste dock betraktas med en nypa salt då den teoretiska begreppsuppsättningen inte kan ge klara besked hur t.ex. en servicerobot hemma ska skilja sig ifrån en apparat som skulle kunna säljas som avancerad hemelektronik. Osäkerheten beror på att vi inte kan slå fast hur dessa olika beteckningar i framtidens marknadsförs eller används av konsumenter.

På utvecklingssidan finns ett antal forskningsfält som har visat intresse för robotik, däribland t.ex. sensorteknologi, digital bildbehandling/datorseende, artificiell intelligens (AI) för att nämna några. Oftast sätts bara vissa frågeställningar i fokus varför delar eller komponenter av system utvecklas i experimentellt syfte. Sådana system kan på grund av detta komma att bortse från centrala problem såsom en fullbordad integration till ett fungerande robotsystem från användartester eller en (industriell) produktifiering.

Som en konsekvens ger en indelning av robotik i olika forskningsfält å ena sidan en antydning om vilken typ av forskning och utveckling som bedrivs, å andra sidan förblir ansatsen att beskriva hjälpmedelsrobotik för bred och ofokuserad när det gäller att ta steget från experimentella prototyper till produkter.

Om man ser till de komponenter som ingår kan robotar indelas i system som "bara" består av en mobil plattform, en mobil plattform med en eller flera manipulatorer (t ex armar), endast en stationär manipulator arm eller system som består av en plattform med begränsad mobilitet och manipulator (ex en arm monterad på räls). Skillnader i systemuppbyggnad möjliggör eller begränsar sedan användningen av roboten i olika uppgiftssammanhang (jfr även Oestreicher under utg.).

Detta förhållande kan exemplifieras med de olika typer av förflyttningsteknik som används på robotar idag, t ex hjul, band eller ben. Robotar som manövrerar med hjälp av ett hjulbaserat system kan ha bra förmåga att ta sig fram i olika inomhusmiljöer med platta golv (och frånvaro av tröskel), däremot är de flesta hjulbaserade system inkapabla att ta sig upp eller ner för trappor (som dessvärre oftast finns i olika inomhusmiljöer). Ett bandburet system eller ett system med mekaniska ben är till fördel för förflyttning i naturliga miljöer (t.ex. skogar) då man får ökad framkomlighet på olika underlag och även olika slags hinder kan passeras. Denna systemdesignfördel måste dock avvägas mot den ökade komplexiteten som ett ben- eller bandburet drivsätt innebär.

I vårt projekt (Severinson Eklundh 1998) är vi främst intresserade av gränssnittsfrågor för en mobil robot (på hjul). Roboten har ingen arm och skall utföra transportuppgifter i inomhusmiljöer som t ex kontor och hem. Tänka användargrupper representeras av folk som behöver hjälp med att bära och transportera föremål.

AMS har tidigare stött ett antal robotikrelaterade projekt, däribland anpassning av arbetsplatser för människor med handikapp. Som exempel på robothjälpmedel på arbetsplatser kan följande projekt anföras: I Helsingborg installerades en robotkontorsarbetsplats, en robotstation för automatiserad posthantering provades hos Volvo Transport och i Göteborg provades en robot för en sopåtervinningsstation (Neveryd m fl. 1999).

Andra projekt i Sverige som inriktat sig på robotar som hjälpmedel har genomförts av Gunnar Bolmsjö och hans forskningsgrupp (www.robotics.lu.se) vid Lunds universitet (Bolmsjö och Efring 1993). Projekt som har realiserats är t ex framtagning av kontrollsystem och användartester av RAID (en robotarbetsstation, (se Efring 1994), utveckling av en prototyp till robotarm för rullstolar (Fridenfalk m fl. 1999) och RAIL (engelsk förkortning för Robotic Aid for Independent Living, diskuterad i Bolmsjö och Efring 1993) för att nämna bara ett urval.

CERTEC (www.certec.lth.se) har en lång historia av arbete med service- och hjälpmedelsrobotik, speciellt i de projekt som är fokuserade på användning av robotarmen MANUS (för montage på rollstolar) och framtagningen av WALKY. WALKY är ett hjälpmedel avsett för arbete i laboratoriemiljö för människor med fysiska handikapp. Håkan Efring (1999) summerar erfarenheterna med en robotarbetsstation som har körts under många år och rapporterar om en utvärdering som genomfördes av robotarmen MANUS (Efring och Boschian 1999) med användarens behov och användbarhet av systemet i fokus.

WALKY (Neveryd 1998) är ett mobilt robotsystem med en liten arm som var tänkt att användas för transportuppgifter, t ex i laboratoriemiljöer av folk med handikapp. Neveryd ger en introduktion till rehabiliteringsrobotik, samt beskriver användning av den mobila plattformen, navigationen som sker med ultraljudssensorer och ett grafiskt gränssnitt.

Kawamura (1996a; 1996b) från Vanderbiltuniversitetet beskriver en helt annan ansats inom människa-robotinteraktion. De definierar sin filosofi för serviceroboter som är baserad på arbete utfört av Ejiri (1995) och Leifer (1995): Robotar skall enligt denna ha ett vänligt och kooperativt gränssnitt, en enkel struktur, funktioner som är lättförståeliga, ett utseende som är i harmoni (Kinsai) med omgivningen samt en design som motsvarar ett socialt beteende efter som robotar kan anses vara sociala agenter (originalkraven på engelska, egen översättning).

Kawamuras designfilosofi är intressant även för vårt robotprojekt. Däremot är deras tänkta scenario, med robotar försedda med två armar som samarbetar i ett hushåll, inte applicerbart på vårt projekt. Vi har begränsat oss till en situation där användaren samarbetar med en enskild robot. Tanken är alltså att användaren och roboten skall jobba ihop, men inte att flera robotar (initialt) skall ingå i systemet.

När det gäller människa-robotinteraktion och användare med handikapp har relativt lite forskning gjorts. Vid universitet i Cambridge (rehab-www.eng.cam.ac.uk) pågår sedan början av nittiotalet forskning kring användbarhet för rehabiliteringsrobotik.

Dallaway (1993) beskriver CURL, ett robotspråk vars grammatiska struktur liknar naturligt språk och därför medger programmering av uppgifter av personer utan stora programmeringskunskaper. Clarkson (1997) gör observationen att även om många metoder inom människa-datorinteraktion används vid utformning av programvara, är de oftast inte lämpliga vid utformning av gränssnitt av interaktiva maskiner och robotar. Dowland (1997) visar nyttan av att använda prototyper för att utforma robotens design, dess kontrollmekanismer och applikationer.

3. Uppgiftsanalys f r intelligenta servicerobo

En del av forskningsarbetet rörande intelligenta servicerobotar vid IPLab har fokuserats kring arbetsuppgifternas innehåll. Vad skall en robot användas till? Eller snarare, hur skall den egentligen användas? Inom ramen för arbetet med robotar i AMS-projektet har dessa frågor kommit att handla om det speciella stöd som en intelligent servicerobot kan erbjuda en person med fysiskt handikapp.

Den typ av robot som vi har använt som utgångspunkt (Nomadic Systems SuperScout II, se Figur 14) saknar manipulatorer och optisk igenkänning av omgivningen vilket har ställt vissa mer speciella krav på arbetsuppgifterna som roboten skall utföra. Bland annat kan roboten inte helt autonomt hämta föremål, utan den kräver assistans av sekundära användare för att kunna lasta och lasta av föremål.

Just att kunna hämta föremål på egen hand är (vilket kommer att framgå mer senare i detta kapitel) något som uttrycktes som ett önskemål från informanterna.

Undersökningen har dels bestått i en genomgång av möjliga uppgiftsanalysmetoder för denna tillämpning, med en analys av brister och förtjänster. Dels har vi gjort en intervjustudie med ett antal personer med varierande grad av fysiskt handikapp. I denna studie har vi försökt isolera möjliga uppgifter och deras struktur, men även eventuella metaforer och liknelser. I detta kapitel skall vi redovisa båda dessa delar.

3.1.Servicerobotar som st d f r handikappade personer

I undersökningen har vi försökt finna de egenskaper som utmärker handikappades behov av stöd från servicerobotar. Det finns många krav som kommer direkt från de handikappades beroendeställning, som t.ex. behovet av trygghet och specifika tjänster från en individuell synvinkel. För att få en bild av de krav som personer med olika former av handikapp kan ställa på denna typ av maskiner har vi genomfört intervjuer med personer som har olika grad av handikapp.

3.1.1Intervjuer

Vi har totalt intervjuat 7 personer, med varierande typer av handikapp — från personer med kraftig reumatisk ledvärk, till personer som är helt rullstolsbundna av olika skäl. Den resulterande bredden på intervjumaterialet ger många perspektiv på vad vi kan förvänta oss som möjliga krav från framtida användare.

De intervjuade kommer från olika användningsmiljöer, och är alla yrkesverksamma i olika grad. Deras respektive situationer har varit följande: tre personer med så gott som full rörlighet, men svaghet i muskler eller leder (I1, I2, I4), fyra personer som har varit helt rullstolsbundna (I3, I5, I6, I7). Intervjuerna har tagit mellan en och två timmar vardera och har skett i arbets- eller hemmiljön.

Intervjuerna har transkriberats i sin helhet, och därefter har de bearbetats för att hitta de viktiga aspekterna på robotens verksamhet i hemmet eller på arbetsplatsen. Speciell vikt har lagts på sådana aspekter som berör hur roboten funktionellt kan stödja eller underlätta användarens arbetsuppgifter. En svårighet har därvidlag varit att flera av användarna naturligtvis har arbeten och arbetsplatser som till viss del är utformade för att passa deras handikappsituation, vilket medför att de ofta inte har speciellt rörliga arbetsuppgifter. Det i sin tur betyder att de-

ras nuvarande arbetsuppgifter inte alltid är lämpade för att understödjas av en robot, eller att de måste tänka sig in i en fiktiv situation där roboten kan komma till nytta för dem. Flera arbetar dessutom i hemmiljö, vilket gör att hem- respektive arbetsmiljö inte alltid kan separeras. Även om det naturligtvis ger mycket viktig information, så betyder det samtidigt att vi inte får tag på de verkliga behoven i dagsläget. Man kan dock konstatera att resultatet från intervjuerna trots dessa kända problem är intressant som utgångsmaterial

3.1.2 Resultat från intervjuerna

Från intervjuerna kan man dra slutsatser både vad beträffar arbetet med den enklare form av robot som vi arbetar med konkret vid IPLab och beträffande en mer avancerad typ av robot som har möjlighet att hitta och plocka upp föremål, t.ex. från golvet. Slutsatserna presenteras i de följande avsnitten.

3.1.2.1. CERO-roboten

Det finns i huvudsak två eller möjligen tre huvudsakliga typer av uppgifter som behöver beskrivas för CERO-roboten i den arbetssituation som vi förutsätter på IPLab:

- Gå och hämta/lämna något på en plats,
- Följ mig till en plats eller utmed en serie platser, och
- Visa en besökare runt eller till en viss person.

Utöver dessa tre grundläggande uppgifter kan man tänka sig variationer, t.ex. att roboten fungerar som en rörlig liten plattform för att flytta objekt lokalt inom ett arbetsrum exempelvis mellan bokhyllor eller mellan bokhylla och arbetsbord, eller att roboten går en bestämd runda, runt ett antal rum t.ex. för att samla in kaffekoppar eller liknande (I1). Det är ganska tydligt att denna begränsade funktionalitet också reducerar användbarheten för roboten. De flesta intervjupersonerna efterlyste någon form av manipulationsmöjlighet i intervjuerna, även om de också såg många användningsmöjligheter för den enklare varianten.

Följ-mig-scenariot kan om man vill förenkla interaktionen dessutom i viss mån reduceras till ett scenario där roboten ombeds att gå till en viss plats oberoende av när ägaren går dit. "Följ-mig"-funktionaliteten är dock inte möjlig att helt ersätta med en "Gå-till" viss plats. Den är nödvändig för det fall då användaren inte vet exakt vilken väg han eller hon skall gå, eller då uppgiften innefattar att gå till flera ställen efter varandra. Som exempel på denna typ av uppgift nämnde en informant att hon gärna ville att roboten skulle bära vattnet åt henne medan hon gick runt och vattnade blommorna i lägenheten (I4). Den uppgiften kan inte förprogrammeras utan bygger på att roboten följer efter användaren i dennes spår.

Roboten utan manipulatorer sågs främst som en intelligent transportvagn, dvs. användaren kunde lägga olika saker på robotens transportyta, och sedan instruera den att antingen följa användaren till en plats, eller att gå till en plats för att lämna eller hämta saker från en annan person. Som ett typiskt exempel nämnde en person att han kunde skicka iväg roboten med en nyckel till ett förråd, så att han själv slapp gå och lämna ut nyckeln. De flesta av informanterna ansåg att roboten skulle fungera väl som en intelligent tevagn, en tredje hand eller en rullande ryggsäck.

En intressant erfarenhet är att för många av de arbetsuppgifter som nämndes i intervjuerna sågs roboten framförallt, eller nästan helt uteslutande som stöd snarare än utförare, t.ex. när man på en arbetsplats skulle byta glödlampor eller tillse laserskrivare (I2) eller som nämndes tidigare, vattna blommor (I4).

3.1.2.2. Robot med syn och manipulatorarmar

Den andra typen av robot som vi har tagit med i undersökningen är en robot som har möjlighet att "se" saker och ting i sin omgivning, och även en möjlighet att manipulera dessa med en eller två armar. I dagens läge är både syn och manipulering att karaktärisera som icke-triviala uppgifter för en robot baserad på en mobil plattform (dvs. en robot som kan förflytta sig själv i sin omgivning). Å andra sidan finns det viss grund till optimism vad gäller utvecklingen inom båda dessa områden, vilket gör att det blir intressant att studera även hur denna robots arbetsuppgifter skall vara utformade.

Här fanns det också betydligt fler tänkbara arbetsuppgifter där roboten fungerar som en tredje hand åt användaren, eller (se mer nedan) som en liten hjälpsam hund. Det behov som främst framkom var att roboten skulle kunna skickas iväg för att hämta olika typer av saker åt personen.

För de flesta av informanterna var en robot med manipulatorer och någon form av visuell igenkänning en intressant möjlighet till stöd, även om de framhöll att enbart den transporterande roboten skulle utgöra en stor fördel jämfört med att klara sig utan.

3.1.2.3. Robothunden som metafor för intelligent servicerobot

Ett sätt att vidareutveckla arbetsuppgifterna för en servicerobot är att hitta bra metaforer för robotarna. Ofta har man diskuterat servicerobotar som en personlig butler, eller en mekanisk piga. En möjlig alternativ metafor kom fram under en av intervjuerna, där intervjupersonen dagligen använde sig av en s.k. servicehund för att klara av det dagliga livet. Hunden (som tränas under ca 1 år utöver traditionell brukshundsträning) hanterar en stor mängd små, triviala, men viktiga och för användaren tidskrävande göromål.

Gemensamt för många av servicehundens uppgifter är att de är relativt enkla och inte kräver komplicerad kommunikation. Uppgifterna är i de flesta fall till största delen begreppsmässigt enkla, t.ex., *öppna/stäng dörren, hämta plånboken, plocka upp handsken, lägg papperet i papperskorgen*. Servicehundens primära uppgift är att förse ägaren med den extra arm eller hand som denne behöver när den egna kroppens resurser inte räcker till (se t.ex. Figur 1 där hunden drar av en strumpa från foten på en person som inte har kraft nog att göra det själv).



Figur 1. En servicehund i arbete (markerat genom västen) som drar av en strumpa på sin matte.

Men vissa mer komplicerade uppgifter kan läggas till om det krävs. Så har man t.ex. tränat hundar för att bädda sängen eller plocka ur tvätt ur tvättmaskinen. Alla hundar får träna in en grundläggande uppsättning uppgifter som de skall klara av, och sedan tränar ägaren in en

egen uppsättning personliga uppgifter om så behövs. De grundläggande uppgifterna som en servicehund skall utföra är bl.a.:

- Apportera olika sorters föremål av olika materialtyper, papper, plast, trä, metall, m.m.
- Öppna och stänga dörrar
- Trycka på dörröppnare
- Bära (transportera) enkla saker

Sedan tillkommer andra uppgifter beroende på ägarens behov och möjligheter. Här är några mer eller mindre avancerade exempel:

- Spola på toaletten
- Riva av toalettpapper
- Dra av jackärmar, strumpor och andra klädesplagg (se figur 2)
- Hissa flaggan i flaggstången
- Plocka ut tvätt ur tvättmaskin

Sammantaget kan vi se att en servicehund måste vara relativt flexibel i sin uppgiftsdefinition, men att uppgifterna också i de flesta fall är relativt enkla till sin karaktär.

Vissa hundar har också tränats upp för att reagera på ljud, och alarmer sin ägare vid olika typer av ljud, t.ex. dörrklockor, väckarklockor, telefoner, alarm, m.m. Beroende på situation och typ av ljud, kan hunden sedan utföra olika uppgifter, t.ex., hämta telefonen, öppna dörren eller liknande.

Vi kan se att dessa uppgifter alla är väldigt enkla, och i många fall variationer på en och samma grundläggande beteende. Uppgifterna är också lagom komplexa för att en servicerobot skulle kunna klara dem (åtminstone i de fall uppgifterna är intressanta att hantera med hjälp av en robot).

Även om intervjusituationen kom att fokusera på hur hunden arbetade i hemmiljö, ser vi att hundens insatser mycket väl kan överföras till en arbetssituation. Hunden följde naturligtvis också med användaren på arbetet, och fick där hämta och bära saker. Även dörröppning och liknande ingick i dess uppgifter på arbetsplatsen. Det framgick att hundar ibland tränas till att hjälpa personer med deras arbete genom att t.ex. hämta eller lägga tillbaka verktyg.

Här kan vi se att det finns många intressanta arbetsuppgifter där hunden kan vara en bra metafor för den intelligenta serviceroboten. Genom att anta en sådan metafor vinner man flera fördelar, nämligen att man får ett sammanhang inom vilket det är naturligt att använda sig av en robot, man får ett förväntat sätt att kommunicera med roboten (genom liknande kommandospråk som till en hund), men inte minst att roboten lättare kan komma att uppfattas som en positiv och ofarlig medarbetare.

3.2.Uppgiftsanalys för intelligenta serviceroboter

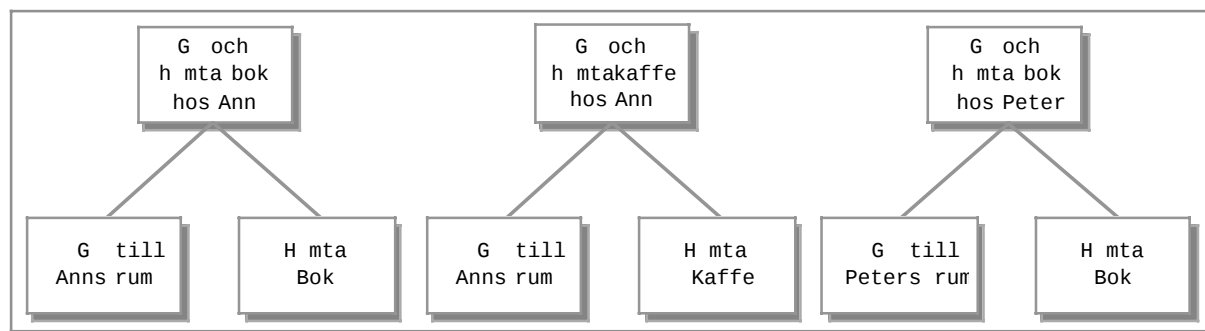
I projektet har vi studerat hur man skall kunna beskriva de uppgifter som en användare vill ha utförda eller understödda. En utgångspunkt för denna del av undersökningen har varit uppgiftsanalys (jfr Kirwan och Ainsworth 1993). Det finns många olika varianter och ett led i undersökningen har varit att försöka fånga problemen med uppgiftsanalys i samband med robotar. Som en följd av det har vi börjat utveckla ett sätt att beskriva uppgifter (arbete eller

arbetsmoment) som lämpar sig för att beskriva robotens verksamhet, även i samarbete med användaren.

3.2.1 Bakgrund till ansatsen i projektet

Uppgiftsanalys har funnits för många olika ändamål, och ett viktigt sådant har bland annat varit att formulera hur arbete med datorer skall utformas. För det ändamålet finns det ett stort antal olika metoder som alla i princip baseras på en av tre övergripande varianter, nämligen att dela upp arbetsuppgifter (och/eller begreppen i verksamheten) i mindre delar, eller att beskriva kunskapen som man behöver för att kunna utföra arbetsuppgiften, eller att beskriva hur relationen mellan olika begrepp förändras över tiden (jfr Diaper 1989; Dix m fl. 1998). Det visar sig relativt snart när man börjar försöka beskriva robotens uppgifter att funktionaliteten är av sådant slag att traditionella metoder inte alls, eller endast med stor svårighet kan beskriva robotens verksamhet. Ser man dessutom robotens verksamhet som ett samarbete mellan två (eller flera) aktörer (användare och robot) saknas mycket stöd i beskrivningsmetoderna.

Ett exempel på problem som är specifikt för CERO-roboten är att den har en mycket begränsad möjlighet till egna aktiviteter. Den förväntas dock arbeta i en komplex miljö, med många olika typer av objekt att transportera och många platser att orientera till. Detta innebär att även om antalet grundläggande handlingar är mycket begränsat, kommer det med en traditionell uppgiftsanalys att krävas en omfattande beskrivningsapparat för att täcka in robotens verksamhetsområde. För att hämta ett objekt på ett rum behövs, t.ex., olika regler för varje kombination av rum och objekt om vi ser dessa som olika aktiviteter.



Figur 2. Tre olika uppgifter som alla innehåller en liknande grundstruktur, och som är olika instanser utav ett och samma grundläggande mönster.

Av den anledningen behöver en uppgiftsanalys för robottillämpningar utformas baserat på mer generella regler där platser och objekt kan utgöra konkretiseringar av generella variabler, snarare än på specifika uppsättningar planer för olika händelser som kan förekomma (se figur 1).

3.2.2 Projektets ansats till uppgiftsanalys

Den metod som utvecklats för att beskriva robotens arbete bygger i princip på två vägledande principer: dels att beskrivningen (specifikationen) av robotens arbetsuppgifter skall vara enkel att utvidga, dels att användaren och roboten båda har mål med sin verksamhet. Utöver detta ser vi att metoden måste ge utrymme för att roboten skall (åtminstone i begränsad omfattning) kunna vidta åtgärder om den upptäcker hinder för att lösa uppgiften. Detta innebär att uppgiften kommer att behöva beskrivas i termer av ett antal (ibland konkurrerande) parallella mål som skall uppfyllas.

Som ett exempel kan vi ta en generell regel som beskriver hur roboten kan gå och hämta en *kaffekopp i köket*:

```
G    och hämta en >>sak>> på en >>plats>>
      Där en >>sak>> kan vara ett objekt av följande typ: kaffekopp, bok,
      Och en >>plats>> är en plats med följande kriterium:
          platsen måste innehålla objektet >>sak>>.
```

Samma regel kan t.ex. beskriva hur man hämtar en bok i ett rum, eller en pappersbunt i ett förråd. Detta förutsätter att man kan tala om för roboten vilka saker och platser som finns i världen och vilka platser som kan innehålla vilka saker. Andra regler kan skapas, t.ex. för att beskriva hur roboten visar en person var någon har sitt arbetsrum. Vi kan även skapa regelsystem som talar om hur två olika aktörer tillsammans löser en uppgift, genom att tilldela olika dellösningar olika aktörers ansvar.

Detta är ett enkelt exempel på hur man kan skapa enkla regler för hur roboten skall bete sig, utan att förlora alltför mycket av precisionen i beskrivningen. Dessutom ger denna form av beskrivning en möjlighet att beskriva en stor mängd av relativt likartade uppgifter på ett enkelt sätt, och där inprogrammering av nya uppgifter till stor del består av att lära roboten nya platser och nya objekt.

Alla dessa regler kan skrivas så att de uttrycker ett mål med en verksamhet. Genom att roboten har flera mål tillgängliga samtidigt kan den få olika alternativ att välja mellan, t.ex. om det skulle uppstå problem på vägen. Ett exempel på en regel som kan ligga som standard parallellt med alla andra regler som den beaktar, är följande:

```
    tervänd till utgångspunkten om >>n got händelse>>
    Rapportera att >>uppgift>> inte är genomförd.
      Där >>n got händelse>> kan vara en händelse av följande typ:
          vägen blockeras, batteriet blir svagt
      Där >>uppgift>> är vilket möjligt uppdrag som helst
```

Denna regel kan ges en övergripande prioritet, eftersom den måste initieras av en händelse, som t.ex. att någon har ställt något i vägen i korridoren, eller att en dörr är låst. Det innebär att så fort roboten upptäcker ett problem, kan den själv välja ut ett nytt mål som den skall utföra istället för den uppgift som inte kunde fullgöras.

3.3. Identifiera fortsatt arbete

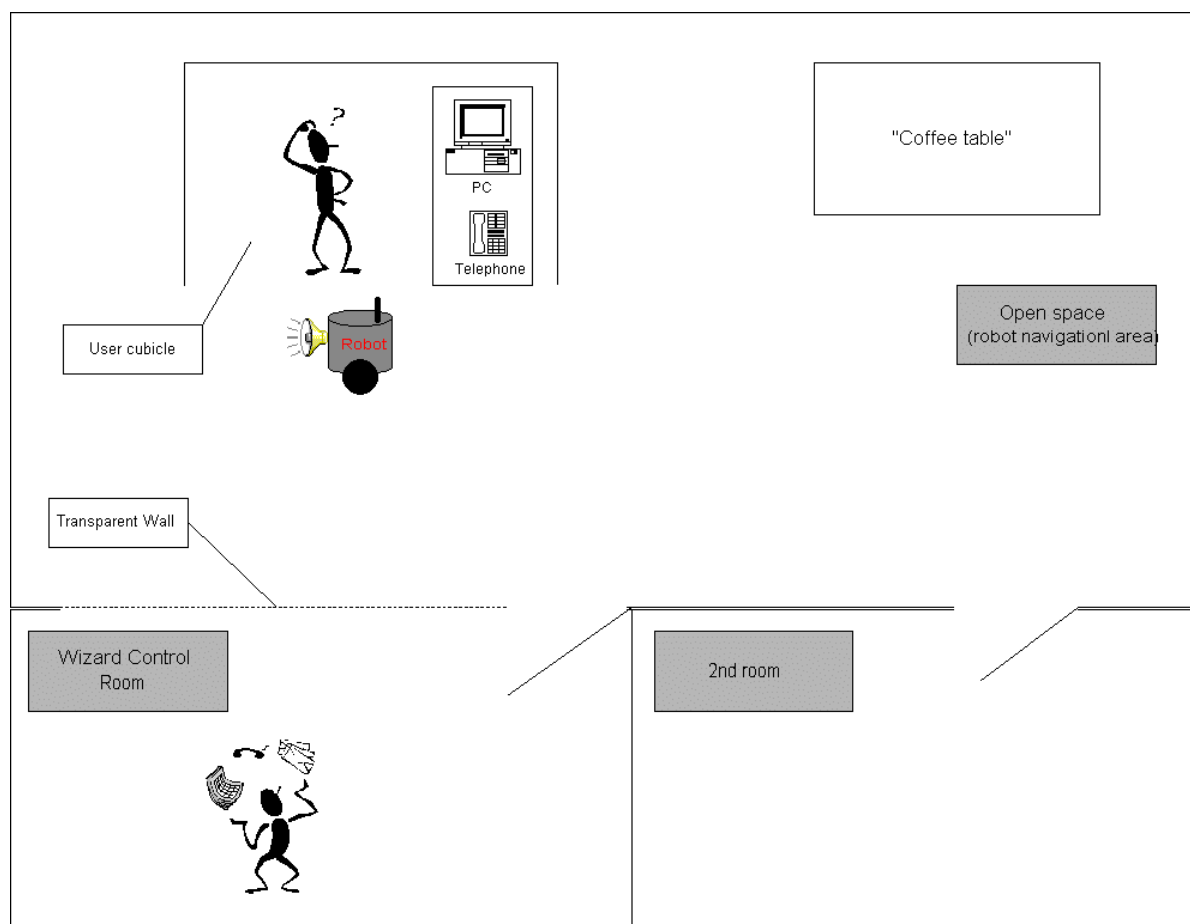
Arbetet med roboten inom ramen för detta projekt har gett upphov till ett antal frågeställningar som måste bearbetas vidare. Framförallt ser vi intressanta problem, dels vad gäller den rent praktiska uppgiften att beskriva robotens arbete, och det på ett sådant sätt att resultatet kan användas för att bygga och vidareutveckla system som fungerar praktiskt i en hem- eller arbetsmiljö, dels vad gäller den mer övergripande aspekten rörande acceptansen för en robot som en servicekapacitet för t.ex. handikappade personer. I det senare avseendet handlar det främst om att förmedla tydliga och tilltalande metaforer för roboten. Istället för att kalla den för *robot*, kanske den bör liknas vid någon annan mer neutral företeelse som användaren enkelt kan knyta an till.

4. Wizard av Oz -studie

I ett första försök att låta slutanvändare faktiskt köra roboten i en experimentuppsättning genomförde vi en s.k. "Wizard av Oz"-studie (Dahlbäck m fl. 1993; Maulsby m fl. 1993). Denna teknik är känd som undersökningsmetod främst inom naturligt-språkforskning och har framgångsrikt använts för att utveckla t.ex. interaktiva tal- och informationstjänster (Bernsen m fl. 1998), som t.ex. tjänster för försäljning av biljetter m.m.

Intressant i "Wizard av Oz"-metoden är att funktionen som skall testas inte behöver vara helt färdig och implementerad i systemet. I stället simuleras den ännu inte färdigställda funktionaliteten av en mänsklig operatör som (re-)agerar för att efterlikna systemets framtida funktion. Operatören som egentligen styr bakom kulisserna kallas för Wizard (eller på svenska trollkarl) eftersom det är han som fyller prototypen eller simuleringen av prototypen med "liv".

Vi använde följande uppsättning: en trollkarl satt gömd för användaren bakom en glasvägg och lyssnade till kommandon som gavs från användaren. Han matade sedan in robotens svar som uttalades av talsyntessystemet på roboten. Robotens rörelse kontrollerades av en andra trollkarl eftersom navigeringssystemet inte var fullt klart ännu..



Figur 3. Uppsättningen av "Wizard av Oz"-försöket.

Flera uppgifter fanns förberedda för användarna att testa: Användaren skulle instruera roboten att transportera en tidskrift till en annan person. Sedan skulle användaren följa med roboten

till ett kaffebord där en kopp kaffe skulle hämtas och transporteras tillbaka till det kontorsliknande område som vi byggt till försöket (se t.ex. Figur 5). Slutligen skulle roboten skickas tillbaka till sin uppladdningsstation och sättas i "standby" mode. Dessa uppgifter ansågs vara representativa för de uppdrag som roboten borde kunna utföra enligt projektspecifikationen. Vi hade som försökssyfte att studera hur användare umgås med vår mobila robotplattform, om det finns systematiska interaktionsmönster bland användare och vilken kommunikationsmodalitet som föredras av användaren.



Figur 4. Kontrollrummet från vilket roboten styrs under Wizard-of-Oz-testet. För att hantera roboten under testen krävdes två personer, en som styrde roboten och en som hanterade talsyntessystemet.

"Wizard av Oz"-metoden har, så vitt vi vet, inte tidigare använts inom ett människa-robot-interaktionsförsök. Därför var denna studie företrädesvis explorativ, men det visade sig mycket givande att pröva den metoden (för en mer detaljerad beskrivning se Hüttenrauch under utg.).



Figur 5. Användare under ett försök.

Vid tidpunkten då vi planerade "Wizard av Oz"-försöket, hade vi kört igång vår robotplattform. Navigeringssystemet var under utveckling, ett första grafiskt gränssnitt hade kort testats, en talsyntesmjukvara ("Festival" från universitet i Edinburgh, se Black m fl. 2000) hade installerats och ett första utkast till en skål som roboten kan transportera saker i var framtagen. Tanken var att användaren skulle testa det grafiska gränssnittet och ge kommando till roboten med tal för att utföra några uppgifter.

För datainsamling spelade vi in på video det som hände i interaktionen mellan användaren och roboten och registrerade det som hände i programvaran (genom dataloggning). Efter att uppgifterna hade utförts av användaren intervjuade vi denne för att fråga hur roboten hade uppfattats. Vi konfronterade användarna med hur de hade interagerat med roboten för att ta reda på varför de hade kommunicerat med roboten på det sättet som de valde.

Under försöket framgick att försöksuppsättningen som vi planerade var för komplicerad och ambitiös. Det gick inte att hantera både ett grafiskt gränssnitt, talgränssnitt och robotens navigering samtidigt. Vi upptäckte att samspelet mellan de olika gränssnittskomponenterna och robotens manövreringssystem å ena sidan och användarens aktioner och interaktionsåterkoppling å andra sidan blev för komplex att hantera tillsammans. Därför valde vi att fokusera studien på att låta användaren köra roboten med bara tal, dvs. låtsas att användaren talar in kommandon till roboten. Sedan reagerar roboten med hjälp av wizards styrkommando vid kontrollpanelen bakom glasväggen. Wizard nummer två svarade på intalat kommando genom att knappa in ett skrivet svar i en dator. Svaret överfördes trådlöst och uttalades via talsyntes på själva roboten.



Figur 6. Besökare som leker med roboten före ett Wizard-of-Oz-försök.

Vi körde i första omgången sex försök med användare som rekryterades inom vår institution och bland studenter som var intresserade att delta. Eftersom studien också var tänkt att lära oss hur robot-användarförsök skulle kunna genomföras i ett tidigt stadium förändrade vi iterativt detaljerna i försöket när det visade sig att det gick att förbättra vår uppsättning. Data som vi samlade in visade sig vara rikliga. Vi upptäckte (och fick bekräftelse på) att interaktionen sker i mycket små, intensiva och snabba tidsenheter, dvs. bara några sekunder räcker för att användaren ska kommunicera med roboten. Samtidigt verkar det som om interaktionen under "Follow-Me"-situationen egentligen inte begränsas till "enheter", alltså perioder med

en bestämd start- och slutpunkt. I stället pågår interaktionen över hela perioden då uppgiften utförs med upprepade blickar och rörelser. I inspelad interaktion speglas detta genom de många observationsvariabler som måste analyseras i förhållande till den absoluta försökstiden och situationen som försöket vid en viss tidpunkt är inne i. Variabler (eller kategorier av händelser) som vi kunde observera handlar till exempel om användarens och robotens position i rummet, det spatiala förhållandet dem emellan, robotens rörelse, steg, gester, minspel, emotionella yttranden från försökspersonen, uttalanden från användaren respektive robotsystemet, reaktionstider mellan interaktionsenheter, fel i kommunikativ turtagning (t.ex. när roboten börjar pratar samtidigt som användaren) o.s.v.

Intressant är att interaktionen med en fysisk robot skiljer sig från kända interaktionsmönster t.ex. i skärmbaserade eller telefonbaserade interaktiva talsystemtjänster: Roboten och användaren har potentiell förmåga att röra sig fritt i miljön, och bara detta gör samspelet mellan robotar och människor mycket komplexare än med traditionella datorsystem. Som ett återkommande mönster verkar en rörelse hos roboten följas av en rörelse hos användaren, eller också tvärtom, dvs. att användaren väntar på rörelse från roboten innan blicken vänts bort och nästa steg tas. Med en metafor kan det kanske beskrivas så att interaktionen sker i en sorts "föräldrar med barn i köpcentrum"-beteende. Barnet går fritt, men är permanent under uppsikt av pappan/modern och deras ingripande (via anrop) eller reaktion via rörelse (springa efter, gå vidare) kan ske i vilket ögonblick som helst.

Denna observation tolkades så att även i en interaktion med bara tal som medium är interaktionen multimodal, dvs. andra kommunikationssätt än tal används för kontroll av roboten. Tillsammans med dynamiska förändringar under försöket (t.ex. roboten och användaren som rör på sig) förblir realtidsreaktioner från ett mobilt robotsystem mycket viktiga för att interaktionen inte ska verka onaturlig. Roboten verkar följa förväntningar som människor har mot en person som kommunikationspartner, inte ett tekniskt system. Detta förefaller vara en följd av roboten som "förkroppsligat väsen" som vi uttryckte det i vår gemensamma diskussion efter försökets observationer.

Speciellt med avseende på återkopplingstiming var vårt system enligt vår bedömning undermåligt beroende på radionätverkets begränsningar i överföringen. Vårt trådlösa radiomodem till roboten skall kunna överföra 9600 bit per sekund, men beroende på mottagningsförhållanden kan denna överföringshastighet inte garanteras, dvs. kommunikationen kan i många fall bli mycket långsammare. Detta var en känd begränsning i vårt experiment och robotuppsättning hittills, men moderna radionät har idag förmåga att överföra data i en teoretisk hastighet av 11 Mb/sekund, dvs. datahastigheten kan vara över 1000 gånger snabbare. Vi hoppas att prova ett sådant system inom en snar framtid på vårt robotsystem och förväntar oss då att kommunikationsflaskhalsen inte finns längre och att reaktionen på ett kommando kan ske snabbare.

Andra observationer och idéer uppkom ur våra videoinspelningar samt intervjuer som hölls med försökspersonerna. Dessa har drivit prototyputvecklingen vidare. Ett återkommande problem var att försökspersonerna ville se robotens "framsida" och ha en förbättrad återkoppling på vad som händer. Detta kan delvis hänga ihop med att prototyprobotens utformning var ganska rund och utan klar front (se Figur 5) samt att vi tills dess saknade en modell av hur återkopplingen skulle kunna fungera explicit via robotens kropp eller rörelse. Med försöket i tanken diskuterade vi oss fram tills idéen med en förändrad robotpåbyggnad och CERO -karakteren (se avsnitten 5, 6.10 och 10.6).

Som kritiska synpunkter kan nämnas att simuleringen av systemets (även i detta tidiga utvecklingsskede) navigerings- och talfunktioner var mycket resurskrävande. Man kan vara tveksam till om man verkligen vinner tid på att testa med ett sådant simulerat system i stället för att prova att verkligen implementera och integrera funktioner på riktigt.

Datamängden som måste samlas in är stor och analysen i efterhand tar mycket lång tid om man vill detaljgranska vad som verkligen pågick under robotkörningen. Så finns bandade försöksdialoger i form av transkriptioner på mer än 50 sidor. En utvärdering i framtiden kanske behöver hjälpmedel som digitala videoredigerings- och analysverktyg för att man ska kunna bearbeta och utvärdera materialet i detalj.

Ändå kan försöket ses som lyckat. Vi kom igång med användartester av roboten och robotgränssnittet, observerade användare under körningen och hur de förhåller sig till roboten. Vi samlade in deras föreställningar kring robotinteraktion med tal som gränssnittsmodalitet och fick många idéer som vägledde vårt fortsatta arbete. Dessutom kunde vi som en biprodukt träna oss själva i testande av robotar med hjälp av tänkta slutanvändare, något som kommer att hjälpa oss även i framtidens utvärderings- och robotförsök.

5. Robotens fysiska utformning

Detta kapitel innehåller en sammanfattning av det arbete som bedrivits med den fysiska utformningen från och med "Wizard of Oz"-testet fram till presentationen för beställargruppen, AMS. I samband med "Wizard of Oz"-testet som gjordes i syfte att studera användarens beteendemönster, gjordes en enkel första modell (se t.ex. Figur 5 på sidan 13). Denna modell är inte att betrakta som ett designförslag utan hade som främsta syfte att kunna hålla de högtalare och sladdar och den lastkorg som krävdes för testets genomförande. Modellen gav trots detta ett antal reaktioner som varit till ledning i det fortsatta designarbetet, gällande såväl praktiska som emotionella aspekter.

Formen på robotens påbyggnad var cirkulär och så gott som samtliga användare besvärades av att det var svårt att utläsa robotens riktning. Anledningen till att riktningen inte betonats mer i modellen var att roboten inte kan upptäcka sin användare och därför var det önskvärt undvika en tydlig front eftersom denna ofta skulle hamna vänd bort från användaren. De största problemen låg dock i kommunikationen med roboten. En styrenhet som föreslagits tidigare, som var en kombinerad mikrofon och fjärrkontroll baserad på "walkie-talkie"-teknik (se figur 7), uppskattades inte eftersom användaren inte tyckte att det kändes naturligt att trycka in en knapp varje gång som roboten skulle anropas. En annan del som skapade osäkerhet var bristen på en direkt återkoppling på det användaren sa till roboten. I den fördröjning som sker medan talsystemet analyserar det sagda, behövs en direkt bekräftelse på att roboten dels har uppfattat att någonting har sagts, dels uppfattat eller inte uppfattat innebörden i vad som sagts.

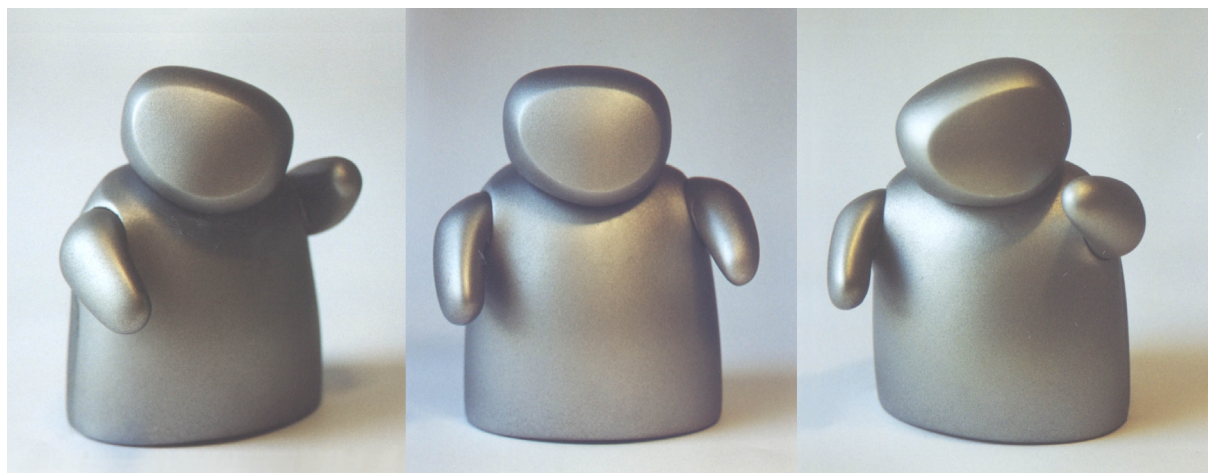


Figur 7. "Walkie-talkie"-liknande fjärrkontroll till roboten. Nödknapp och tumstyrd joystick används för att detaljstyra roboten. Övrig kontroll sker genom talad input (mikrofon).

Utifrån denna problematik har vi skapat en ny metafor för roboten där den "traditionella" roboten, såsom exempelvis R2D2, delats upp i två delar – en transportdel och en chaufför. Chauffören blir då en representant för robotens intelligens och blir ett fokus för interaktionen. Vi har skapat en figur som med hjälp av servomotorer kan röra sig och gestalta ett enkelt kroppsspråk. Därigenom kan man ge en återkoppling direkt då användaren sagt något till roboten och när roboten är färdig med sitt svar kan kroppsspråket användas för att förtydliga och levandegöra utsagan. Figuren har fått namnet CERO – *co-operative embodied robot operator*.

Genom att vi har använt oss av en separat figur har vi fått möjlighet att styra bilden av robotens personlighet. Eftersom roboten kommer att röra sig, tala och agera autonomt, kan vi utgå

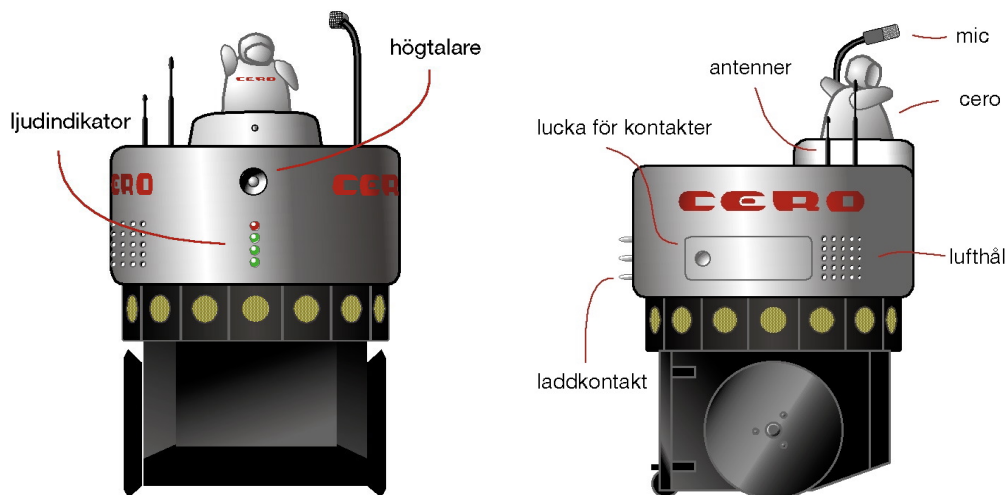
från att roboten kommer att uppfattas som "besjälad". Om vi skulle utveckla roboten utan stöd i chaufför/transportdel-metaforen skulle vi behöva ta hänsyn till en rad fysiska och mekaniska begränsningar när vi formger uttrycket. I den nuvarande uppdelningen har vi haft större möjlighet att precisera den personlighet roboten representerar. Användaren får också en tydlig motpart att fokusera på i interaktionen med roboten. Uttrycket är viktigt av dels psykologiska skäl, dels rent funktionella. Eftersom CERO inte har någon möjlighet till fysisk manipulation kommer CERO ta kontakt med personer och be dem om hjälp. Dessutom kan man inte förvänta sig att det talade gränssnittet kommer att fungera helt smärtfritt. Det är därför viktigt att figuren har ett uttryck som vädjar till användarens överseende. Roboten får naturligtvis inte heller uppfattas som ett hot.



Figur 8. CERO – Co-operative Embodied Robot Operator. En enkel mekanisk docka som med ett enkelt rörelsemönster förstärker känslan av interaktion mellan användare och robot.

Transportdelen av roboten är inte utformad i första hand för att skapa ett specifikt intryck (jfr. Figur 9). Där är det istället främst rationella motiv som ligger till grund för utformningen. Dels att delarna skall vara rimligt enkla att tillverka, dels att inga delar skall inverka på den plattform vi utgår ifrån. De delar vi behöver addera; mikrofon, CERO -figuren, kontakter etc. placeras i en del ovanpå roboten. Konstruktionsprincipen är att lämna den ursprungliga plattformen orörd (i synnerlighet att inte göra något som kan påverka navigationssystemets sonarer) dvs att inte göra några irreversibla förändringar. Plattformens prestanda får alltså inte påverkas och allt som monteras måste kunna tas bort utan att plattformen förstörs. Detta är naturligtvis en stor begränsning i utformningen men detta är inte någon väsentlig förlust eftersom robotens uttryck främst kommuniceras genom CERO -figuren. Roboten har på ovansidan en lucka för last.

Det normala scenariot är antagligen att man bara ställer ifrån sig det man vill ha transporterat på robotens ovansida. Luckan är därför försänkt så att lasten inte glider av. Det har under arbetsprocessen framkommit önskemål om ett låsbart fack på roboten för de fall då man vill transportera någonting värdefullt. Detta skulle dock lätt kunna ge upphov till en rad onödigt invecklade scenarier. Det kan dock vara befogat att inte behöva exponera det man vill transportera, och det är därför roboten har ett lastutrymme med lock istället för endast ett flak. Lastutrymmet utföres tätt så att tex. utspillt kaffe inte tränger ner i plattformsdelen. Lastdelen är placerad bakom robotens centrum eftersom plattformens styrgeometri gör att roboten tippar om den belastas i framkant. Den del av påbyggnaden som inte utgörs av lastutrymme kan tas i anspråk för microcontrollers etc. så länge delarna är lätta.



Figur 9. Vy av robot med transportenhet a) framifrån och b) från sidan.

På sidan av roboten sitter en lucka för anslutningar, dvs för att komma åt de kontakter som sitter på plattformens ovansida. Roboten – dvs plattformen, påbyggnaden och CERO -figuren – bör i sin helhet vara så låg som möjligt. Detta på grund av att hjulens placering medför en viss tipprisk. Det är dessutom bra om roboten får plats under ett bord, inte i första hand för att det kan vara trevligt att ha den där utan för att placeringen på sonaterna gör det svårt att upptäcka bordsskivor. Mikrofonen måste dock av akustiska skäl vara placerad tillräckligt högt för hålla nere avståndet till den som talar med roboten. Mikrofonen är placerad på en böjlig arm för att mildra effekten av en kollision med en bordsskiva. På roboten sitter också stift för laddning. De är placerade på robotens baksida så att roboten kan vara ”stand by” med fronten mot rummet när den laddar. Genom att flytta robotens transformator från plattformen till laddaren kan man bygga en lågspänningsladdare med en öppen struktur utan att skapa risk för elektriska olyckor. En sådan laddare kan utformas med en öppen struktur så att roboten lätt kan docka med den utan större krav på precision i navigering.



Figur 10. Förslag på utformning av stödhjul., med en bakre glidskena, för att kunna ta sig nerför trösklar.

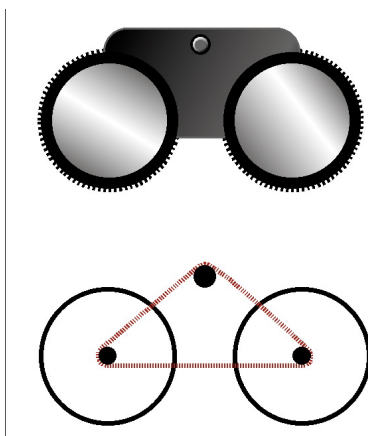
Ett svårlost problem med den robotplattform vi utgår ifrån är oförmågan att ta sig över hinder såsom mattkanter och trösklar. Detta beror på en kombination av den låga markfrigången, viktfordelningen och hjulens utformning och placering. Även om problemet har en praktisk karaktär snarare än en direkt betydelse för interaktionen, är det en uppenbar brist. Det kommer att bli svårt att testa interaktionen under naturliga förhållanden och bristen påverkar antagligen också förhållningssättet till roboten. Vi riskerar att i stället för en hjälprea skapa en

kontorets flygande holländare som aldrig når sin egentliga destination. De praktiska problemen är dock de som väger tyngst. Roboten måste antagligen låsas in om nätterna och måste laddas emellanåt. Att bära ekipaget över en tröskel varje morgon och kväll är inte i analogi med vår ambition att skapa ett handikapphjälpmedel.

Det försök som företagits för att få roboten att komma över ett hinder har inte fallit väl ut. Idéen byggde på att med hjälp av glidskenor och ett klättrande hjul använda robotens fart framåt till att glida över tröskeln. Principen fungerade i viss utsträckning men inte med tillräcklig tillförlitlighet. För att komma över en tröskel krävdes hög hastighet och roboten landade på ett icke kontrollerbart sätt vilket i längden skulle utgöra ett allvarligt slitage. Principen var inte heller förenlig med det beteende som roboten har för att ta sig igenom dörröppningar och smala passager – att åka långsamt fram för att hinna samla in noggranna sonardata.

I en utveckling av projektet vore det värdefullt att förbättra plattformen så att roboten klarar av hinder på ett kontrollerat sätt. Man måste dock vara medveten om att om man ändrar hjulbasen påverkar detta robotens indata och navigeringsförmåga. Detta är alltså ett projekt man kan ge sig in i först när man har tid och resurser att göra justeringar i navigationssystemet. Det kan dock gå att göra förändringar av styrgeometrin utan att behöva ändra grunderna för styrsystemet.

För att förklara vad som kan bli ett problem och hur detta kan undvikas krävs en kort beskrivning av principen för robotens nuvarande styr- och navigeringssystem. Robotens sonarer är ordnade i en cirkel. Cirkelns centrum är utgångspunkten för robotens orientering. Roboten är också beroende av att veta hur hjulen rör sig för att hålla reda på sin position. Hjulen är placerade i mitten av roboten för att roboten, när den svänger, skall rotera runt just den punkt som är centrum för robotens navigering. Hjulens placering i mitten är en förutsättning för att få en samverkan mellan sonarer och styrning och denna hjulplacering är förklaringen till att roboten har en benägenhet att tippa framåt. För att kompensera den dåliga balansen har man försökt placera så mycket av robotens tyngd så lågt som möjligt vilket i sin tur förklarar den låga markfrigången.



Figur 11. Förslag på alternativ framdrivningsmekanism, baserad på fyra drivande hjul.

Figur 11 beskriver ett förslag till fyrhjulsdraft som skulle kunna göra det möjligt att ta sig fram i en normal kontorsmiljö. Även denna del av projektet är anpassad för att vara reversibel. Hjulen sitter infästa på plattformens undersida, kraften överförs via rem eller kedja från den gamla hjulaxeln. Utväxlingen på drivhjulet anpassas för att kompensera den nya hjulstorleken. Hjulens placering är symmetrisk så att centrum för rotationen förblir i överensstämmelse med centrum för sonarerna. I huvudsak kan man göra dessa förändringar utan att roboten

”märker” någon skillnad. Det som måste korrigeras är dock att roboten får mer friktion sidledes och måste instrueras att ”ta i” mer när den svänger, något som inte är oöverstigligt att ändra i mjukvaran. Med denna lösning får man en robot som kan komma över trösklar och som inte är benägen att tippa.

6. Systembeskrivning

Mjukvarusystemet hos roboten är uppbyggt som ett distribuerat (geografiskt uppdelat) system där de olika ingående delarna har skilda ansvarsområden och samarbetar med andra komponenter för att planera och uppnå sina mål. Systemet använder en CORBA-baserad arkitektur för kommunikationen mellan komponenterna (se avsnitt 6.2 om CORBA).

Genom att använda en enhetlig systemarkitektur underlättas modifiering och utbyggnad av systemet. Komponenter kan läggas till, modifieras och tas bort beroende på användningsområde och robot. Då systemet är distribuerat kan de olika komponenterna spridas ut på olika datorer i ett nätverk efter behov.

För att kunna ta fram ett system som planerar och utför uppgifter krävs det att systemet har någon form av förståelse av uppgifterna, uppgifternas uppbyggnad och utförande. Systemet behöver därför en kunskapsrepresentation. De uppgifter som systemet utför är enkla transportuppgifter samt interaktion med användare. Interaktionen är i vårt fall en mycket väsentlig uppgift då systemet behöver hjälp av användare för att utföra uppgifterna.

Roboten är tänkt att transportera föremål mellan olika platser i en kontorsmiljö. För att klara detta krävs att systemet har en internrepresentation av platser och föremål samt hur systemet kan interagera med dessa genom att ta sig mellan olika platser och hämta och lämna föremål. Systemet behöver dessutom förstå vad användaren vill att systemet skall utföra. Dessa uppgifter kan man dela in i två olika kategorier av uppgifter, nämligen navigationsuppgifter samt interaktionsuppgifter.

6.1.Distribuerade system

Distribuerade system omfattar tekniker och metoder för konstruktion av autonoma och geografiskt åtskilda datorsystem som samarbetar för att nå ett gemensamt resultat. Dessa system är sammankopplade via kommunikationsnät.

Ett distribuerat system består av ett antal delsystem där varje del har hand om sin egen information men också kan dela med sig av sin information till andra delsystem. Denna spridning av systemet är genomskinlig för användaren. Detta betyder att från användarens synvinkel märks det inte att systemet är uppdelat utan denna uppdelning och interna kommunikation sköts utan användarens inblandning eller kännedom.

6.2.CORBA

CORBA (Common Object Request Broker Architecture) (OMG, Siegel 2000) är en standard för att skapa distribuerade objektbaserade system oberoende av programmeringsspråk, datorarkitektur och operativsystem. Förutom distribuerad kommunikation mellan objekt tillhandahåller CORBA även en generell distribuerad systemarkitektur med en mängd fördefinierade tjänster, antingen generella tekniska tjänster som namntjänster, händelsehantering mm eller domänspecifika tjänster inom områden som finans, tillverkning mm. Genom att använda sig av CORBA som grund för systemkommunikationen förenklas arbetet med att skapa ett distribuerat system då CORBA hanterar och underlättar mycket av arbetet kring skapandet av distribuerade system. Grunden i CORBA är dess ORB (Object Request Broker) som ser till att objekt blir distribuerade och kan anropa varandra. För att underlätta för systemet att finna rätt

objekt använder systemet sig av CORBAs Namntjänst (Name Service) som låter objekt finna varandra genom att använda namn.

6.3.Kunskapsrepresentation

Då systemet är tänkt som hjälp för att hämta och bära föremål, stödjer systemet följande typer av uppgifter:

- gå till en plats,
- ta emot ett föremål,
- lämna ett föremål,
- transportera ett föremål mellan två platser.

Ur dessa typer av uppdrag kan vi särskilja ett antal operationer som systemet måste klara av samt vilka element som systemet behöver ha kunskap om (ha en representation av) för att kunna utföra sina uppgifter. Systemet behöver dessutom en kunskapsrepresentation av platser och föremål för att kunna utföra sina uppgifter. Platser behöver roboten kunskap om för att kunna navigera och utan kunskap om föremål kan roboten inte ta emot eller leverera objekt. Det är tveksamt om det krävs kunskap om personer för att utföra uppgifterna då den nuvarande robotprototypen inte har någon möjlighet att särskilja användare.

Operationer	Objekt
1. förflytta sig mellan två platser	1. platser
2. ta emot föremål	2. föremål
3. leverera föremål	3. roboten

Tabell 1 Operationer systemet behöver klara av

Tabell 2 Objekt som systemet behöver känna till

Utifrån de begränsningar systemet har (t ex avsaknad av manipulationsförmåga) behöver roboten ha hjälp av någon person för att kunna utföra sina uppgifter. Systemet behöver samarbeta med användare för att kunna uppnå sina mål. Det är möjligt att systemet i framtiden får tillgång till extra utrustning och färdigheter och kan skulle då bli mindre beroende av människor för att lösa sina uppgifter.

Dessa element och operationer har vissa egenskaper och förhållanden. Objekten har t ex följande:

- föremål kan befinna sig på en viss plats eller på roboten,
- roboten kan befinna sig på en viss plats eller vara på väg mellan två platser.

Följande gäller för operationerna:

- roboten kan förflytta sig mellan två olika platser,
- roboten kan ta emot ett föremål om både föremålet och roboten befinner sig på samma plats,

- roboten kan leverera ett föremål till en viss plats om roboten befinner sig på platsen och har föremålet.

Dessa operationer är i sig komplicerade att utföra, men kunskapen om hur operationerna utförs delegeras till delar av systemet som har en mer avancerad representation av de operationer och element som de har ansvar för och utför.

Då systemet är uppdelat på olika delkomponenter behöver de kommunicera med varandra på ett sådant sätt att de förstår varandra. Därför använder sig systemet av en enhetlig kunskapsrepresentation även om de olika komponenterna sedan inte behöver eller använder hela representationen. Det är t ex inte nödvändigt att alla delar har en komplett representation av en bok utan det kan räcka med att veta att det är ett föremål. För att komponenterna skall kunna delegera uppgifter och skicka information tillvarandra behövs också en representation av vad som är en uppgift och vad som är kunskap. För detta så har systemet också en representation av uppgifter, kunskap och förfrågningar.

En uppgift är antingen ett mål som behöver uppnås t ex en viss bok skall vara i en viss bokhylla. Men den kan också vara mera specifik genom att uppgiften beskriver vad som skall utföras t ex placera en viss bok i bokhyllan. I det första fallet är det upp till komponenten hur den löser uppgiften medan det andra fallet talar om hur uppgiften skall lösas.

Kunskap är information om tillstånd och används för att uppdatera vad komponenterna vet om varandra, roboten och omvärlden. T ex är robotens nuvarande position kunskap som andra komponenter kan vara intresserade av.

En förfrågan är en begäran om specifik kunskap hos någon annan komponent. T ex vad en viss komponent har för uppgift eller på vilken plats roboten befinner sig just nu.

6.4. Prototypsystemet

Följande komponenter ingår i systemet och beskrivs kortfattat – se Figur 12 nedan.

Kontroll

Kontrollkomponenten är den centrala delen av systemet som planerar och ser till att uppgifterna utförs.

Grafiskt användargr nssnitt (GUI-komponent)

Låter användare övervaka och kontrollera roboten.

Interaktion

Ansvarar för den direkta interaktionen med användare.

Navigation

Sköter kontakten med den fysiska roboten. Ser till att roboten går dit den skall.

CERO

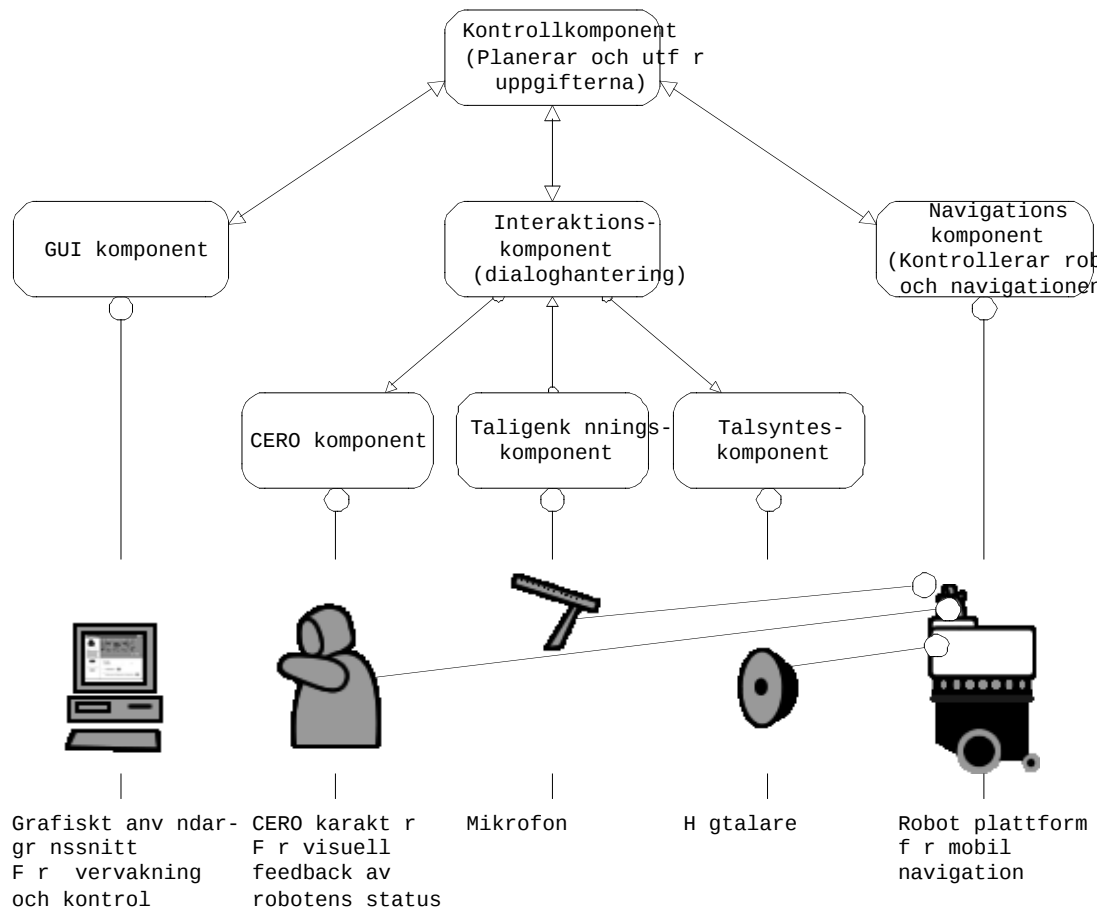
Underställd interaktionskomponenten och kontrollerar CERO-karaktären.

Taligenk nning

Tolkar det användaren säger och skickar detta till interaktionskomponenten.

Talsyntes

Skickar ut de meddelanden den får ifrån interaktionskomponenten som tal.

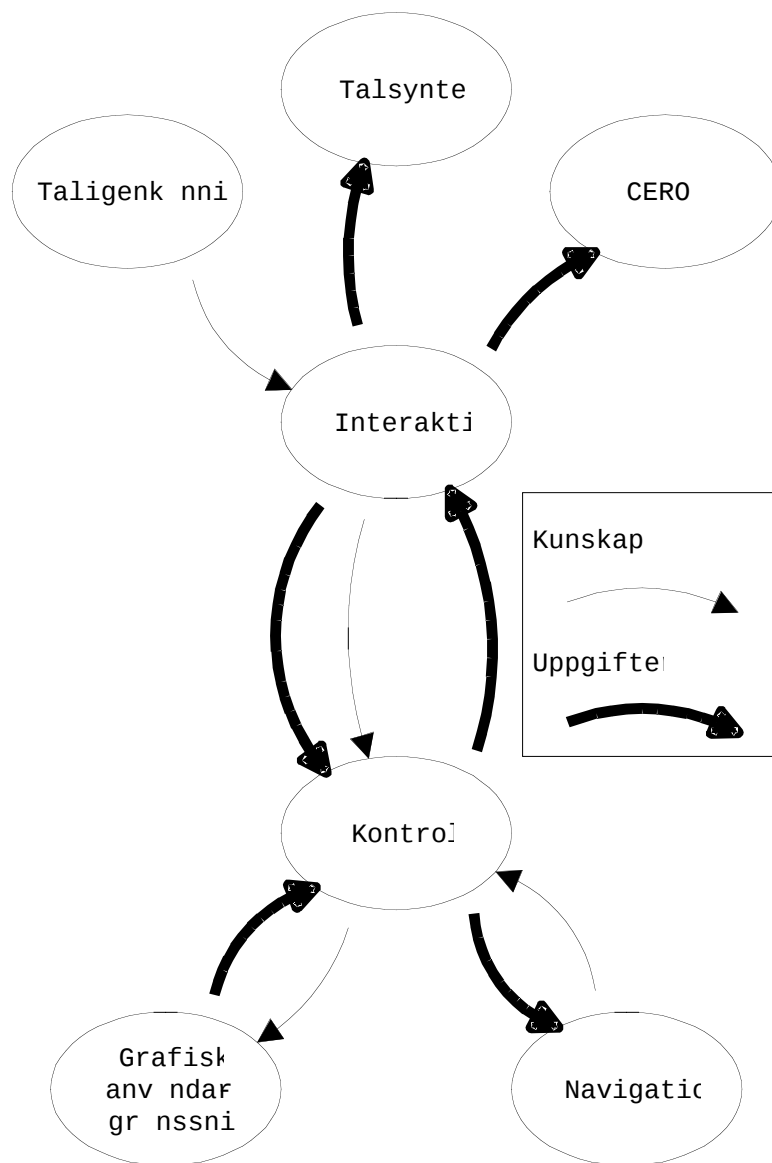


Figur 12. Systemöversikt för servicerobotprototyp.

Komponenterna kommunicerar med varandra genom att skicka information sinsemellan. Informationen som komponenterna skickar mellan varandra avser uppgifter, kunskap om tillstånd och förfrågningar. De olika komponenterna är ansvariga för att utföra olika typer av uppgifter och har kunskap om sitt område. De kan besvara förfrågningar om sig själv och den kunskap som komponenten har. Denna kommunikation leder till ett informationsflöde mellan de olika komponenterna enligt Figur 13 nedan. I kommunikation särskiljer vi tre olika informationstyper, dessa är:

- **Uppgifter.** En uppgift är något som skall utföras av komponenten. Det kan t. ex. vara att förflytta sig, ta emot föremål eller avbryta en uppgift.
- **Kunskap.** Kunskap är information om komponenten, roboten eller dess omgivning. Komponenterna kan dela med sig av sin kunskap till andra komponenter. Det kan gälla t.ex. var roboten befinner sig eller att ett visst föremål har tagits emot.

- **Förfrågningar.** Om någon komponent är i behov av någon information kan den skicka ut en förfrågan t.ex. för att få reda på var roboten är eller om den är på väg någonstans.



Figur 13. Systemets informationsflöden mellan komponenterna

6.5. Informationsflöden

Under utförandet av uppgifter samarbetar de olika komponenterna med varandra genom att skicka uppgifter som behöver utföras och kunskap om vad som sker (på grund av yttre omständigheter eller som ett resultat av utförandet av uppgifter).

6.6. Kontroll

Systemets centrala del kallas för *Kontrollkomponenten*. Den planerar och samordnar de övriga komponenterna för att utföra uppdrag. Komponentens består av två delar, en *planerare* och en *exekverare*. Planeraren tar utifrån uppgift och begynnelsestillstånd fram en plan bestående av en serie uppgifter. Exekveraren försöker sedan utföra dessa uppgifter genom att delegera dem

till de övriga systemkomponenter som är ansvariga för respektive uppgift. Under tiden som uppgifterna utförs väntar exekveraren på svar om hur utförandet av uppgiften gick för att veta om nästa uppgift kan utföras eller om uppdraget misslyckats.

Uppgifter

Systemet hanterar 3 typer av uppgifter. Dessa är:

- leverera ett föremål mellan två platser,
- gå till känd plats,
- stanna (avbryt).

Dessa uppgifter utförs när de kommer in, dvs om Kontrollkomponenten tar emot en ny uppgift medan den utför ett annat uppdrag kommer det gamla uppdraget att avbrytas och den nya att utföras. Berörda komponenter informeras om vad som skett. Detta är en förenkling då vi antar att vi endast har en primär användare som skickar roboten på ett uppdrag i taget. Om flera användare skulle få tillgång till roboten är det kanske inte lämpligt att ignorera tidigare order utan systemet behöver prioritera och köa uppgifterna för att utföra dessa i en lämplig ordning.

Kunskap

Kontrollkomponenten hanterar systemets övergripande kunskap. Den tar emot information från övriga delar för att kunna samordna utförandet av användarnas uppgifter. Till sin hjälp har Kontrollkomponenten kunskap om var roboten och olika föremål befinner sig samt vad de övriga delarna av systemet har för status. Denna kunskap delar Kontrollkomponenten med sig av till de övriga komponenterna.

6.7.Navigation

Navigationskomponenten hanterar kommunikationen med den fysiska roboten (motorer, sensorer osv). Komponentens utför därmed navigationsuppgifter och håller systemet uppdaterat om robotens fysiska position och status.

Navigationskomponenten hanterar de uppgifter från kontrollkomponenten som gäller systemets navigation. Detta hanteras genom att skapa en anslutning till ISR-systemet (Andersson m fl. 1999) på roboten (se avsnitt om Robotkontroll). Via denna anslutning kan komponenten skicka uppgifter och ta emot information från ISR-systemet som kontrollerar robotens styrsystem, som ser till att roboten förflyttar sig.

Den implementation av ISR-systemet som vi använder oss av består av följande delar:

- lokaliseringsdel som använder sig av sonarkartan för att hålla robotens position uppdaterad,
- planeringsdel som givet ett kommando planerar och utför kommandot,
- interaktionsdel som tillåter vårt system att kommunicera med ISR-systemet.

Vi använder oss av följande funktioner hos ISR-systemet:

- skicka roboten till olika fördefinierade platser,
- stoppa roboten,
- fråga ISR-systemet om robotens status (t ex vilken position roboten har, vad roboten har för uppdrag).

Dessutom skickar ISR-systemet ut information vid olika händelser t ex då:

- roboten anländer till en plats och uppgiften är utförd,
 - den har mottagit en order om att gå till någon plats,
 - roboten har stannat.

Uppgifter

Navigationskomponenten hanterar följande uppgifter:

- gå till känd plats,
- stanna.

Dessa uppgifter utförs när de kommer in, dvs om roboten tar emot en ny uppgift medan den utför en annan uppgift kommer den gamla uppgiften att avbrytas och den nya att utföras. Navigation kommer dock att informera berörda komponenter om vad som skett.

Kunskap

Navigationskomponenten har kunskap om var roboten befinner sig och vilken status roboten har (t.ex. att den står stilla eller är på väg till en viss plats). Denna information sprider Navigationskomponenten till berörda delar av systemet.

6.8.Grafiskt användargränssnitt

GUI-komponenten (*GUI* betyder *Graphical User Interface*) ger en grafisk representation av systemets tillstånd (t ex var roboten är och vad den håller på med) samt tillåter användaren att avbryta och ge roboten uppgifter. Denna komponent presenterar den kunskap som den får från Kontrollkomponenten för användaren samt låter användaren skicka uppgifter till systemet. Detta medför att denna komponent har i stort sett samma kunskap om robotens tillstånd som Kontrollkomponenten. Det grafiska gränssnittet beskrivs närmare i kapitel 9.

6.9.Interaktion

Interaktionskomponenten har i sin tur ett antal komponenter under sig för att interagera med omvärlden. Huvuddelen i interaktionskomponenten är dialoghanteraren som utför interaktionsuppgifter genom att använda sig av sina externa interaktionskomponenter (CERO, taligenkänning och talsyntes). Interaktionskomponenten tar emot instruktioner från användaren och vidarebefordrar dessa till kontrollkomponenten. Se avsnitt 6.6 för närmare beskrivning.

Uppgifter

Interaktionskomponenten hanterar 2 typer av uppgifter. Dessa är:

- ta emot ett föremål,
- leverera ett föremål.

Kunskap

Kunskapen som Interaktionskomponenten hanterar är de olika platser som föremål befinner sig på. Interaktionskomponenten informerar intresserade delar av systemet var föremål befinner sig då den får reda på det.

6.10 CERO

CERO-komponenten använder sig av systemets lilla modelldocka för att ge användare visuell återkoppling under dialogen. För en mera detaljerad beskrivning se avsnitt om CERO.

Uppgifter

CERO-komponenten har endast en uppgift och det är att utföra de gester som den blir tillsagd att utföra.

6.11 Talsyntes

Talsyntes genererar systemets utgående tal under dialog, se avsnitt om talgränssnitt för detaljer.

Uppgifter

Genererar det tal som den blir tillsagd att säga.

6.12 Taligenkänning

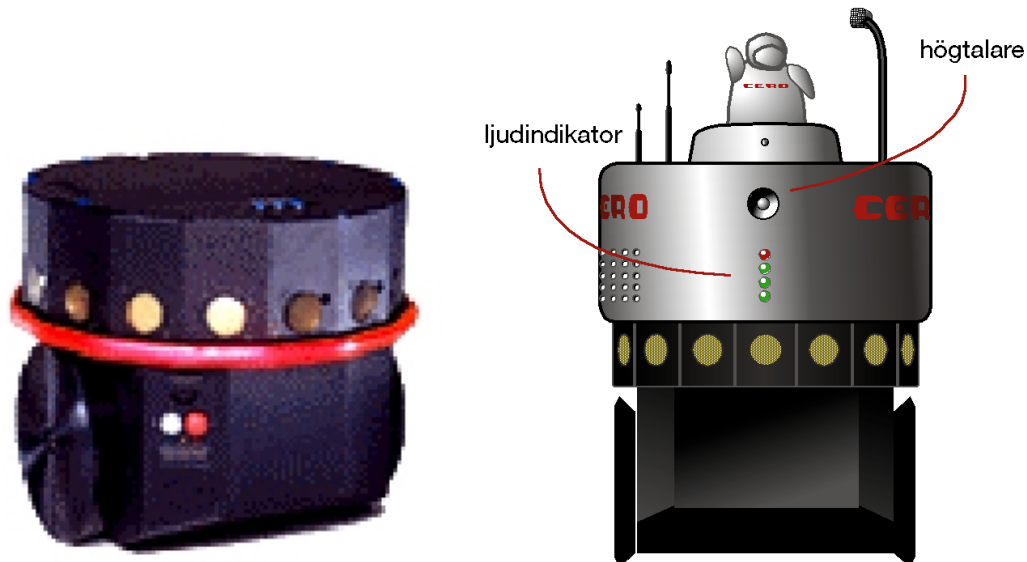
Taligenkänningen tolkar det användaren säger under dialog och skickar detta till interaktionskomponenten. Funktionen beskrivs närmare i talgränssnittsdelen av rapporten.

Kunskap

Taligenkänningen försöker tolka de ljud den får in. Om komponenten förstår vad som sägs skickas den informationen till dialoghanteraren i Interaktionskomponenten.

7. Robotplattform

Den robotplattform som används i detta projekt, är en Nomadic SuperScout II, en kommersiellt tillgänglig plattform som främst används inom mobilrobotikforskning.



Figur 14. Robotplattform samt robotprototypen med påbyggnad

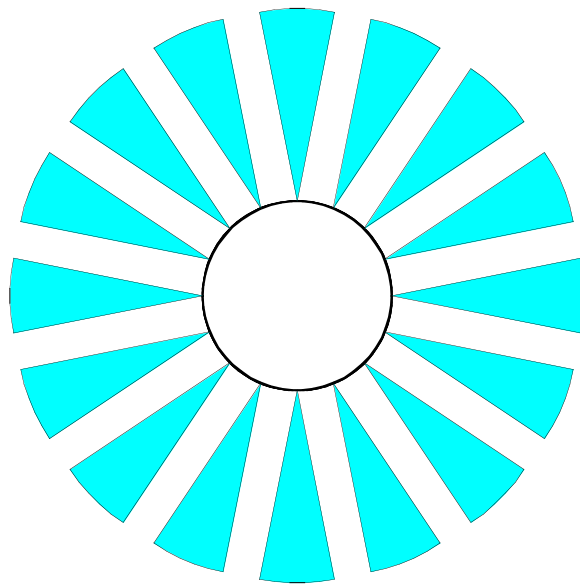
Scoutroboten är 35 cm hög och har en diameter på 40 cm. Den väger 25 kg varav ca hälften av vikten utgörs av batterier. Robotens markfrigång är endast 1,5cm och det i kombination med hjulens konstruktion och placering går det omöjligt för roboten att ta sig över även små hinder t ex trösklar (se även kapitel 5). Den är utrustad med 16 st ultraljudssensorer och en odometer (en mätare som, baserat på hjulens rörelse, håller reda på position och riktning). Vidare är den utrustad med en 233 Mhz Pentium dator och radiomodem för kommunikation med övriga delar av systemet. Dessutom har en påbyggnad iordningstälts innehållande mikrofon, högtalare, lastutrymme och annat (se kapitel 5). Med påbyggnad är roboten 70 cm hög och 45 cm bred.

7.1.Odometer

Robotens odometer används för att hålla reda på position och riktning i omgivningen. Position och riktning beräknas genom att det sitter en räknare på varje motor som räknar hur mycket hjulen vrids. Utifrån detta beräknas en position och en riktning fram. Tyvärr är odometern på plattformen ganska inexakt p.g.a. hjuldimensionerna och andra omständigheter. Detta medför att positions- och riktningsbestämningen efterhand som roboten rör sig, kommer att bli allt mindre exakt. Redan vid ganska små fel i riktningsbestämningen kommer t.ex. felet i positionsuppskattning att bli stort om roboten rör sig över längre sträckor. Detta brukar benämnas *drift*.

7.2.Ultraljudssensorer

Robotens primära sensorer är 16 st ultraljudssensorer som används för att mäta avstånd till föremål i robotens omgivning. Sensorerna är placerade symmetriskt i en ring runt roboten och skickar ut ultraljudspulser enligt figur nedan.



Figur 15. Sonarernas placering och ljudpulsernas utbredning.

Genom att sända ut ljudpulser och ta tid på när ljudpulsens reflektion träffar sensorn beräknas avståndet till närmaste föremål i sensorns riktning. Sensorerna klarar av att mäta upp avstånd på mellan 15 och 650 cm med en noggrannhet på 2,5 cm. Ju längre tid det tar för reflektionen att återvända desto längre bort befinner sig föremålet. Då varje ultraljudssensor både sänder ut och tar tid på ljudpulsens innebär det att under tiden som ljudpulsens sänds ut kan sensorn inte lyssna. Detta medför att kortare avstånd än ca 15 cm, den sträcka som ljudpuls och reflektion hinner färdas under tiden som pulsen skickas, inte kommer att mätas. För att föremål skall registreras av ultraljudssensorerna krävs det att tillräckligt mycket av ljudpulsens reflekteras tillbaka till sensorn. Reflektionen beror på föremålets material, form och med vilken vinkel ljudpulsens träffar ytan. Släta ytor reflekterar ljudpulsens i en smalare vinkel än skrovliga ytor som sprider reflektionen mer. Raka ytor reflekterar ljudpulsens i en smalare vinkel än små rundade ytor som sprider reflektionen mer.

7.3.Framdrivning

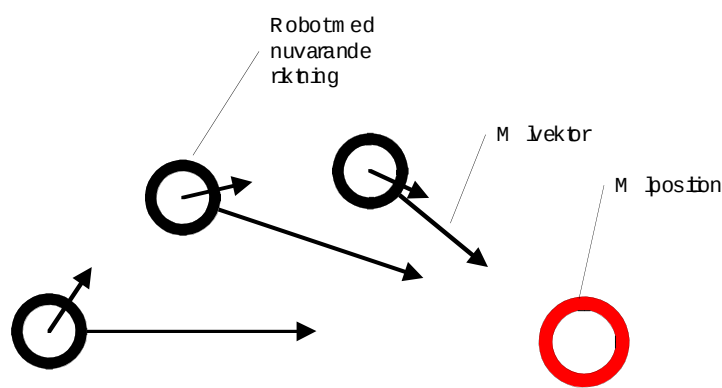
Robotens drivsystem består av två stycken centralt placerade drivhjul samt ett bakre litet stödhjul. De två drivhjulen drivs av separata motorer som kan ha av varandra oberoende hastighet och riktning. Detta i kombination med det faktum att robotens hjulbas ligger i centrum innebär bland annat att roboten har en vändradie på 0 cm d.v.s. roterar på stället.

8. Robotkontroll

Robotens styrsystem ISR (Intelligent Service Robot) är utvecklat vid KTH på Centrum för Autonom System (CAS) (Andersson m fl. 1999). ISR-systemet utnyttjar robotens sonarer för att uppdatera robotens position utifrån en intern karta. Detta löser de problem man annars skulle ha med robotens positions drift under navigation. Dessutom har systemet ett kollisionsundvikningssystem för att undvika hinder under navigation.

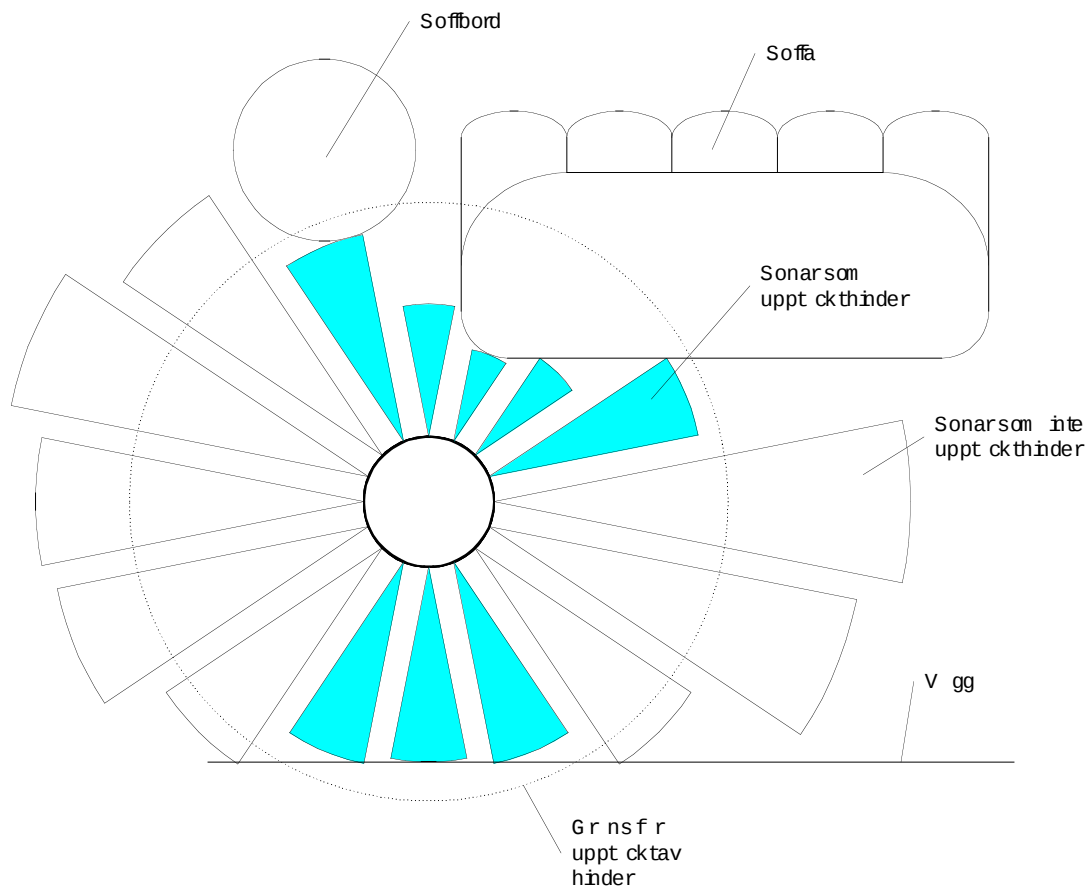
8.1.Navigation

Roboten navigerar genom att utifrån en målpunkt och sin egen position beräkna en vektor, dvs. riktning och avstånd, till målpunkten. Allteftersom roboten rör sig uppdateras den egna positionen och målvektorn.



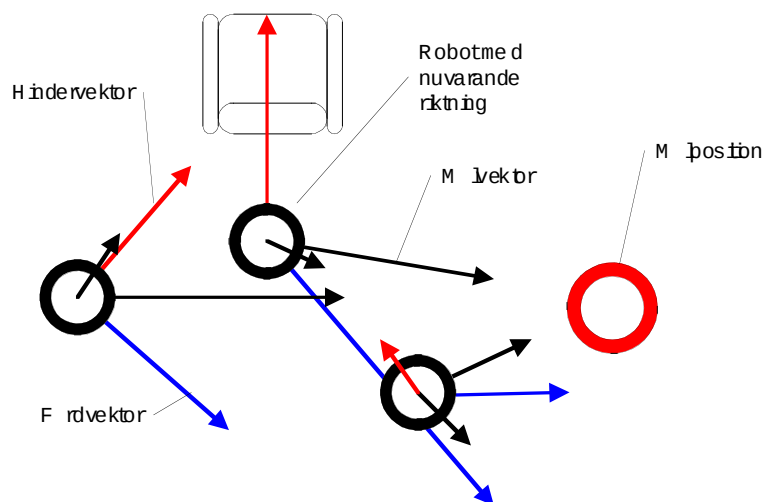
Figur 16. Uppdatering av målvektor utifrån position

Då det kan förekomma hinder mellan roboten och målpunkten, både fasta och rörliga, måste roboten kunna undvika dessa under sin färd. Genom att använda ultraljudssensorerna kan roboten skapa sig en grov bild av vilka hinder som finns i omgivningen. Endast föremål som finns i närheten av roboten räknas som hinder.



Figur 17. Upptäckta hinder vid färd mellan en väg och en soffa med bord

Dessa hinder påverkar roboten genom att systemet beräknar en hindervektor till det närmaste hindret (ju närmare hindret roboten befinner sig desto större blir vektorn) och tar fram en önskvärd färdvektor genom att ta målvektorn och dra bort hindervektorn.



Figur 18. Uppdatering av riktningvektor med hänsyn till hinder

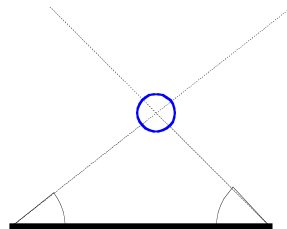
I smala utrymmen, där hinder förekommer på flera sidor, måste hänsyn tas till flera hinder. Då ultraljudssensorerna inte kan mäta kortare avstånd än 15 cm försöker roboten att undvika att

komma för nära hinder. Om roboten kommer för nära ett hinder vänder den sig bort från hindret och avlägsnar sig så att avståndet överskrider 15 cm. Detta kan i vissa fall ge upphov till att systemet kommer i något slags självsvängning och därvid uppvisar ett underligt beteende.

Kända målpunkter finns definierade i en karta där de olika rummen finns beskrivna med de målpunkter som finns i rummet samt angränsade rum. Systemet med målpunkter betyder att för att kunna navigera mellan punkterna måste det vara mer eller mindre fri sikt mellan punkterna. Mindre korrigering klarar systemet av genom sin kollisionsundvikning men om vägen är blockerad kommer roboten inte fram. För att kunna navigera mellan rum tar systemet reda på vilka rum som systemet måste navigera mellan för att komma fram till målet. Detta sker utifrån dess topologiska karta där det finns beskrivet hur rummen sitter ihop.

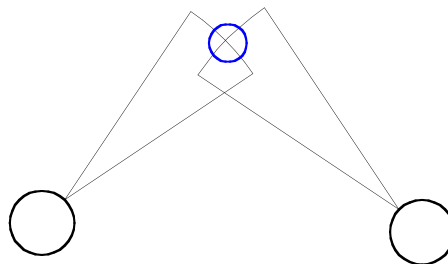
8.2.Lokalisering

Då odometern inte har tillräckligt stor noggrannhet för att användas för att lokalisera roboten krävs det att roboten själv kan uppdatera sin position då avdriften blir för stor. Till detta använder systemet sina ultraljudssensorer och en karta med välkända landmärken. Landmärken är för roboten positioner i rummet som ger tydliga ultraljudsreflektioner t ex olika hörn på möbler och dörrposter. För att ultraljudsreflektioner skall vara tydliga krävs det att de kan upptäckas under längre tidsperioder från olika positioner. För att leta efter tydliga landmärken använder sig roboten av ultraljudstriangulering. Tekniken har sitt ursprung i vanlig triangulering där man utifrån en känd baslinje, vars längd man känner till, kan positionsbestämma en punkt genom att mäta upp vinklarna mellan baslinjen och syftningslinjerna mot punkten och utifrån detta bestämma punktens position.



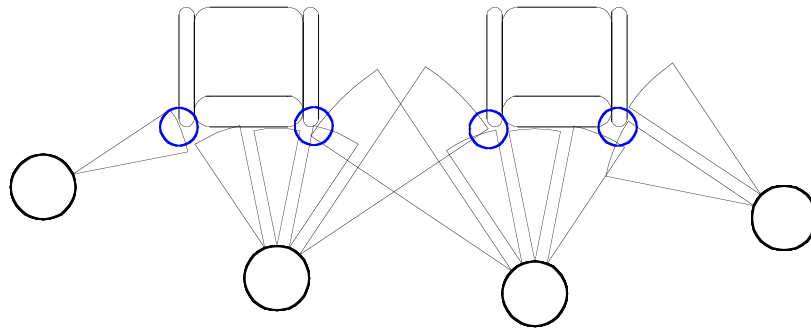
Figur 19. Vanlig triangulering

Till skillnad från vanlig triangulering som utgår ifrån vinklarna i förhållande till baslinjen använder sig ultraljudstriangulering sig av avståndsmätning förutom vinkel och baslinje. Detta på grund av den dåliga noggrannheten i vinkelbestämning och utseendet på ljudpulsen, vi måste ta hänsyn till att vi inte använder syftningslinjer utan cirkelsegment. Punktens position bestäms av skärningen mellan bågarna på cirkelsegmenten.



Figur 20. Ultraljudstriangulering

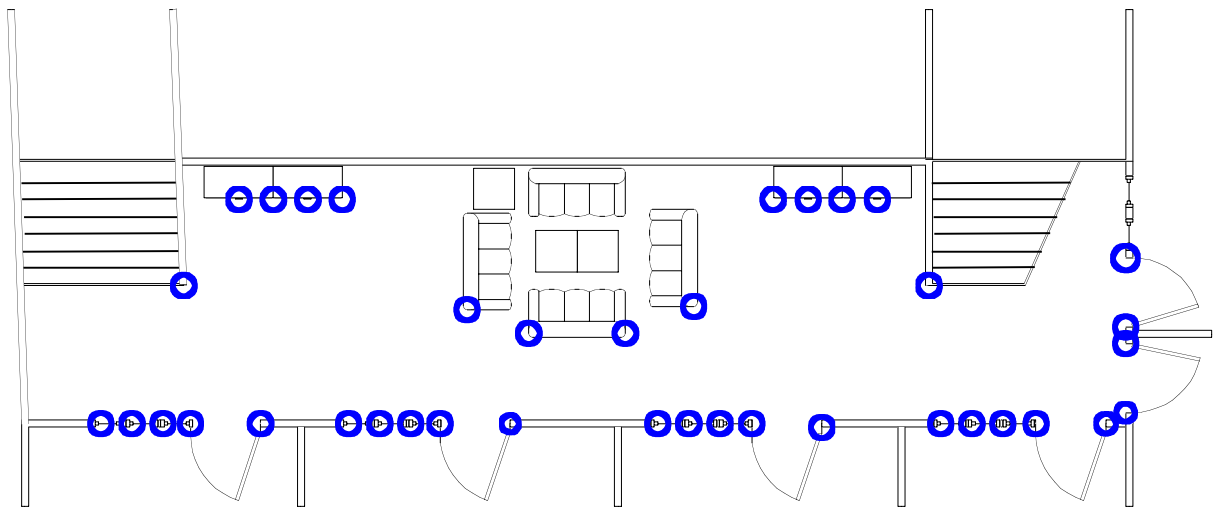
Under navigation samlas trianguleringsdata in och ur dessa tas landmärken fram. Systemet håller sedan reda på dessa landmärken. De landmärken som systemet kommer att använda sig av är de som roboten kan upptäcka från olika platser; vanligtvis är detta hörn och liknande.



Figur 21. Landmärkesinsamling

För att kunna använda sig av dessa insamlade landmärken har roboten en karta med kända landmärken som den jämför med de insamlade landmärkena. När systemet lyckas matcha insamlade landmärken med kartan uppdateras robotens position.

ISR-systemets karta lagrar stabila landmärken rumsvis och under körning byter systemet karta då den byter rum. Systemet har funktioner för att skapa dessa kartor själv men då det är implementerat för en annan robot, Nomadic 200 som är större och mer avancerad än vår, hade vi en del problem med detta vilket i slutändan resulterade i att vi skapade en egen landmärkeskarta utifrån för hand uppmätta punkter i omgivningen.



Figur 22. Landmärken som roboten lokaliserar sig efter

För att kunna navigera mellan målpunkter använder sig ISR också av en topologisk karta. Denna karta beskriver omgivningen med de målpunktioner och passager till angränsande rum som varje rum innehåller. I denna karta beskrivs målpunkterna med namn och absolutposition i rummet. Alla positioner i ett rum har unika namn liksom kartans alla rum har unika namn. Detta gör att man kan specificera platser med samma namn i flera olika rum. Genom att specificera både rum och målpunktion kan man bestämma vilken plats som menas. T ex om flera rum har en papperskorg specificerad kan man bestämma att det är till papperskorgen i köket roboten skall gå, genom att specificera målet som papperskorgen i köket.

9. Grafiskt Gränssnitt

I det här avsnittet kommer det grafiska gränssnittet (eng. *Graphical User Interface*, GUI) för vår servicerobot att beskrivas. Vi beskriver vilka modeller, processer och antaganden som använts, hur gränssnittet utvecklades i detalj, vilka resultat som har kommit fram i olika iterationssteg och användartester och hur en ökad kunskap kring hjälpmedelsrobotik bidrog till den nuvarande robotgränssnittsversionen. En detaljerad beskrivning och diskussion av resultatet med exempel, samt en utblick över intressanta frågeställningar avslutar kapitlet.

9.1.Introduktion

Gemensamt för system med ett grafiskt gränssnitt från ett människa-datorinteraktionsperspektiv är att de bjuder på en visuell modalitet för information som skall kommuniceras till användaren eller mellan olika användare med hjälp av teknik. Denna teknik kommunicerar å ena sidan till användaren vilka tillstånd som är aktuella och å andra sidan ger den information om vilka aktioner som kan krävas från operatören i nästa steg. Det grafiska gränssnittet är på det sättet ett medel för en maskin att visa sitt nuvarande tillstånd samt att erbjuda en visuell, lättförståelig representation av de objekt som skall interageras med, tex. ikoner som symboliserar tillståndsändringar. I ett sådant system är det grafiska gränssnittet följaktligen en komponent som måste ses i samband med situationen som apparaten/tjänsten används i, vilka uppgifter som skall utföras och vilka användare man räknar med. Stöd av andra media (t.ex. ljud eller tal), sätt att manipulera det grafiska gränssnittet, dvs. handhavande av t.ex. mus eller tangentbord baserat input är också viktiga grundfunktioner i moderna grafiska gränssnitt.

9.2.Designansats: teori och process

Den traditionella systemdesignen omfattar även utvecklingen av användargränssnitt. En förhärskande ingenjörsteknisk utvecklingsmetod utgår från den s.k. *vattenfallsmetoden*. (se t.ex. Sommerville 1996), dvs. en uppföljning av olika steg som leder till den slutliga produkten. Huvudkomponenter i en sådan utvecklingsmodell är framställande av (1) kravspecifikationer, (2) design av systemarkitektur, (3) detaljerad design, (4) produktion/ kodning och enhetstester, (5) integration och systemtester, samt (6) operation och service under drift. Denna metod har också i stor utsträckning använts inom mjukvaruframställning.

Samtidigt som de sex nämnda stegen fortfarande utgör viktiga delmoment i en utveckling så har forskning i människa-datorinteraktion (MDI) bidragit med viktiga förändringar i metoden: För det första finns det fördelar med att bryta upp den sekventiella metodiken genom ett återkopplande iterativt arbetssätt, dvs. stegen följer inte bara efter varandra, utan upprepas flera gånger. Det leder fram till en s.k. spiralformad modell av utvecklingen. Två huvudfördelar uppnås med detta: Förändringar som upptäcks under utvecklingen kan fortfarande inkluderas i produkten, samt alla steg kan testas mot användarens förväntningar och krav i varje steg och åter hjälpa till att iterativt påverka slutresultatet. Att den tänkta användargruppen ses som en utvecklingsresurs att samarbeta med utgör grunden till metoder för användarcentrerad design (eng. *User Centred Design*, UCD). Det betyder att man har användare eller representanter för användargruppen med under alla faser i designen och som aktiva granskare och projektdeltagare, för att påverka och förändra utvecklingsprocesser, testa produktens funktioner, begrepp och implementationsidéer.

Metoder för att skapa robotens grafiska gränssnitt är i sig inte nyuppfunna. Beprövade riktlinjer, design- och evalueringsmetoder (Smith och Mosier 1986; Apple 1992; Mayhew 1992;

Tognazzini 1992; Nielsen 1993) osv. har utvecklats eftersom ett grafiskt gränssnitt för operation av en mobil enhet inte skiljer sig mycket från andra skärmbaserade gränssnitt. De metoder som används här kommer att förklaras, men för en djupare insyn i metoden refereras till litteraturen.

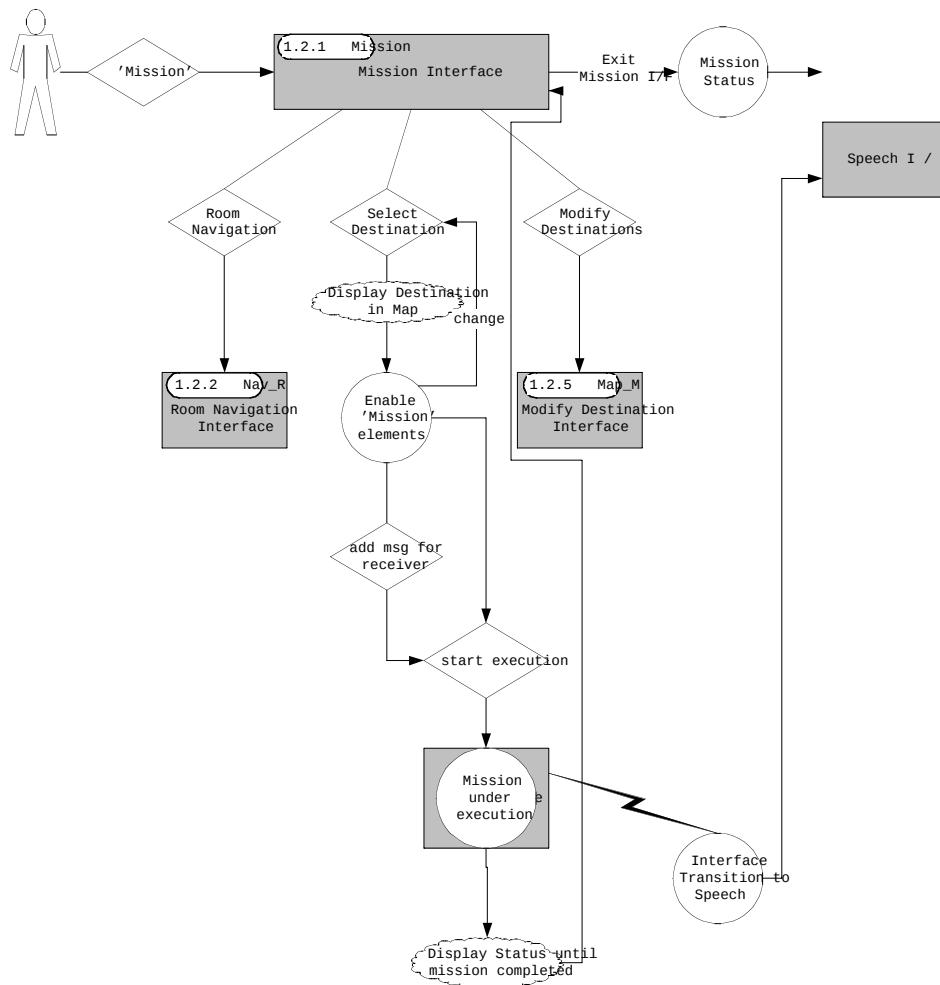
9.2.1 Kravspecifikation

Som utgångspunkt för kravspecifikationen fanns vår projektbeskrivning (Severinson Eklundh 1998): Vår servicerobot är tänkt att hämta och bära saker och ting åt användaren. Handlingar som krävs för att utföra tjänsten skall kunna initieras, inspekteras och avbrytas på ett intuitivt sätt. I projektet har vi som möjliga användare specificerat människor med behov av transportuppgifter som de själv har svårt att utföra. Användare är vana vid att använda datorer med fönsterbaserade operativsystem, det finns en sådan (stationär) maskin med storskärm (minst upplösning 800x600 bildpunkter) och minst 256 färger till förfogande. Maskinen är dessutom kopplad till ett nätverk och har troligen stöd för multimedia, dvs. en mikrofon för inspelning av ljud och ett (inbyggt) högtalarsystem för uppspelning av ljud. Användaren har inga hinder att köra datorn med de vanliga inmatningsredskapen som mus och tangentbord. Systemet skall vara så enkelt att inga tekniska kunskaper krävs för att använda det grafiska robotgränssnittet. Systemets användare definieras som *icke-expert* utan förkunskaper i robotkörning.

Med detta scenario i bakhuvudet framställdes en första kravspecifikation för det grafiska gränssnittet som beskriver olika krav först som text, sedan förklarar de olika funktionerna i sammanhängande flödes- och tillståndsdigram för att därefter diskutera olika lösningsalternativ mer detaljerat. Som exempel för det tänkta "mission-gränssnittet" anfördes Figur 23 nedan med huvudfunktioner i rektangulära lådor, övergångar eller förhållande i rombiska boxar emellan olika element, blixtar för att markera övergångar i interaktionsmodalitet (t.ex. från det grafiska gränssnittet till vårt talgränssnitt), elliptiska "bubblor" för statusåterkoppling till användaren, och cirklar för systemets interna variabeländringar.

Som exempel kan en användare enligt huvudmenyschemat nedan alltså välja en uppgift som roboten skall utföra genom att välja en mission ("Select Destination"). I kartan visas då platsen som har utsetts samtidigt som alla nödvändiga gränssnittelement aktiveras. Användaren kan vid den tidpunkten skicka roboten ("start execution"), vilket leder till att den interna statusen i gränssnitt ändras till "mission-under-execution" och användaren får klar och tydlig återkoppling av denna förändrade situation som bekräftelse på att missionen verkligen har kommit igång. Detta tillstånd bibehålls tills roboten har kommit fram, då det grafiska gränssnittet faller tillbaka till utgångsläge ("Mission Interface") igen.

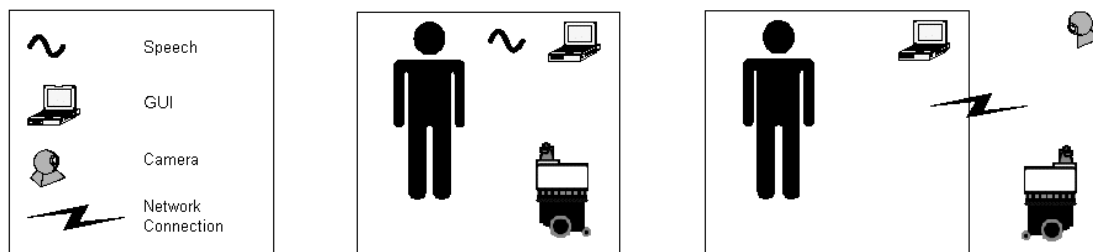
Notera blixten som går från objektet "Mission_under_execution" till "Interface_transition_to_speech". Denna illustration skall tydliggöra att roboten kan skickas på en mission och när den har kommit fram, kanske kan tilltalas där av en annan person. I klartext betyder det att det grafiska gränssnittet och huvudanvändaren kan ha bestämt att en viss aktion skall utföras, men att det bara lite senare eller fullt ut även parallellt är möjligt att tilltala roboten med tal, dvs. via en annan gränssnittsmodalitet än den grafiska. Detta förhållande skall belysas närmare i nästa avsnitt.



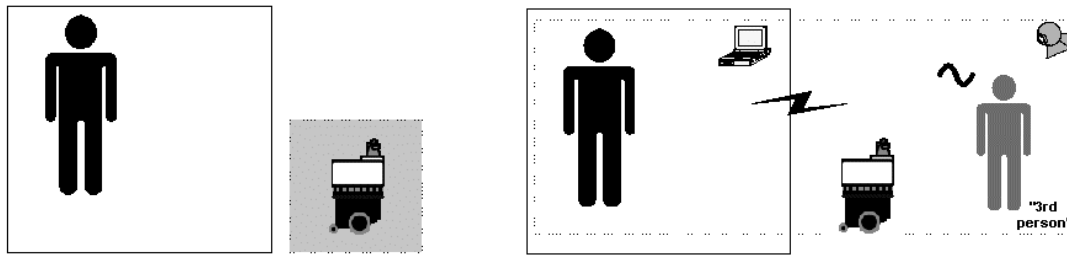
Figur 23. Huvudgränssnitt 1.2.1 "Mission"

9.2.2 Systemarkitektur för robotinteraktion

När man tänker skapa ett grafiskt gränssnitt för en mobil servicerobot som också skall få ett talgränssnitt, finns det minst fyra grundläggande kommunikations- och interaktionssituationer som måste täckas av interaktionsmodelleringen: Användare och dator i samma rum, teleoperation, autonom robot, och tele-presence med interaktion med 3:e person, se figur 24 och 25.



Figur 24. a) använda symboler, b) samma rum, c) teleoperation.



Figur 25. a) Autonom robot, b) tele-presence och interaktion med 3:e person

Förenklat kan man också säga att i interaktionen med en mobil robot, då både ett grafiskt och ett talgränssnitt skall kunna användas, måste en konstruktör i sitt tankeschema kunna svara på tre frågor:

1. när interaktionen skall ske, var är användaren respektive roboten i detta läge?,
2. när interaktionen skall ske, finns roboten då under antingen nätverkskopplad eller direktkontroll av användaren eller en annan person? och
3. vilka gränssnittsmodaliteter skall användas för denna interaktion?

Den första frågan siktar mot det spatiala förhållande som råder mellan användaren och den mobila roboten när interaktionen äger rum. Att det system som styrs från ett grafiskt gränssnitt kan röra sig i miljön behandlas dåligt eller inte alls av kända grafiska gränssnittsteorier. I normalfallet antas implicit att den interaktiva artefakt eller objektet som styrs med det grafiska gränssnittet står i ett fast spatialt förhållande till användaren av systemet. Med ett exempel kan det göras tydligare: När man använder datorn för att antingen beräkna någonting, skriva ett dokument eller övervaka en maskin, så finns användaren framför datorskärmen under utförandet, och systemet själv förflyttar sig i regel *inte* mellan olika platser. Men en sådan förflyttning är faktiskt en av robotens huvudfunktioner.

Den andra frågan ovan har som mål att kartlägga om roboten kan kontrolleras omedelbart och direkt eller över distans via ett tekniskt kommunikationssystem, t.ex. ett datornätverk. Är användaren och roboten i samma rum och ett talgränssnitt finns på roboten så skall det vara möjligt att kontrollera roboten genom att tilltala den. Ett grafiskt gränssnitt kan däremot antingen vara placerat på roboten (detta är inte implementerat hos oss, se ovan för kravspecifikationsbeskrivning) eller stå till förfogande på en extern skärm. I detta fall måste interaktionen som sker på skärmen överföras via ett nätverk till antingen roboten själv eller en annan dator som i sin tur tar upp kommunikation med roboten i en distribuerad systemarkitektur (se kapitel 6, sidan 22 ff., för detaljer). Viktigt för oss i detta sammanhang är att vi utgår från att roboten har kontakt med det grafiska gränssnittet via nätet, dvs. användaren kan ge anvisningar till roboten på distans. Detta kallas för "telerobotik"-kontroll som i motsats till "teleoperation" inte kräver att varje steg för sig måste utföras av användaren själv, men att användaren kontrollerar roboten på ett högre plan, dvs. genom att specificera missionens mål och sedan överlåter till roboten att ta reda på hur detta missionsmål skall verkställas i detalj.

En exempel: Att ge användaren en joystick för att köra en robot skulle betyda att användaren måste hålla i spaken hela tiden, att varje rörelse måste kontrolleras exakt och att varje förändring överförs till roboten som i sin tur rullar omkring enligt kontrollspakens läge. Återkopplingen till användaren av utförd action måste följaktligen ske i realtid. Denna teleoperation fordrar därför permanent kontroll, uppmärksamhet och snabba reaktioner från användaren.

Om användaren i stället via ett förprogrammerat sätt (jfr med s.k. 'makro' i programmeringsspråk) kan definiera målet för en aktion (och inte varje detaljsteg i utförandet för att nå detta mål) så kan man kalla detta för telerobotik, eftersom bara några kommandon måste överföras till roboten. Användaren kan i detta fall låta en sorts "utförandeautomatik" sköta alla detaljer.

Det är värt att observera här att en sådan telerobotik inte bara möjliggör en *delegering* av en uppgift, utan att en sådan delegering också kräver ett förtroende för att roboten kommer att utföra precis det som har delegerats på det sättet som förväntas av användaren. Detta förtroende måste därför byggas upp av roboten genom att de flesta missioner lyckas och eventuella fel förklaras tydligt. Roboten måste, med andra ord, vara eller upplevas som förutsägbar av användaren.

Ett specialfall är en autonom robot. Roboten har i detta tillstånd antingen ingen direkt- eller nätverkskontakt med användaren (som därmed har "tappat" alla kontrollmöjligheter) och/eller så behöver roboten ingen kontakt för utförandet av en specificerad aktion. Ett exempel som implementerats på vår robot är beteendet att undvika hinder. Detta sker fullt autonomt, dvs. när robotens sensorer upptäcker ett objekt eller person i sin omgivning så blir roboten först långsammare innan den stannar helt om inte hindret försvinner eller går att passera vid sidan om.

Detta låter ganska naturligt, men måste ses som ett uttryck för det som uppfattas som "smart" eller "intelligent" i en robot. Det är inte en operatör som sitter och initierar detta undvikande genom knapptryckningar i ett gränssnitt, och det är därför som det inte behövs kontrollelement för autonoma beteenden. Roboten, och med detta också gränssnittet, agerar i sin karaktär som en agent som med sin egen "vilja" eller helt autonomt utför en reaktion på insamlade och tolkade sensordata.

Uppgiften att skapa ett grafiskt gränssnitt till ett sådant system kan därför också breddas genom att säga att gränssnittskapande måste inkludera perceptuella beteenden hos systemet. Strategin som valts i vårt grafiska gränssnitt är att *gömma* denna utökade funktionalitet och medvetet göra uppträdandet *heltransparent* inför användaren. Att roboten undviker hinder är därför ingenting som en användare kan styra eller påverka, utan en upplevd "egenhet" hos roboten själv. Roboten tillåts att bete sig naturligt, dvs. som ett levande väsen.

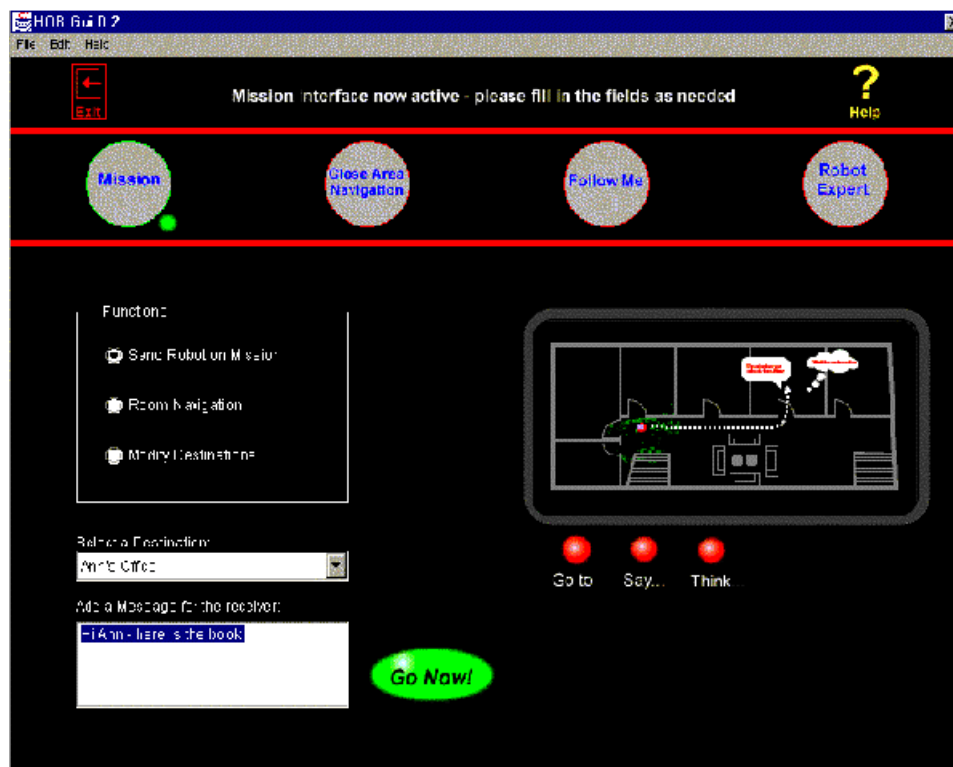
En hård granskning med kritisk eftertanke visar nackdelen med vårt förfarande. Eftersom vi inte tillåter användaren att styra robotens väjningsbeteende själv så tar vi effektivt bort användarens möjlighet att själv till hundra procent bestämma över roboten, dvs. vi inskränker användarens rätt att själv avgöra vad som skall få hända. Ett konstruerat exempel kan problematisera en möjlig konsekvens: En rullstolsbunden användares lägenhet brinner och den enda utgången är blockerad med ett hinder som skulle kunna forceras genom robotens förmåga att köra på det. Väjningsbeteendet förhindrar dock detta eftersom roboten inte tillåts att åka på saker.

Den sista av de tre ovan nämnda punkterna berör vilken modalitet som används, i vårt fall antingen det grafiska gränssnittet eller talgränssnittet. Vi utgår tills vidare från att användaren i början av robotkörningen föredrar det grafiska gränssnittet eftersom utförbara missioner och kontrollelement där är representerade på ett självförklarande sätt. Det är dock oklart om denna användningspreferens kommer att stå sig under längre användningsperioder. Det är fullt tänkbart att bestämda mönster av preferenser för antingen det grafiska eller det naturliga talgränssnittet (och/eller en kombination av båda) kommer att framträda i vissa situationer.

9.3. Detaljdesign och enhetstester

Olika prototyper har tagits fram för att användartesta koncept och lösningar med det grafiska gränssnittet. Inom projektgruppen hade vi tidigt kommit överens att använda oss av Java som programmeringsspråk för att producera gränssnittsprototyper. Java är en plattformoberoende, tolkad, kraftfull, lätt och objektorienterad utvecklingsmiljö, som också kan användas som verktyg för *rapid prototyping*, t.ex. med integrerade mjukvarupaket som Symantecs Visual Café eller IBMs JavaBuilder. Fördelen är att prototyper lätt kan vidareutvecklas till fullskaliga gränssnitt, dvs. prototyperna kan direkt tas in i verkliga användartester (det finns ingen skillnad mellan prototyper och en riktig version för en användare), mjukvaruproduktion och sedan systemintegration.

Den första prototypen (se Figur 26) togs fram med robotens ursprungliga utseende som metafor och med alltför stora förväntningar på robotens möjliga funktionalitet. Gränssnittet presenteras som en vanlig tillämpning på skärmen med t.ex. en menyrad ("File, Edit, Help"), dock anpassad till robotens svarta gestalt, dess röda linjer och runda sensorer. En upp- och ner, vänster till höger prioritering av element har implementerats, dvs. huvudfunktioner ligger ovanpå detaljstyrelement och kan väljas från alla aktiva "sidor".



Figur 26. Första prototypen.

Längst upp finns en "Exit"- och en "Help"- knapp, däremellan ett statustextfält som visar kontextkänslig information om vad som kan göras i gränssnittet. Kontextkänslig hantering och stöd för explorativt lärande av gränssnittet är genomgående implementerat, t.ex. i de runda objekten mellan de två röda linjerna så är kanten också rödmarkerad tills musen förs över ett element. Med musen ovanpå elementet växlas färgen till ett grönt sken och statustexten visar vad som kan göras. Väljs sedan ett element genom ett musklick så visas dessutom en liten grön punkt för att markera tydligt att objektet har aktiverats och underlättar på så sätt användarens navigation i applikationen.

De fyra runda elementen representerar från vänster till höger ett gränssnitt för utförande av uppdrag, gränssnitt för närmiljönavigation, följ-a-mig-beteende, samt en rubrik "Robot Expert" (eller "Settings") som reserverades för andra möjliga inställningar.

Funktionsområdet under den andra röda linjen representerar olika sätt att köra ett uppdrag. Till vänster väljs först en övergripande funktion ("Send robot on mission", "Room Navigation", eller "Modify Destination"), för att sedan längre ner specificerar detaljer i den valda funktionen, t.ex. kan man välja olika platser som roboten skall gå till från en nerfällbar lista och vilket medelande roboten då skall lämna. Är en mission tillräckligt specificerad kan man trycka på "Go Now!" knappen för att skicka roboten.

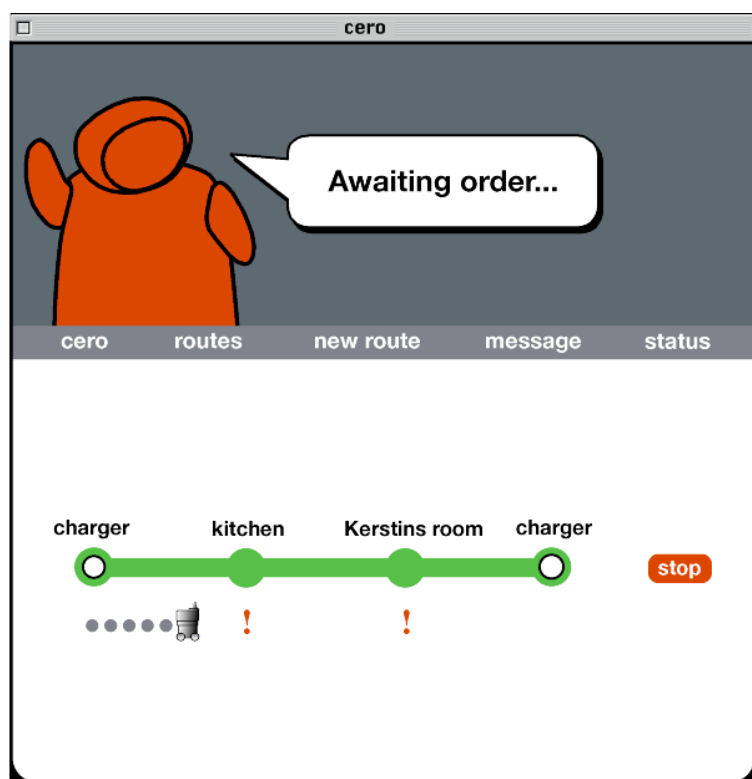
Ett sätt att följa robotens framgång i uppdraget och rörelse i miljön visas till höger i kartan. Det går också att skicka roboten till olika platser genom att klicka direkt i kartan och robotpunkten till höger. Knappsatsen under kartan kan användas för att specificera vad som ska hända på den utpekade punkten.

Gränssnittsprototypen utvecklades tills det gick att köra användartester med den. En kooperativ observationsmetod användes för att utsätta detta gränssnitt för en preliminär utvärdering. En försöksperson körde prototypen explorativt (utan att roboten verkligen var inkopplad) på skärmen för att utföra några tänkta huvudfunktioner. På video spelades in vad som hände på skärmen samt det som försökspersonen och utvecklaren pratade högt om (med en s.k. "think-aloud" metod, jfr, t.ex. Nielsen och Mack 1994).

Metoden är relativt lätt att lära sig och genomföra. Försökspersonen uppmuntras att kritisera systemet och utvecklaren kan hjälpa till med funktioner som användaren har problem med, dvs. områden i gränssnittet som identifieras som i behov av förbättringar.

Resultatet visade stort sett på en stor miss: Prototypen blev underkänd vad beträffar att visa systemets status tillräckligt tydligt. Användaren kände sig inte tillräckligt informerad om vad som pågick i systemet och det saknades lämplig återkoppling antingen via simulerade aktioner eller via den riktiga roboten. Här verkar det som beprövad metodik inom GUI-utvecklingen leder fel. Robotik är ett område där användare vill se en reaktion på en simulerad eller riktig robot, när en aktion har utförts i gränssnittet. Som utvecklare av det grafiska gränssnittet hade vi som val att därefter försöka oss på en simulerad miljö eller skjuta upp robotgränssnittsutvecklingen tills roboten verkligen kunde utföra några funktioner. Vi valde dock att försöka förena både ansatserna.

I en vidareutvecklad prototypversion var det tänkt att simulera robotens funktionalitet i en "Wizard av Oz"-studie (Dahlbäck m fl. 1993, se även kapitel 4) Tyvärr visade det sig att det tänkta Wizard-av-Oz försöket inte gick att genomföra med den grafiska prototypen på grund av för hög komplexitet för ett gemensamt simulerat tal- och grafiskt gränssnitt.



Figur 27. CERO-inspirerad prototyp

I samband med "Wizard av Oz"-studien och framtagandet av robotens fysiska design med en transportenhet visade det sig att robotens talgränssnitt behövde ett mycket tydligt återkopplingssystem och roboten själv en tydligare framsida. CERO-idén (se avsnitt om tal och CERO, sidan 51) föddes snart efter försöket. Det kändes också som det grafiska gränssnittet behövde anpassas både i färger, karaktär och t.o.m. en CERO-figur för att bevara ett helhetsintryck av roboten och kontrollgränssnitt (se Figur 8 på sidan 18).

Det CERO-anpassade gränssnittet fick en radikalt annorlunda färgsättning och uppbyggnad. Applikationen i form och utseende motsvarar inte längre en standardapplikation och sätter det färgglada och mer lekfulla uttryckssättet i fokus. CERO figuren används som huvudelement i gränssnittet för att förklara gränssnittets funktionalitet med en agentmetafor.

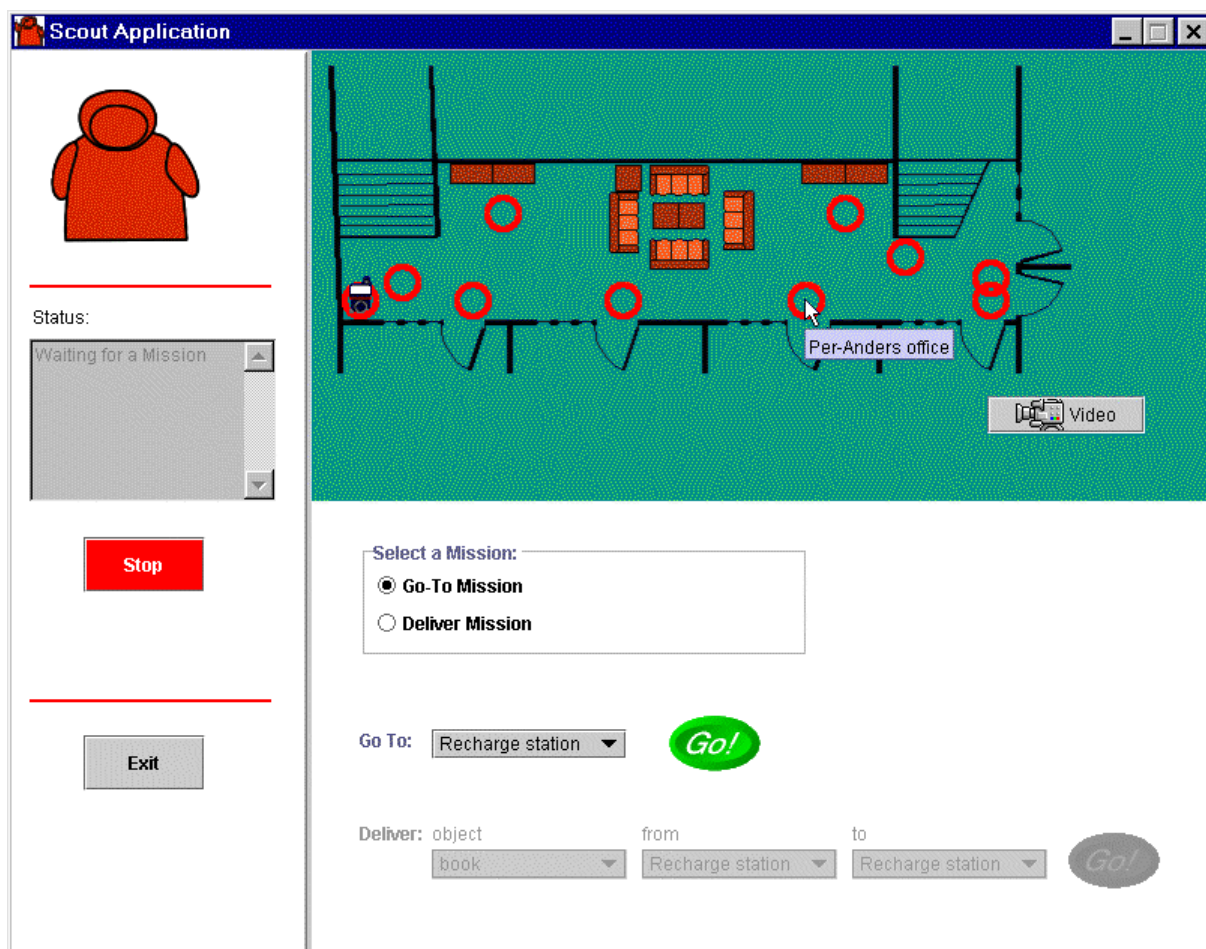
Ett annat viktigt bidrag som designen av denna prototyp försökte sig på var att lösa sig från den spatiala representationen av den fysiska omgivningen som roboten skulle rulla i. Kartrepresentationer är svåra att generera automatiskt; speciellt om roboten skulle åka mellan olika byggnadsdelar skulle det inte bli lätt att byta t.ex. mellan olika kartor eller hantera olika storleksförhållanden t.ex. via en zoomfunktion. En möjlig lösning är skisserad i Figur 27. Designen är baserad på en tunnelbaneliknande representation av olika färdvägar som roboten skulle kunna åka, med olika sträckor med slutstationer och mellanpunkter på vägen. Utropstecken talar om att det finns mer information på punkterna. Robotens färdväg under ett uppdrag kan visas genom en robotsymbol som förflyttar sig under en speciell linje.

Denna prototyp har framställts först på ritningsnivå och sedan implementerats som Java-prototyp som man kunde testa på skärmen. Eftersom den tänkta navigeringen inte kom igång enligt planering (se Tåqvist 1999) vidareutvecklades dock aldrig prototypen fullt ut eftersom det var oklart vilka funktioner som verkligen gick att realisera och komma åt i robotens system.

I ett försök att till sist integrera alla robotkomponenter programmerades en ny prototyp med en annan ansats. Bara de funktioner som kunde förverkligas på roboten implementerades också i grafiska gränssnittet. Idéer och lärdomar från tidigare prototyper fick dock sina markanta nerlag även i detta integrerade användargränssnitt. Detta gränssnitt är också det som körs i dag för att kontrollera roboten med en utformning som beskrivs i detalj i nästa avsnitt.

9.4. Integrerat gränssnitt

Diskussionen omkring det nuvarande grafiska gränssnittet är lättast att förstå genom att visa hur det ser ut och förklara detaljer. Självklart går det dock bättre att uppleva alla dynamiska förändringar i gränssnittet genom att köra roboten på riktigt än att bara läsa beskrivningarna här.



Figur 28. Senaste versionen av robotens gränssnitt

Gränssnittets layout är indelat i tre områden. Till vänster finns CERO-figuren som animeras under körningen, ett fält med statusmeddelande i klartext samt en "STOP"- och en "EXIT"-knapp. I övre delen av huvudområdet finns en karta över den miljö som roboten kan köra i. I nedre delen finns slutligen kontrollpanelen som styr robotens aktioner.

Uppbyggnaden är inspirerad av webbsidors navigeringsdesign, dvs. orientering sker till vänster, sedan finns informationen till höger i en större fönsterdel. Till skillnad från webbsidors design av navigering och återkoppling av besökt sida används dock nästan aldrig element på vänstra sidan aktivt. CERO och statusfältet visar vad som pågår, "STOP"-knappen är bara

tänkt att användas i en sorts nödsituation (symboliserat via den röda färgsättningen) och "EXIT"-knappen avslutar användandet av denna applikation.

På kartan kan användaren först orientera sig och ta reda på var roboten befinner sig just nu (i bilden ovan är roboten vid en punkt som kallats för *batteriladdningsstation*). Det går också att föra muspekaren över en punkt i kartan för att informera sig om de olika punkter roboten kan gå till. Hela kartan är skisserad som en arkitektonisk översiktsplan, dörrar visar ingångar till olika kontor och vår soffgrupp är också markerad.

Roboten kan skickas på två olika uppdrag, ett "Go-To"- och ett "Deliver"-uppdrag. För att skilja dem åt införde vi att uppgiften måste väljas innan detaljerna kan specificeras. Detta sker via en s.k. "radioknapp" där man väljer det elementet som skall aktiveras. Har man valt "Go-To Mission" så är bara de knappar som tillhör denna funktion aktiverade lite längre ner. Komponenter som tillhör "Deliver Mission" är vid den tidpunkten ej tillgängliga och visas i grått. Likadant fungerar det för "Deliver Mission" när det har valts i övre lådan, bara tvärtom.

"Go-To"-uppdraget specificeras genom att välja en plats som roboten skall åka till från en nerfällbar lista. När användaren bläddrar igenom elementen i listan markeras dessa dynamiskt i kartan för att ge återkoppling. Har en plats bestämts kan robotens körning startas genom att trycka på "Go!"-knappen.

"Deliver"-funktionen fungerar likadant som "Go-To"-funktionen i interaktionen. Användaren måste dock specificera lite mer detaljer för att aktionen skall kunna utföras. Ett objekt som skall transporteras skall specificeras, en utgångspunkt där objektet skall hämtas och en målposition där objektet skall lämnas eller transporteras till måste vara satt.

När användaren har tryckt på "Go!" knappen så sätter roboten igång vilket visas i gränssnittet genom att robotbilden börjar röra på sig i kartan samt animeringen av CERO-figuren (uppe i vänstra hörnet av gränssnittet) som markerar tydligt att roboten är på väg. Statusfältet talar också om för användaren vad som utförs just nu av roboten. Förhoppningen med designen är att användaren kan ha uppsikt över roboten samt få återkoppling över aktioner som har startats i gränssnittet.

För att hjälpa en användare att komma igång med robotens gränssnitt har en PowerPoint-presentation skapats som förklarar de olika delarna samt visar hur de olika element fungerar. Denna presentation har testats och finjusterats flera gånger. Tanken är att en sådan multimedial snabb presentation skall vara allt som behövs för att köra roboten, dvs. inom bara några minuter kan man bli en robotoperatör.

Det sistnämnda speglar inriktningen i designen som genomsyrar hela grafiska gränssnittet: att med mycket enkla och tydliga symboler, klara processteg och mycket återkoppling på olika nivåer skapa ett gränssnitt som ska vara lätt att lära sig och enkelt att använda. Gränssnittet har också reducerats till de element som absolut krävs för att köra roboten. Inom utvecklingen av grafiska gränssnitt kallas en sådan designstrategi för reduktionistisk och förenklande. Detta motiveras av att roboten själv inte kan utföra avancerade funktioner som kräver mycket komplexa och komplicerade aktioner av användaren. Designkrav och funktion bildar en enhet, dvs. "*Form equals function*".

Det nuvarande gränssnittet har tagits fram iterativt och flera användare har fått pröva det i olika omgångar. Användarna kunde på så sätt bidra med kritik, förbättringsförslag och idéer. Efter några omgångar listade vi viktiga punkter och problem som observerats, värdesatte hur

allvarliga de påpekade punkterna var och gjorde sedan om gränssnittsdetaljer som kom högst upp i förändringsprioritet. Som exempel kan ett utdrag av vår förändringslista ges:

Nr.	Observation	Orsak	Värderat	Resultat
1	Användaren vill ha återkoppling i drop-down boxes	man ser inte vilka platser som är valda i kartan -- svårt om man inte känner till platser i förväg	Viktig att ha	Ändrat
2	Oklart när ett uppdrag verkligen sätts igång eller när en knapp har tryckts i prototypen	Tekniskt: tidpunkten för "musklick" och "mus upp" i programmeringen utgörs av två olika meddelanden	Borde ändras	Ändrat
3	Markera robotens start och målpunkt i kartan, rita en linje emellan de två punkterna	Önskemål från en användare	Kan läggas till senare	Bara delvis implementerat - - för mycket grafisk information gör GUI:n för komplex och förvirrande

Efter prototyptestande i detta utvecklingsskede kom det fram att främst återkoppling till användaren borde förstärkas ytterligare. Det gäller speciellt hanteringen av de grafiska elementen och återkopplingen på uppdragsnivå (uppdrag startat, uppdrag på gång, uppdrag utfört). Andra förslag hade huvudsakligen till syfte att *utöka* inte bara det grafiska gränssnittet, men också funktionaliteten hos roboten eller interaktions- och kommunikationsmodaliteter. En användare nämnde t.ex. att det borde gå att övervaka ett område över distans med roboten, dvs. roboten kan ge en (video-) bild av det som händer i en korridor längre bort, vem som kommer in m.m. En sådan övervakningsfunktion går naturligt att programmera in i vår plattform, men eftersom detta varken skulle gynna en okomplicerad gränssnittsdesign eller (som vi antar) bidra till en allmänt ökad acceptans av roboten har vi avstått från en sådan funktion.

Den framtida utvecklingen av grafiska gränssnitt inom servicerobotik har flera uppgifter kvar att undersöka. Vår robot är avsedd för att hämta och bära föremål, men många andra uppgifter går att föreställa sig, speciellt manipulerande arbetsuppgifter. De har redan efterfrågats av användare, men skulle kräva en robotarm. Förhållandet mellan direktstyrda aktioner av en operatör och intelligenta autonoma funktioner som utförs av robotens styrsystem komma att spela en viktig roll i manipulation. För utvecklare av grafiska gränssnitt kommer det också att vara betydelsefullt att ha bra kunskap om hur mycket återkopplingsinformation som måste ges till användaren. Det har visat sig i användning av robotar för övervakning och intelligenta system att antalet och utformningen av robotsystemets övervakningselement har direkt påverkan hur effektiva gränssnitt kan vara (Johannesen och Woods 1991; Murray 1995).

Ett annat område med utmaningar är skapande av kartsystem för mobila robotar. Om ett spatialt förhållande måste representeras är det idag "svårt" att producera en karta från en existerande miljö. Ordet "svårt" har satts inom citattecken inte för det är matematiskt komplicerat, men för att det borde kunna göras utan att någon expert är tvungen att mata in datasiffror i någon numeriskt gränssnitt eller programmeringsspråk. Arkitekturtiktor finns för byggnader, för nyare byggnationer även i digitalt format som CAD-ritningar, men överföringen från två- resp. tredimensionella filformat till något format som de flesta robotar generellt kan använda sig av är inte standardiserad hittills.

En annan ansats för skapande av kartor är att förlita sig på ett sofistikerat robotsensorsystem med t.ex. en tredimensionell laserscanner. En laserscanner skickar ut laserstrålar enligt ett

förbestämt mönster och träffar rummets väggar och möbler. Där "studsar" strålarna tillbaka och mäts igen av laserscannern. Eftersom det går att räkna ut avstånd till föremål i rummet, kan kartor över miljöer ritas upp. Kombinerar man en sådan teknologi med en människa-robotdialog för inläring av olika platser samt en grafisk representationsform som är okänslig mot spatiala detaljfel (se tunnelbanan som kartmetafor ovan) så skulle det vara fullt möjligt att låta slutanvändaren anpassa robotens gränssnitt till de fysiska förhållanden som råder i hemmet eller på kontoret. En sådant moment av automatiserat byggande av kartor skulle främja en anpassning av robotens navigeringsfunktion. Detta är viktigt att kunna utföra för att möjliggöra en slutanvändarstyrd robotkartanpassning, eftersom det är orealistiskt att förvänta sig "standardiserade" omgivningar som en hjälpmedelrobot skulle fungera i och som kan programmeras in i förväg ifrån tillverkaren.

Det bedöms också vara av stort intresse att öppna robotens grafiska gränssnitt mot utveckling av småskärmsinteraktion. Många tekniska apparater för styrning av hushållsmaskiner (t.ex. symaskiner och diskmaskiner) eller kommunikation (mobiltelefoner samt s.k. personliga digitala assistenter — PDA) kommer med eller kan kontrolleras via småskärmar antingen direkt eller via speciella nätverk t.ex. avsett för hemautomatisering. Stora datorskärmar som i regel används för stationära eller t.o.m. för bärbara datorer idag kan köras med en mycket högre upplösning än dagens småskärmsapparater. Eftersom det inte går att visa samma antal funktionselement på de små skärmarna måste antalet grafiska kontrollelement reduceras. Flera olika strategier är teoretiskt tänkbara. Antingen kan funktionerna i sig reduceras och begränsas till dem som absolut krävs eller också måste funktioner integreras mycket tätare än tidigare och deras presentation till användaren kanske göras mer kontextberoende. Som ett första konkret steg förslås till exempel att det grafiska gränssnittet för vår robot förflyttas till någon digital assistent typ PalmPilot™, en mobiltelefon eller liknande för att visa på hur flexibel gränssnittsmodelleringen kan göras.

I samband med småskärmsinteraktion finns också ett annat intressant "spår" att följa i vidareutvecklingen av grafiska gränssnitt: Placeringen, storleken, utformning m.m. av olika skärmar som kan användas för att kontrollera robotar. Med en småskärmsinteraktion skulle det vara möjligt att placera robotens skärmbaserade kontrollenhet antingen på roboten själv och/eller förflytta den i någon slags handhållen apparat (se ovan). Beroende på vilken skärm eller vilka skärmar som används för interaktion måste flexibiliteten i det grafiska gränssnittet vara tillräckligt utvecklad för att kunna visa det som krävs för att styra roboten i den aktuella situationen. Det är sannolikt att interaktionen med roboten förändras från det som vi har observerat med det grafiska gränssnitt byggt för att köras på en vanlig datorskärm, t.ex. genom att dela upp en kontroll- eller styrprocess som kräver flera inställningar av användaren delas upp i flera sekventiella steg eller menyer att välja ifrån.

Ett annat forsknings- och utvecklingsområde som är värt att engagera sig i är att försöka använda det grafiska gränssnittet bara till visuell återkoppling av ett talgränssnitt. Idén här är att det grafiska gränssnitt bara används för att stödja användarens talade interaktion med roboten. I ett sådant system skall användaren egentligen bara prata till systemet, men det grafiska gränssnittet ger återkoppling på det som kan sägas, det som redan har känts igen m.m. Funktioner som "gissningar" av ord, fraser, eller satser skulle göra det möjligt för användare med funktions- och språkhinder att använda sig av roboten utan krävande motorisk detaljstyrning för att nå knappar eller andra grafiska komponenter på en skärm.

10. Utformning av talgränssnitt

I arbetet med att ta fram ett talgränssnitt till roboten har vi utgått från psykologiska och lingvistiska principer som vi återfinner i samtal mellan människor. Att utforma ett talgränssnitt innebär att på syntetisk väg skapa en social karaktär som kan agera gentemot användaren. Genom att använda vår kunskap om principerna för mänsklig dialog kan vi skapa ett tillförlitligt men samtidigt flexibelt system.

Ett dialogsystems generella funktion är att verka som ett gränssnitt mellan en människa som använder naturligt språk och en dator som använder ett datorspråk. För att systemet, i vårt fall roboten ska kunna agera krävs att de instruktioner som den får är fria från tvetydighet och tillräckligt specifika. Människors sätt att behandla språket är normalt det omvända, vi både använder och tolererar vaghet, flertydighet och ofullständiga uttryck. Det är detta komplicerade förhållande som utgör kärnproblemet när det gäller gränssnitt med naturligt språk. Dialogsystemets roll är att genom sitt beteende verka både som en naturlig samtalspartner och en precis komponent i robotens styr- och kontrollsystem.

I det följande ska vi beskriva systemet från olika perspektiv. Utgångspunkten är de dialogscenarier som vi identifierat under utvecklingsarbetet. Därefter behandlar vi de lingvistiska och psykologiska principer som ligger till grund för systemets generella beteende. Till sist ges en systembeskrivning av det implementerade systemet samt en kort diskussion av de erfarenheter som gjorts under arbetet med dialogsystemet.

10.1 Dialogmodeller

Man kan tala om dialogmodeller på åtminstone två nivåer när det gäller dialogsystem. Dels en nivå som handlar om de principer som används för att hantera systemets generella beteende gentemot användaren. Dessa principer brukar hämtas från lingvistiska och psykologiska teorier och uttrycks oftast som abstrakta mål eller villkor ställda på dialogen. En annan nivå handlar mer om systemets konkreta utformning. Detta omfattar alla aspekter av utformning både vad gäller de uppgifter som systemet ska lösa, vad det ska säga i olika situationer, men också den fysiska utformningen av systemet. I ett idealiskt system är den nivå som omfattar principerna statisk medan den konkreta nivån kan förändras vid utformningen av systemet för att ge det en viss karaktär.

10.2 Dialogscenarier – användningsfall

För att skapa oss en bild av hur det slutliga systemet ska utformas har vi arbetat med olika typer av scenarier som medger att roboten används på ett sätt som passar användaren. För utformningen av dialogsystemet innebär detta att vi konstruerar fiktiva dialoger mellan användare och robot varefter vi väljer ut dem vi anser mest gångbara. Exemplet nedan (exempel 1) visar en sådan dialog. Exemplet återges här på svenska. I det system vi beskriver i rapporten är den här dialogen implementerad för engelska. Resterande exempel, som är tagna direkt ur systemet, kommer dock att återges på engelska.

Exempel 1

- (1.1) A: *Robot*
(1.2) R: *Hur kan jag hjälpa till?*
(1.3) A: *Hämta kaffe*
(1.4) R: *Var är kaffet?*
(1.5) A: *På Karins rum*
(1.6) R: *Hämta kaffe i Karins rum?*
(1.7) A: *Ja*
(1.8) R: *Hämtar kaffe i Karins rum!*

I det kompletta systemet har vi två olika huvudscenarier för röststyrning. Användaren kan ge sina kommandon på två sätt:

- Via en riktmikrofon placerad på roboten.
- Med ett headset, eller handhållen mikrofon som bärs av användaren.

I båda exemplen svarar roboten via talsyntes genom sin fastmonterade högtalare.

I figur 1 visas den ena av dessa situationer. För att inmatning ska kunna ske med riktmikrofon krävs att användaren befinner sig nära roboten och kan avge sina kommandon i en relativt tyst miljö. Ett headset kräver inte att roboten och användaren befinner sig på samma plats. För att kommunikationen ska fungera krävs att användaren kan höra robotens högtalare, något som medför att avståndet vid interaktion i praktiken inte kan vara så stort.

10.3 Lingvistiska och psykologiska modeller för dialog

I ett samtal mellan två människor är kontrahenterna involverade i en verksamhet som innebär att de genom sitt beteende bidrar till att uppnå ett eller flera gemensamma mål. En av utgångspunkterna vi kan använda för att beskriva hur denna process kan återföras till vårt dialogsystem bygger på arbetet av Herbert Clark (Clark 1996). Här är tanken om *gemensamma projekt* (Joint Projects) central. När två eller flera deltagare enats och genomfört en dialog som ett gemensamt projekt innebär det oftast att många *minimala projekt* (Minimal Projects) har avhandlats under vägen. I exempel 2 (nedan) är det minimala paret fråga/svar där talare B svarar talare A. I exempel 3 utförs handlingen ”att ge tillstånd” genom att B svarar ”ja” på A:s begäran.

En viktig aspekt av att kunna utföra handlingar genom tal är att det oftast inte räcker med att ordens yttras för att talhandlingen ska utföras. För att talhandlingar som dessa ska kunna ha någon effekt krävs det att vissa villkor är uppfyllda. I exempel 3 krävs att talare B har makt att bestämma över A (till exempel att B är förälder till A).

I exempel fyra har två minimala projekt nästads så att B:s svar på A:s begäran om att få gå ut beror av den aktuella tiden.

Exempel 2: Fråga–Svar (Question–Answer)

- (2.1) A: *Vad är klockan?* FRÅGA
(2.2) B: *Klockan är tio.* SVAR

Exempel 3: Begäran–Tillstånd (Request–Permit)

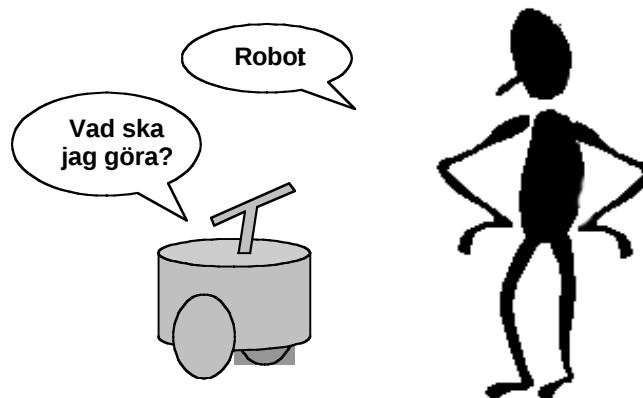
- (3.1) A: *Får jag saft?* BEGÄRAN
(3.2) B: *Ja.* TILLSTÅND

Exempel 4: Nästning

- (4.1) A: *Får jag gå ut?* BEGÄRAN
(4.2) B: *Vad är klockan?* FRÅGA
(4.3) A: *Klockan är 10.* SVAR
(4.4) B: *Ja.* TILLSTÅND

När det gäller vår systemutformning har vi förutom tanken om minimala projekt också inbegripit två andra viktiga aspekter nämligen *återkoppling* och *grundning* av dialogen (grounding).

Återkoppling på olika nivåer är mycket viktig när det gäller att bygga upp den gemensamma plattform av information som deltagarna behöver för att kunna agera inom ramen för det gemensamma projektet. Återkoppling kan ges på olika nivåer. En talare kan ge återkoppling i olika dimensioner, exempelvis att hon är *tillgänglig* för dialog (kontakt), att hon kan *uppfatta* det som sägs (perception), att hon *förstår* vad som sägs och sist men inte minst genom att *reagera* på den avsedda intentionen i det som sägs. För en närmare diskussion om återkoppling i mänsklig dialog se (Allwood m fl. 1991).



Figur 29. Den tänkta situationen när talgränssnittet används av en rörlig användare "ute på golvet".

I vårt system avser användaren att delegera en uppgift till roboten. Eftersom systemet är tänkt att vara flexibelt bör det successivt under det att dialogen förs kunna kräva mer information när det identifierat uppgifter som bara specificerats partiellt av användaren. Detta kräver att användaren får veta hur mycket information som roboten har om uppgiften som konversationen rör. I processen att skapa en gemensam plattform att agera utifrån är alltså återkoppling mycket viktigt. Nedanstående exempel (5) visar tydligt hur återkoppling på olika nivåer samverkar till att bygga upp den information som systemet och användaren behöver för att kunna agera och reagera på den andres yttranden och handlingar.

I exempel 5 initierar användaren dialogen genom att säga "Robot". Eftersom robotens mikrofon finns tillgänglig och roboten svarar överhuvudtaget kan användaren anta att hon fått kontakt med roboten och att robotens ljudupptagning fungerar. Genom att roboten svarar med ett för användaren relevant yttrande kan man anta att systemet förstått vad användaren sagt. Eftersom robotens yttrande är inriktat på den övergripande uppgiften som roboten har, nämligen att utföra uppdrag åt användaren, kan denne förhoppningsvis sluta sig till att roboten är beredd att ta emot ett uppdrag.

Exempel 5

		Talhandling
(5.1)	A: Robot	ANROP
(5.2)	R: Hur kan jag hjälpa till?	SVAR
(5.3)	A: Hämta kaffe	BEGÄRAN-UPPGIFT
(5.4)	R: Var är kaffet?	FRÅGA-PLATS
(5.5)	A: På Karins rum	SVAR
(5.6)	R: Hämta kaffe i Karins rum?	KLARTECKEN-UPPGIFT
(5.7)	A: Ja	GE-TILLSTÅND
(5.8)	R: Hämtar kaffe i Karins rum!	INFORMERA
<roboten rullar iväg>		

I nästa yttrande specificerar så användaren ett uppdrag, att hämta kaffe, genom yttrande (5.3). Eftersom informationen i användarens yttrande inte räcker för att utföra uppdraget (en plats saknas) kan systemet inte genast reagera positivt på användarens begäran utan begär tilläggsinformation genom att fråga "var är kaffet?" i yttrande (5.4). Genom yttrande (5.4) genomför systemet också en talhandling i syfte att grunda dialogen genom att repetera ordet "kaffe". Genom att användaren specificerar platsen där kaffet ska hämtas i (5.5) accepterar hon också att det är kaffe och inte något annat objekt som ska hämtas. Om systemet istället för (5.4) hade sagt "ange plats?" vore det i det läget inte klart för användaren vad roboten avsåg att hämta.

När användaren avgivit tillräckligt med information för att systemet ska kunna försöka utföra handlingen måste roboten inhämta klartecken från användaren. Detta eftersom handlingen innebär att navigationssystemet ska användas för att förflytta roboten fysiskt. Detta görs genom att parafrasera hela uppdraget i (5.6). När användaren säger "ja" i yttrande efter (5.7) accepterar hon hela uppdraget. Det sista som görs i dialogen är att roboten informerar användaren om vilket uppdrag den avser att utföra. Information i den här bemärkelsen är inte ett gemensamt projekt, eftersom roboten inte har skyldighet att grunda dialogen. Handlingen informera kräver därför inte något svar från användaren. En mindre informativ, men på många sätt likvärdig handling från robotens sida vore att endast rulla iväg, utan att yttra (5.8).

10.4 Dialogutformning

Grammatiken i prototypsystemet täcker ett språk innehållande ett fåtal satstyper, men medger ändå en ganska avancerad dialog. Grundtanken med utformningen av systemet är att språket både ska medge en flexibel dialog och dessutom vara lätt att lära sig för en användare. Vi har redan sett prov på den önskade flexibiliteten i exempel 5 (ovan).

Genom att utgå från de uppgifter som användaren vill kunna utföra med systemet och de tekniska möjligheter som systemet har idag har vi fått fram några uppgiftstyper som vi anser att systemet kan utföra (jfr beskrivningen av robotens arbetsuppgifter i kapitel 3.2). Med hjälp av dialogsystemet kan man utföra tre typer av uppgifter:

Uppgift	Beskrivning
Goto <location>	<u>Åka</u> till en <plats>
Deliver <object> <location>	<u>Leverera</u> ett <föremål> på en <plats>
Fetch <object> <location>	<u>Hämta</u> ett <föremål> på en <plats> (återvänd till utgångspunkten)

För att dialogsystemet ska känna igen vilken uppgift användaren vill ska utföras krävs att ett verb finns med bland det som användaren säger. Till exempel kan "deliver coffee" (leverera kaffe), tas som ett tecken på att uppgiften är deliver. Detta får till följd att systemet försöker få användaren att specificera ett föremål (<object>) och en plats (<location>), företrädesvis görs detta genom att ställa frågor till användaren till exempel: "what is the object?" (vad är föremålet?). När systemet har tillräcklig information för att kunna leverera föremålet, alltså både vilket föremål det rör sig om och var det ska leveras, får användaren en fråga där hon uppmanas ta ställning till om uppdraget ska utföras.

Systemet har ett fåtal generella strategier för att lösa sina dialoguppgifter.

- Ta över initiativet
- Grundning av dialogen genom återkoppling
- Felhantering genom minsta möjliga tillbakaspårning (*backtracking*)

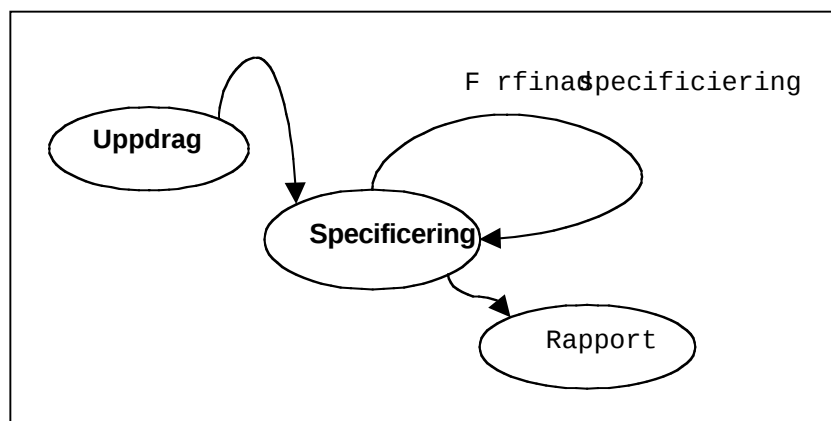
Användaren förväntas avge sina instruktioner i huvudsak enligt det schema som visas i figur 30. Det första stadiet innebär att systemet försöker känna igen ett av de uppdrag som systemet kan utföra. Därefter tar systemet initiativet och försöker få användaren att specificera den information som krävs för att uppdraget ska kunna utföras. När systemet har tillräcklig information för att utföra uppdraget, rapporterar systemet vad det avser att göra. Därefter sänds styrdata till robotens planeringssystem som försöker genomföra uppgiften i den fysiska världen.

Efter det att systemet känt igen en uppgift försöker det behålla initiativet genom att grunda dialogen med hjälp av återkoppling på användarens senaste yttrande och den information systemet har om dialogens fortskridande.

Systemet använder sig av tre typer av återkoppling:

- Återkoppling av graden av perception: Via CERO-dockan visas gester som signalerar uppmärksamhet från systemet (se senare avsnitt).
- Explicit återkoppling av fraser: Detta görs exempelvis genom att upprepa objektet i samband med begäran om mer information (se exempel 5 ovan).
- Parafrasering av order: När en uppgift specificerats tillräckligt uttalar systemet denna som en fråga som ska besvaras med ja eller nej (en så kallad ja/nej-fråga).

Felhanteringen i systemet är också avsedd att bidra till att systemet blir flexibelt och intuitivt att använda. I exempel 6 gör systemets taligenkänning ett fel i inledningen av dialogen. Felet repeteras som återkoppling i 6.2 (understrykningen). När användaren upptäcker detta yttrar denne "no" varefter systemet backar tillbaka lite och startar om dialogen genom att fråga den fråga som skulle ha använts om bara "Get" känts igen, alltså ett läge där varken plats eller objekt specificerats av användaren.



Figur 30. När ett uppdrag identifierats av systemet försöker det få användaren att lägga till fler uppgifter tills informationen är tillräcklig för att uppdraget ska kunna utföras. Därefter rapporterar systemet om hur uppdraget lyckas.

Strategin för felhanteringen i systemet som handlar om oklarheter eller fel i samband med att informationen om uppgiften byggs upp är genomgående att genom tillbakaspårning inleda en dialog med ett mer abstrakt innehåll. I vissa fall dock är det oklart vad som avses och då tar systemet det säkra före det osäkra och avbryter uppgiftsspecifikationen för att starta om från systemets initiala tillstånd (alltså där inga uppgifter specificerats). Detta syns tydligt i exempel 7 där ett fel liknande det som gav upphov till felsituationen i 6.2, det vill säga att taligenkänningen misslyckas. Eftersom roboten parafraserar *hela* uppdraget i 7.3 innebär användarens nekande (7.4) att systemet avbryter hela uppdraget eftersom det är det som är innebörden av frågan i 7.3. Om detta är ett korrekt beslut när det gäller utformningen av felhanteringsdialogen återstår att pröva i närmare tester med användare.

Exempel 6

- | | | |
|-------|--------------------------------------|-------------------------|
| (6.1) | A: <i>Get paper</i> | <i>Hämta papper</i> |
| (6.2) | R: <i>Where is <u>the coffee</u></i> | <i>Var är kaffet?</i> |
| (6.3) | A: <i>No!</i> | <i>Nej!</i> |
| (6.4) | A: <i>What is the object</i> | <i>Vad är objektet?</i> |

Exempel 7

- | | | | |
|-------|----|---|--|
| (7.1) | R: | <i>Where is the paper?</i> | <i>Hämta papperet</i> |
| (7.2) | A: | <i>In Karins office</i> | <i>Hämta papperet</i> |
| (7.3) | R: | <i>Get paper from <u>John's office</u>?</i> | <i>H.p. från John's kontor?</i> |
| (7.4) | A: | <i>No!</i> | <i>Nej!</i> |
| (7.5) | A: | <i>Cancelling mission <mission>*</i> | <i>Avbryter uppdrag <uppdrag>*</i> |

*) parafrasen av uppdraget utelämnas här ur exemplet

10.5 Fysisk utformning för talgränssnitt

Ett område där det idag råder stor osäkerhet kring interaktionssituationen är hur mikrofonen till systemet skall vara placerad. Vi har identifierat tre olika sätt att fånga upp användarens röst, upptagning kan exempelvis ske via:

- ett headset som bärs på huvudet
- en riktad mikrofon placerad på roboten
- en handhållen apparat (kommunikationsradio, telefon eller handdator)

Att låta användaren ha ett headset medger god ljudupptagning, vilket förbättrar resultatet av taligenkänningen.

För en uppsättning där datorn som hanterar taligenkänningen sitter på roboten skulle en riktad mikrofon kunna användas. Vi har istället valt en lösning där användaren har en trådlös förbindelse mellan headset-mikrofonen och datorn. För detta används professionell studioutrustning av den typ som används vid teveinspelningar. Denna typ av utrustning är förhållandevis dyr men det finns olika tillverkare som erbjuder trådlösa headset för diktering som är billigare.

En riktad mikrofon innebär att användaren slipper bära något med sig, i gengäld är risken att ta upp omgivningens ljud större, något som får negativa effekter för taligenkänningen. Användaren måste också hela tiden hålla sig inom det område där mikrofonen har sitt känslighetsområde. Alternativet med riktmikrofonen medger dock flera användare.

10.6. Terkoppling med gester

Vi har funnit att svarstid och återkoppling är avgörande för att nå en nivå där systemet är blir användbart. I de flesta talgränssnitt som finns på skrivbordsdatorer används någon form av grafisk komponent för att ge återkoppling till användaren om systemets aktivitet.

Talgränssnittet är tänkt att vara den primära interaktionsmodaliteten när roboten inte är nära användarens dator, då kontroll kan ske via det grafiska gränssnittet. Vi har undersökt två sätt att återkoppla information från dialogsystemet. Det ena sättet är en via en rad lysdioder och det andra är med hjälp av den dockliknande gränssnittsrobot, som vi kallar CERO (se avsnitt 6.10). Syftet med denna är att den ska kunna ge återkoppling på ett naturligt sätt till användaren. Om man betraktar CERO som en komponent i gränssnittet, så är avsikten med dess metafor att den ska leda tankarna till en chaufför eller en pilot som kör roboten.

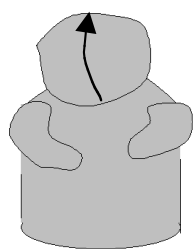
Ett viktigt argument för att använda en tredimensionell docka är att den ska kunna ses från flera håll. Dessutom gör den att roboten får en tydlig front. Genom att använda en docka som har mänskliga drag är det möjligt att sända kommunikativa signaler i form av gester på ett sätt

som liknar mänskliga gester, något gör att användaren inte behöver lära in ett nytt arbiträrt teckensystem. Vi anser att detta håller ned den kognitiva komplexiteten hos systemet.

En annan aspekt är CERO-dockans relativa litenhet, som vi anser bidrar till att roboten inte uppfattas som hotfull. CERO-dockan är mycket liten i förhållande till roboten (140 X 120 mm), dess huvud är stort och armarna korta.

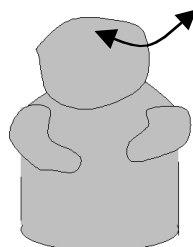
Vi har också undersökt möjligheten att använda oss av lysdioder för att ge återkoppling till användaren. Det är möjligt att sända ut likadana signaler med en rad av lysdioder. Skillnaden mellan CERO-dockan och lysdioderna är att blinkningarna hos dioderna måste följa ett konventionellt mönster av något slag, antingen det är en ljudstyrka som visas (som på en stereoanläggning) eller någon typ av systemtillstånd. Detta innebär att användaren måste lära sig vad signalerna betyder. Det mest givande sättet att använda lysdiodraden verkar vara att låta den indikera på ljudstyrka då det är en mycket vanlig konvention.

Dockans huvud kan röras framåt och bakåt och sidledes. Armarna kan röras uppåt och nedan. Detta innebär att de gester som dockan kan utföra är relativt få men trots det kan ett antal viktiga kommunikativa gester utföras (se figur 31 nedan).



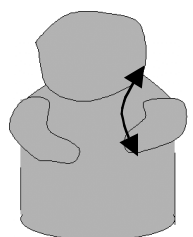
Uppmärksamhet

Uppmärksamhet signaleras genom att höja huvudet med en kort nickning.



Skaka/nicka

En skakning på huvudet kan användas på olika sätt, bland annat för att signalera att systemet inte kunde utföra en handling (tillsammans med återkoppling i form av ett talat felmeddelande).



Vinka

Armarna kan röra sig upp och ned oberoende av varandra. Detta kan användas för att vinka till användaren, exempelvis för att påkalla dennes uppmärksamhet.

Figur 31. Olika gester som kan utföras med hjälp av CERO.

10.7 J mf relse med andra typer av dialogsystem

Talgränssnitt till robotar skiljer sig från de talgränssnitt som används för diktering och telefonbaserade talsvarssystem. Vi har valt att utrusta roboten med en riktad mikrofon som har till uppgift att fånga upp användarens röstkommandon när roboten används ute i kontorsmiljön (se Figur 29). Det som skiljer olika talgränssnitt åt är, bland annat, användarnas rumsliga förhållande till systemet och de krav som systemet ställer på användarens perception. I det robot-

system som vi skisserar här är både användaren och roboten mobila i förhållande till varandra. I ett telefonbaserat system däremot, är användaren visserligen mobil rent fysiskt, men vid utformningen av dialogsystemet kan man alltid anta att användaren har sin telefonlur tryckt mot örat – därför kan man tala om två sorters mobilitet. Den första typen av mobilitet handlar om användarens förmåga att uppfatta samt avge kommandon. Den andra typen har mer att göra med användarens förhållande till den omgivande rumsliga kontexten.

System	Användnings-situation	System position	Återkoppling
Diktering	Vid datorskärm visuell återkoppling	Stationärt	Återkoppling via grafiskt gränssnitt
Telefonbaserade system	Luren hålls i handen systemet ej synligt	Stationärt (eftersom användaren alltid kan höra systemet)	Återkoppling via ljud t ex. inspelat tal/toner
Talgränssnitt till projektets robot	Användaren rörlig Avstånd varierar	Mobilt en av användaren känd miljö	Återkoppling via ljud eller rörelse (handlingar)

Tabell 3. Likheter och skillnader mellan olika typer av talade gränssnitt.

En annan skillnad mellan de mer traditionella systemen är möjligheten att ge direkt återkoppling till användaren. I ett dikteringssystem har man en mängd möjligheter att ge snabb och exakt återkoppling på vad användaren sagt. När användaren säger orden skrivs de ut direkt på skärmen samtidigt som en amplituden visas som en pelare. Detta innebär att användaren hela tiden kan kontrollera vad som dikterats. I ett telefonsystem har man endast möjlighet att ge återkoppling med ljud, antingen genom förinspelat tal eller genom rörsyntes. Systemet brukar normalt låta användaren avsluta sitt yttrande innan återkopplingen spelas upp. I ett robot-system som vårt kan både robotens rörelser (gester) samt förinspelat tal eller talsyntes användas för att ge återkoppling. Förutsättningen är dock att användaren och roboten befinner sig på samma plats. I tabell 3 (ovan) ges en översikt av olika egenskaper som påverkar användningssituationen.

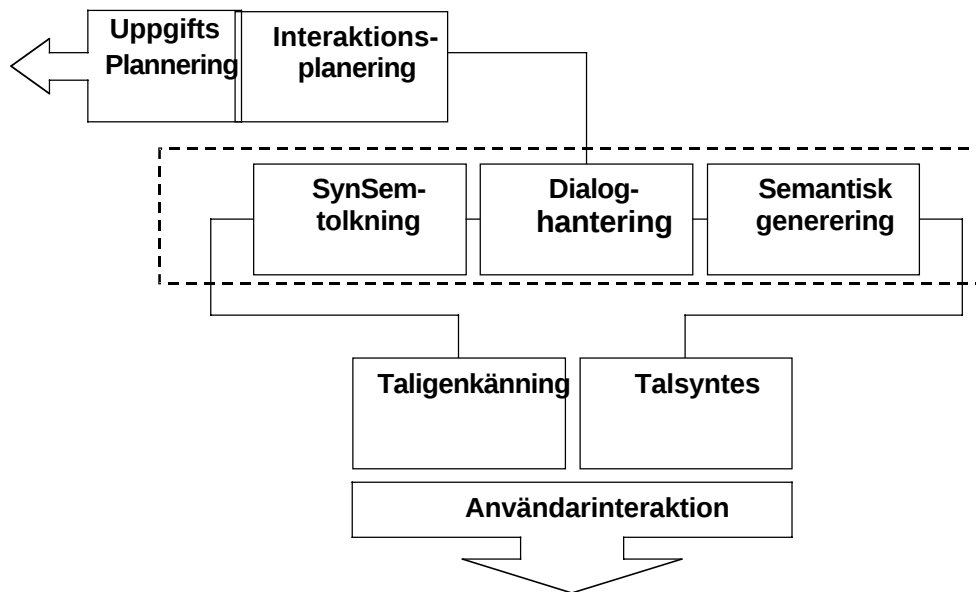
10.8 Systemutformning

Dialogsystemet består av ett antal komponenter där dialoghanteraren har till uppgift att tolka information från den syntaktiska och semantiska analysmodulen och sända information till genereringsmodulen och robotens planeringssystem. Figur 32 visar en konceptuell skiss av systemet. De delar i figuren som är inramade med en streckad ram innehåller systemets logik medan de övriga delarna (exempelvis taligenkänningen) används som tjänster och är mer eller mindre utbytbara.

De senaste årens utveckling av programvaror för taligenkänning och datorer med tillräcklig kraft för att bearbeta sammanhängande tal i realtid har i viss mån styrt utformningen av den talgränssnittet. Istället för att lägga stora insatser på att samla in fonetiska data och specialskriva program för vårt system har vi använt utvecklingsmiljöer och programbibliotek som kan kopplas ihop med programvaror som kan köpas färdiga ute på marknaden.

Systemets funktionella uppbyggnad illustreras bäst av ett exempel. När ett kommando yttrats tolkas användarens tal av taligenkänningsmodulen och omvandlas till en textsträng. Därefter analyseras strängen av den syntaktiska och semantiska analysmodulen med ett semantisk representation av yttrandet som resultat. Därefter försöker dialoghanteraren tolka yttrandet med utgångspunkt från den semantiska representationen. Tolkningen i dialoghanteraren syftar till

att avgöra vilken handling system ska utföra i nästa steg i interaktionen. I det fall systemet har för lite information för att kunna agera använder det genereringsmodulen för att formulera en begäran, eller fråga, om ytterligare specifikation från användaren.



Figur 32. En konceptuell systemskiss. Det streckade området utgör de delar som är specifika för systemet

När det gäller dialogsystemets tekniska egenskaper är en regelrätt kravspecifikation lik den som gjordes för det grafiska gränssnittet svår att göra i praktiken. Kraven på systemen kan snarare sammanfattas i ett antal generella designmål:

- Modulär uppbyggnad kring ett antal standardkomponenter som hanterar olika aspekter av tolkningen av användarens kommandon (bland annat taligenkänning och talsyntes).
- Systemet är avsett att vara plattformsoberoende.
- Kontinuerlig taligenkänning för ett antal engelska fraser. Taligenkänningskomponentens programvara är IBM ViaVoice

Vad gäller hur systemets plattformsoberoende kan realiseras i praktiken avgörs detta av programkomponenternas tillgänglighet för olika plattformar. För vårt system innebär detta vi endast kunnat testköra det till fullo på en PC med operativsystemen Windows och Linux. Den implementation som vi beskriver här körs med Windows som operativsystem. De programspråk vi använt i vår implementation är SICStus Prolog och Java, vilket medger att överflyttning (portning) enkelt görs till andra plattformar. För den syntaktiska och semantiska analysen och genereringen har Prolog använts medan Java har använts för att länka ihop programmet taligenkänning och talsyntes samt som gränssnitt till robotens övergripande kontrollsystem.

Taligenkänningssystemet är den komponent som kräver mest av hårdvaran. Vi använder IBM ViaVoice med den utvecklingsprogramvara som kan laddas ner från IBM:s webbplats. För att kunna användas praktiskt kräver detta system en PC med Windows som operativsystem, med en processorhastighet på ca 350 – 700 MHz. Mikrofonen bör vara av god kvalitet, men det krävs inte att den är av ett särskilt fabrikat. Vi har använt ett headset från Plantronics som kostar ca 200-300 kr.

På en del plan har detta givetvis lett till att vi måst sänka våra förväntningar på vad som kan kännas igen. Exempelvis kan vårt system inte identifiera olika talare utan måste snarare tränas

för en enda användare för att kunna användas praktiskt. Taligenkänningsprogrammet är inte heller främst inriktat på talspråk utan på dikterat tal. Programvaror för taligenkänning utvecklas dock kontinuerligt vilket lovar gott för framtiden.

10.9 Resultat

Fortfarande återstår mycket att göra när det gäller robusthet och repertoar i systemet. Den implementation vi har gjort stödjer, i begränsad omfattning, ett naturligt och flexibelt interaktionssätt. Denna flexibilitet är en direkt följd av att generella principer för mänsklig dialog beaktats och använts i vid utformningen av systemet.

Vad gäller de tekniska delarna av implementationen kan vi konstatera att den till stora delar är gjord med programkomponenter som kan köpas färdiga. De delar som är uppgiftsspecifika: grammatisk analys och dialoghantering samt de domänspecifika delarna (till exempel robotens ordförråd och uppgiftsplaner) har dock utvecklats med den här specifika roboten i åtanke. Enligt vår mening är det dock förhållandevis enkelt att anpassa och modifiera systemet till andra robotplattformar, förutsatt att uppgifterna som roboten ska lösa är likartade.

En fortsatt utveckling av den här roboten skulle kunna inriktas på två områden, dels en utveckling av ett helsvenskt system och dels att arbetet fokuseras på utöka systemet robusthet och kompetens.

Svensk implementation: Vad gäller möjligheten att överföra systemet till svenska vore det i princip möjligt att göra detta idag, men tillgänglighet på mjukvarukomponenter för taligenkänning är den osäkra faktorn. Talsyntes finns däremot sedan länge på marknaden. Vi har undersökt ett antal taligenkänningsmotorer och har funnit att det verkar möjligt att använda dessa även för detta system. Vi har gjort några praktiska försök i den riktningen men inte gjort någon regelrätt implementation.

Robusthet: Det nuvarande systemet känner igen förhållandevis få kommandon. Användaren behöver kunna använda synonymer och ett ledigare språk överhuvudtaget. Systemet har dessutom svårt att veta om det blir tilltalat vilket gör att det ibland tror sig känna igen saker som inte sagts. Detta är delvis relaterat till den underliggande taligenkänningskomponenten men det har också att göra med mikrofonens prestanda. En bra mikrofon kombinerat med en fonetisk modell anpassad för talade kommandon med en riktmikrofon skulle ge bättre prestanda för taligenkänningssystemet.

Reparationsdialog: Dialoghanteraren i systemet har små möjligheter att reparera dialogen när det blivit fel i igenkänningen. Detta innebär att systemet ibland måste börja om, vilket leder till sämre användbarhet.

11. Utvärdering med användare

Vår robot har utvecklats iterativt i syfte att framställa en hjälpmedelsrobot som kan skickas till olika platser i kontorsmiljö av folk med rörelsehandikapp. För att utvärdera om våra gränssnitt och robotens design i dag fungerar och motsvarar användarens förväntningar har vi inlett de första användarförsöken med det färdigställda systemet under november och december 2000. Dessutom kan försöksverksamheten ses som ett förberedande steg i fortsatt vidareutveckling av och fördjupad forskning om serviceroboter som hjälpmedel.

I detta kapitel ges först en introduktion till försökets uppläggning, där användare, uppgifter och insamlade data beskrivs. Genomförda försök redovisas sedan mer detaljerat för att tydliggöra vad vi har lärt oss och vilka slutsatser vi tror oss kunna dra. Slutligen ges förslag till hur arbetet med att demonstrera roboten för tänkta användargrupper och låta dem praktiskt delta i robotförsök kan fortsättas.

11.1 Försöksuppsättning

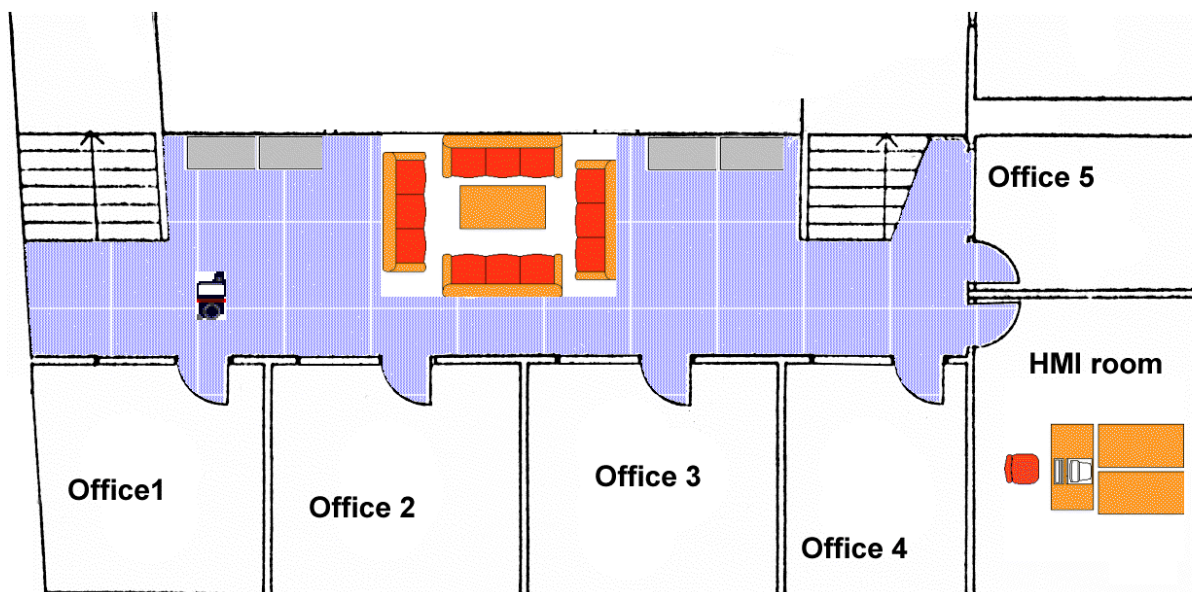
Som motivering för våra praktiska användarförsök kan sägas att vi ville testa vårt nuvarande robotsystem (se avsnitt 6) genom att låta "användare" (se nedan) köra roboten med hjälp av vårt grafiska gränssnitt (se avsnitt 9) och vårt talgränssnitt (se avsnitt 10 ovan). Vår förväntning var att en sådan omgång skulle kunna ge oss återkoppling om i vilken grad vi har (miss-)lyckats att ge roboten ett adekvat utseende, en funktionalitet och ett bra gränssnitt för att hantera roboten.

Vi försökte få svar på tre frågor:

- 1) Motsvarar roboten i sin nuvarande utformning och funktionalitet användarens förväntningar på en servicerobot med transportuppgifter?
- 2) Kan nybörjare använda vår robot efter en kort inlärningsperiod för att utföra de uppgifter som roboten var tänkt för, både med det grafiska gränssnittet och vårt talgränssnitt?
- 3) Kan vår servicerobotprototyp utvärderas av icke-expertanvändare som utför uppgifter med roboten?

Med denna frågeställning i bakhuvudet satte vi upp våra försök i en kontorsmiljö på KTH och bjöd sedan in användare att testa roboten och dess gränssnitt. Rumstekniskt var uppsättningen tvådelad: Roboten rullar omkring i ett rum som också kallas för "lilla världen", ett större utrymme (ca. 4 x 16 meter) framför sex kontor. En soffgrupp och två skåp finns där, samt två trappor som leder till andra korridorer. Roboten har olika platser i denna miljö som den kan åka till. Huvudsakligen är dessa "målpunkter" framför ingångsdörrarna till de sex kontor som finns där. Användaren kommer under användningen av det grafiska gränssnittet att sitta i ett av dessa kontor, också kallat för "HMI-läsrummet".

En spatial översikt över denna miljö ges i Figur 33. Poängen med denna uppdelning är att placeringen skulle motsvara ett normalt transportrobotsscenario. Anställda kan sitta i sitt kontor och skicka roboten på olika turer att hämta eller leverera saker och ting. Samtidigt går det att titta efter roboten och hur den beter sig genom att användaren går till HMI-rummets dörr varifrån det går att se roboten.



Figur 33. Karta över miljön för robotförsöket; roboten kan åka på det blå golvutrymmet

I del två av användartestet var det tänkt att talgränssnittet skulle testas. För det ska användaren förflytta sig till "lilla världen"-utrymmet. I denna situation befinner sig därmed roboten och användaren i ett och samma rum, dvs. roboten kan tilltalas direkt. Eftersom roboten försöker föra en dialog med användaren genom att "lyssna" till inkommande tal samt svara med tal-syntes (se talgränssnittsdelen, sidan 48) så är det fördelaktigt att användaren befinner sig i fysisk närhet av roboten.

Genom att beskriva försöksområdet har redan en del om själva försöket avslöjats, men en mer detaljerad beskrivning följer nedan.

Användarna blev inbjudna till våra försökslokaler, och fick se roboten som var uppställd i den "lilla världen". Efter en introduktion om försöket (vad som skulle testas, vem som är ansvarig, syftet, datainsamlingsmetod etc.) så fick användaren en kort PowerPoint-presentation av roboten, dess funktion och gränssnitt. Detta animerade multimedia-bildspel visades på en dator i HMI rummet. Härnäst blev användaren ombedd att svara på några allmänna frågor om sig själv, dvs. bakgrund (ålder, handikapp mm) och tekniskt kunnande (yrke, datorvana, roboterfarenhet mm). Sedan började vi med att låta användaren köra roboten ett tag med det grafiska gränssnittet för att bli van vid det. När användaren kände sig någorlunda bekant med att hantera gränssnittet fick han/hon två uppgifter. Uppgift ett var att köra roboten till en förbestämd plats i "lilla världen" och uppgift två var att transportera en bok från ett kontor till ett annat. Efter varje uppgift skulle användaren bedöma hur lätt eller svårt det var att utföra uppgiften samt kommentera sin bedömning.

Efter detta fick användaren träna upp talsystemet. Detta sker genom att man går genom en ca. 10 minuters sekvens med diktering, då man läser upp texter som visas på skärmen. Genom att känna igen dessa texter kan talsystemet i datorn bygga upp en modell över hur olika användare uttalar ord. Även denna del av försöket betygsattes av användaren.

Härnäst förflyttade sig användaren till "lilla världen" för att använda talsystemet för att köra två ytterligare transportuppgifter. Användaren fick ett A4-papper med olika fraser och ord som talsystemet kan hantera. Med denna "fuskklapp" bad vi användaren att åter förflytta robo-

ten och att leverera en bok från ett kontor till ett annat. Liksom vid uppgifterna för det grafiska gränssnittet bad vi användaren att betygsätta svårigheten och sin belåtenhet med talgränssnittets förmåga att lösa uppgifterna.

Slutligen frågade vi om våra testpersoner kunde svara på några frågor som berörde robotens utformning, CERO-figuren (se avsnitt 6.10) samt robotens funktionalitet som hjälpmedel. Detta enkät del var också utformad så att man kunde lämna fria kommentarer om intryck som man fått av robotkörningen och hela testförloppet.

Försöket tog cirka en timme. På grund av problem t ex med inspelningsteknik, robotens teknik och i början även försöksuppsättningen så kunde denna tänkta tidsram överskridas.

Data som insamlades under försöket består av enkätsvar, videoinspelningar samt loggfiler som producerades t ex genom de knapptryckningar som användaren utförde i det grafiska gränssnittet. Som exempel visas följande stycke av en sådan loggfil för det grafiska gränssnittet:

```
[Fri Nov 17 11:01:17 GMT+01:00 2000] Go-To ComboBox changed to: Karin
office
[Fri Nov 17 11:01:17 GMT+01:00 2000] Go-To ComboBox changed to: Karin
office
[Fri Nov 17 11:01:27 GMT+01:00 2000] GUI Elements disabled
[Fri Nov 17 11:01:27 GMT+01:00 2000] Robot send on Go-To Mission
Destination: office4
[Fri Nov 17 11:01:30 GMT+01:00 2000] User feedback dialog: Robot is now
on a mission
[Fri Nov 17 11:01:30 GMT+01:00 2000] GUI Elements disabled
[Fri Nov 17 11:01:30 GMT+01:00 2000] Robot got a new GoToMission - GUI
disabled
[Fri Nov 17 11:01:30 GMT+01:00 2000] GUI Elements disabled
[Fri Nov 17 11:01:30 GMT+01:00 2000] Robot got a new GoToTask - GUI disabled
[Fri Nov 17 11:01:30 GMT+01:00 2000] CERO animation changed to: 0
[Fri Nov 17 11:01:30 GMT+01:00 2000] CERO animation changed to: 1
[Fri Nov 17 11:01:30 GMT+01:00 2000] Robot in Map now at:78 57
[Fri Nov 17 11:01:30 GMT+01:00 2000] Robot in Map now at:79 56
[Fri Nov 17 11:01:34 GMT+01:00 2000] Robot in Map now at:81 57
[Fri Nov 17 11:01:39 GMT+01:00 2000] Robot in Map now at:91 60
[Fri Nov 17 11:01:39 GMT+01:00 2000] Robot in Map now at:99 63
[Fri Nov 17 11:01:40 GMT+01:00 2000] Robot in Map now at:109 64
```

I första kolumnen står datum och tidpunkten när en händelse inträffade, i kolumn två noteras vilken komponent som berörs och slutligen vilken aktion som utfördes. Aktioner som noteras är t ex vilka grafiska element som användaren har aktiverade, vad som visas på skärmen och robotens status. När roboten rör sig i miljön (och som en symbol i gränssnittet) noteras t.ex. olika punkter som roboten befinner sig på. Notationen är då "Robot in Map now at 78 57, Robot in Map now at 79 56" osv.

Efter den här introduktionen till vår försöksuppläggning följer en mer detaljerad beskrivning av genomförda försök samt resultaten innan vi drar några preliminära slutsatser.

11.2 Genomförelse

Inledningsvis får det sägas att våra användarförsök inte är avslutade samt att det är för tidigt att ge sig på detaljerad redovisning av resultaten, eftersom vi ännu inte har analyserat vårt material fullständigt. Ändå beskriver vi här våra försök för att ge ett första intryck av våra observationer.

Vi valde i första hand användare som lätt kunde bjudas in för försök, dvs. studenter (ingen hade ett handikapp) och projektansvariga professorn (eftersom hon representerar vår användargrupp). När försöksuppsättningen och undersökningen i alla detaljer är prövade i flera omgångar är det tänkt att vi antingen ska utvidga försökspanelen (t ex med kontakter som vi fick genom Inger Rundström) eller gå över till mer långvariga försök, t ex låta anställda i vårt forskningsavdelning köra roboten över flera veckor.

Vi körde sammanlagt fyra försök som kort ska redovisas. Två manliga, tekniska studenter, en kvinnlig professor samt en kvinnlig användbarhetskonsult körde vår robot. Alla användare var datorkunniga samt var flytande på engelska. Bara professorn hade kört en (nämligen vår) robot tidigare. För de andra var försöket första gången som de arbetade med en robot och ett robotgränssnitt.

Studentanvändaren 1 tyckte att det var mycket lätt att köra uppgifter med robotens grafiska gränssnitt. Dock tyckte han att ett dialogfönster som kom upp som bekräftelse var störande. Den första uppgiften som utfördes med talsystemet var också lätt, men han tyckte att det var svårt att utföra den andra uppgiften med tal. Han motiverade sin bedömning med att han förväntade sig att kunna använda ett kommandospråk och inte bli involverad i en dialog med roboten. Prototypen bedömdes se ut som en robot och användaren tyckte om utseendet. CERO-figuren bedömdes som ”cool” och ”interesting”. Ett intressant svar fick vi på nästa fråga om han kunde föreställa sig att använda en sådan hjälpmedelsrobot själv. Studenten svarade på frågan med ”ja, det kan [han] föreställa sig, men inte för att [han] har behov, men för att den [roboten] är kul!”. Som generella kommentarer återkom dock att han inte tyckte om talsystemet och att han ansåg att roboten var för långsam.

Student nummer två tyckte det var lätt att utföra den första uppgiften med det grafiska gränssnittet, men ansåg att det gick dåligt att utföra den andra uppgiften. Här måste sägas att roboten upphörde att fungera under den andra uppgiften, och det är sannolikt att den negativa bedömningen berodde på detta systemfel.

Talträningen fick mindre bra bedömningar. Den ansågs vara för lång, för svår och förvirrande. Bägge uppgifterna som skulle utföras med talsystemet kommenterades att vara lätta att utföra, även om roboten gick sönder en gång emellan. Användaren tyckte om roboten och fann att den var praktiskt användbar och nyttig. Talsystemet fick beröm att vara excellent.

Inför beskrivningen av den tredje personens robottestande får det sägas att hon är projektledare och därför själv engagerad i projektet. Det är bara naturligt att hennes uttalande inte kan vara helt opartiska – samtidigt är hon per projektdefinition en bra representant för vår tänkta användargrupp och mycket intresserad att föra robotanvändbarhetsforskningen framåt. Efter utförande av deluppgift ett och två (under användning av det grafiska gränssnittet) betygsatte användaren användbarhet med en ’3’ respektive en ’4’ på en skala av 1-7 (med ’1’ som *mycket lättanvänd* och ’7’ som *mycket svårt att använda*). Robotens system gick ner under uppgift nummer ett, dvs. användaren kommenterade att *”roboten fungerade först på 2:a försöket”*. Kommentaren till den andra uppgiften uppmärksammade ett logiskt fel: Användaren skulle, med robotens hjälp, transportera en bok till en annan person. Boken skickades med ett

”Go-To” uppdrag till mottagarens kontor. Det var dock tänkt att roboten skulle skickas iväg med en ”Deliver” uppdrag. Skillnaden är, att en ”Go-To” bara förflyttar roboten från en punkt till en annan. Däremot utlöses en dialog av roboten när man ger den ett uppdrag med ett ”Deliver”-kommando. Skickar man t ex roboten med ett föremål till en annan person, så frågar roboten när den har anlänt vid mottagaren, om föremålet kan tas emot. Eftersom användaren valde ”fel” uppdrag, utfördes dock ingen sådan kommunikation på mottagarens plats, dvs. uppdraget slutfördes med att roboten kom fram till målpunkten. Vår försöksperson sa då att *”jag visste inte om [namn av person anonymiserad] hade hämtat boken”* för att påpeka bristen på återkoppling i gränssnittet. Detta är en korrekt observation, problemet ligger dock, som nämnts ovan, djupare: Det går att ”utföra” uppdrag på flera alternativa sätt – även om detta innebär att inte ”rätt” funktion används, vilket kan leda till flertydigheter och suboptimalt gränssnittsagerande.

Den följande talträningen upplevdes igen som för lång, förvirrande och utmattande. När försökspersonen sedan styrde roboten med tal bedömdes det vara relativt lättanvänt (uppgift ett fick betyg ’3’ och uppgift två ’2’). Negativa kommentarer fick dock själva försöksuppsättningen: Att ta reda på *”vad-man-kan-säga-till-robot-lathunden”* (A4 pappret) samtidigt som enkäten och talutrustningen måste medföras är *”svårt att hantera”*.

Bland annat antecknades denna punkt för en framtida förbättring av försöket (se nedan). Roboten som helhet var omtyckt, CERO karaktären kryssades in att vara *”childish”, ”playful”, ”interesting”* och *”fun”*. Viktigt också: Användaren tyckte att det är möjligt att använda roboten som assistans och tros kunna verkligen utföra någonting praktiskt. Både det grafiska och talgränssnittet var uppskattat, dock antecknades att ett dialogfönster som kommer upp efter utförande av ett uppdrag var mindre bra (samma observation som tidigare, se ovan).

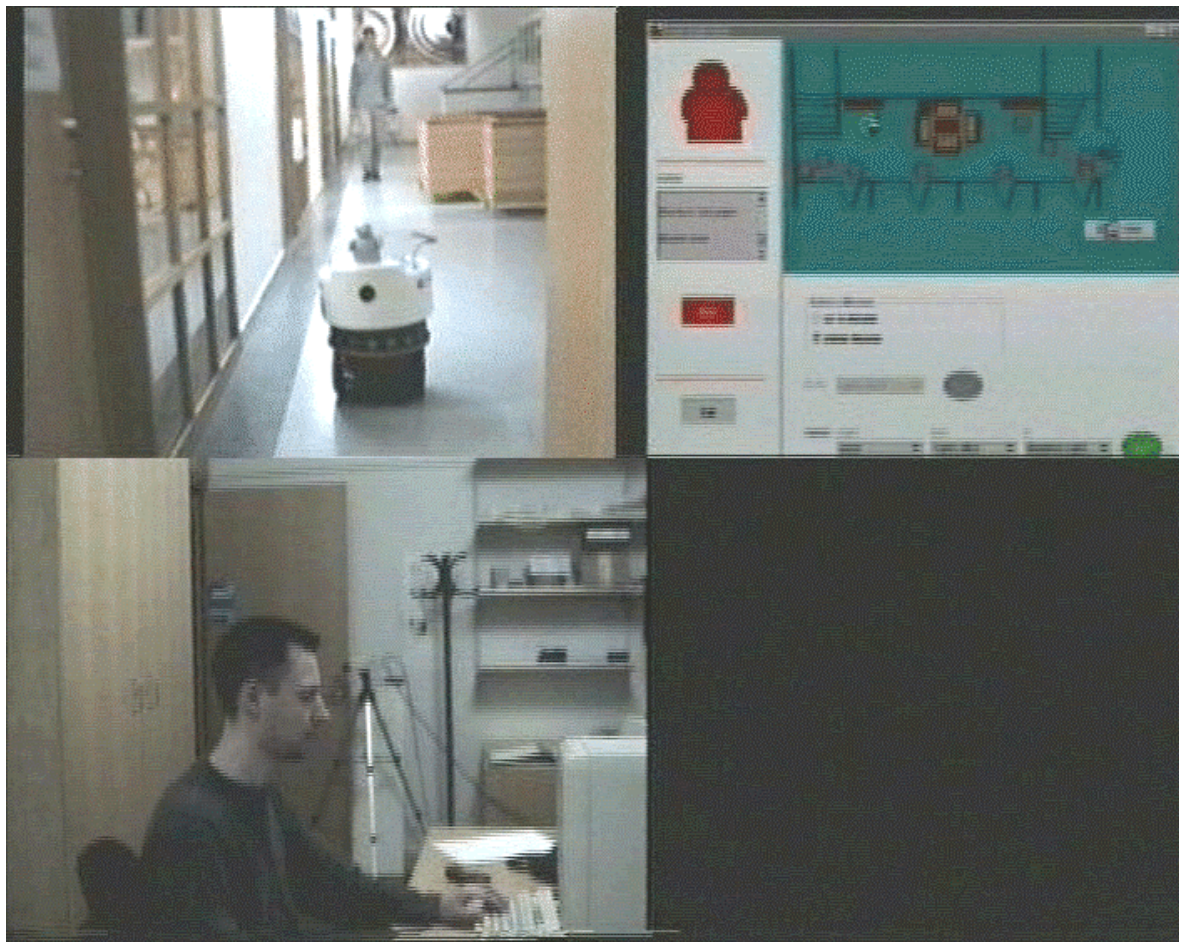
Efter denna försöksomgång diskuterade vi i detalj om försökets uppbyggnad verkligen är adekvat för att svara på våra frågeställningar. Ett antal punkter kritiserades, t ex försöket som helhet är avsett bara för att testa nybörjare. Alldeles för få uppdrag körs, varför det kan vara svårt att uttala sig om hur våra gränssnitt skulle bedömas långsiktigt, dvs. inlärningseffekterna mäts, inte verkligt användande av roboten. Detta återspeglas också genom att man inom loppet av cirka en timme bara får köra roboten några gånger. Robotintroduktionen och presentationen av uppgiftsscenario samt talträning tar för stor andel av försökstiden, tycktes det i vårt interna diskussion. Som lösningsförslag kom vi fram till att det är värt att prova att dela upp försöket i två omgångar. I den första sessionen får användare en introduktion till roboten och får sedan träna upp talsystemet. I den andra omgången körs roboten under en längre tidsperiod och på ett annat sätt. Idén var att låta två personer jobba ihop genom att skicka olika föremål till varandra med hjälp av roboten i stället för att utföra uppdrag *”ensam”* med roboten. Detta skulle bättre motsvara en naturlig arbetssituation där roboten ingår som ett verktyg i samarbete och inte, som hittills, bara fokusera på robotens gränssnitt under utförande av uppdrag.

I försök nummer fyra ändrades uppsättningen enligt de ovan diskuterade idéerna. Vi började användartestet med nästa försöksperson genom att ge vår robotintroduktion samt låta henne träna upp talsystemet med sin röst. Uppdelningstanken visade sig var bra eftersom användaren tyckte att talträningen var utmattande. I detta läge ansåg vi det vara bäst att avsluta del 1 av robotförsöket för att träffas igen efter några dagar och fortsätta med att köra del 2 av försöket, dvs. verkligen prova roboten.

Efter några dagar träffade vi försökspersonen igen och då var det flera uppdrag som gick ut på samarbete mellan två parter som skulle utföras. Under användning av det grafiska gränssnitt

fick vi fram intressanta observationer som motsvarar den diskussionen som hade lett till att prova den nya försöksuppsättningen. Användaren skulle utföra sammanlagt 5 uppgifter med det grafiska gränssnitt och bedöma svårighetsgraden att utföra uppgifter. Det visade sig att det första uppgift fick betyg '3' (samma skala som ovan användes), den andra likadant, den tredje och fjärde omgången sattes en '2' och slutligen var sista omgången så lätt att användaren satte en '1' (som motsvarar bedömningen "mycket lättanvänd").

Detta kan tolkas som att det bara tar några omgångar att köra roboten för att lära sig hur det grafiska gränssnittet skall användas. Har man lärt sig detta, är det lätt att sedan köra roboten.



Figur 34. exempel på videoinspelning med användaren, robot och det grafiska gränssnittet

En annan observation och dess eventuella konsekvenser har vi ännu inte färdiganalyserat och diskuterat: Gränssnittsdesignen utgick ifrån ett s k *single-user* scenario, dvs robotens gränssnitt används av en bestämd person vid en bestämd tidpunkt. Detta antagande är dock minst delvis fel när två (eller fler) personer skall jobba med en och samma robot, då var och en av de involverade parterna behöver ett gränssnitt för att kunna köra roboten med. Här uppstår därför två intressanta frågeställningar: Tillgångsrättigheter och eventuella användningsprioriteter för en gemensamt utnyttjad "resurs" (läs roboten) måste hanteras tekniskt och kommuniceras i de olika användargränssnitten. Det är t ex en spännande fråga vem som ska kunna ta kontroll över roboten då den befinner sig på väg mellan två personer. Ska mottagaren kunna styra roboten först när den har kommit fram, när utgångsoperatören har släppt kontroll, eller kanske när roboten har kommit halvvägs?

En annan utmaning, som samtidigt observerades, är att när två personer använder roboten för att samarbeta så finns det minst en kommunikationssituation till som måste hanteras. Hittills har vi bara diskuterat kommunikationssättet mellan användaren och robotsystemet, dvs. de gränssnitt som möjliggöra kommunikation emellan dem. Vi kunde dock notera att båda samarbetspartner också kommunicerade med varandra, t ex när roboten skulle skickas tillbaka och den ena person ringde upp den andra personen för att tala om för denna att ”nu har jag skickat roboten tillbaka till dig”. Vi har, utöver kommunikation mellan användare och systemet fått en *metakommunikation* mellan olika involverade personer som delar roboten emellan sig.

Tyvärr gick talsystemet ”sönder” när vi skulle testa att köra roboten med det så att vi var tvungna att avbryta denna försöksomgång utan att testa talsystemet med den nya försöksuppsättningen.

11.3 Slutsatserdiskussion och rekommenderad fortsättning

På grund av att få användare kunnat delta i försöken, och att alla data ännu inte har analyserats, får resultaten i detta avsnitt tolkas med en viss försiktighet. Det kan ändå vara på sin plats att försöka sig på några försiktiga slutsatser för att besvara de tre frågor som vi ställde oss i början (se ovan).

Om vi börjar med fråga tre om det är möjligt att användartesta vår framtagna prototyp med icke-expert genom att låta dem utföra ett antal uppgifter, så kan den besvaras positivt. Visserligen hade vi fortfarande (alldeles för) många systemfel och komponenter som strejkade mitt under försökskörningen, men de flesta systemfel tror vi att vi har åtgärdat nu. Samtidigt verkar det som användare störs mindre i sin bedömning av gränssnittet och roboten när en systemkomponent slutar fungera eftersom de är medvetna om att det är en prototyp. Däremot kan man också avläsa av deras yttrande att de förväntar sig att robotens stabilitet komma att vara förbättrad, när en sådan apparat kommer ut på marknaden och säljs kommersiellt.

Det grafiska gränssnittet fick i genomsnitt ganska bra betyg. Några större problem med att använda det förekom inte, och användarna kunde utföra sina uppdrag. Som svag punkt kritiserades en återkopplingsdialog, då den kommer upp med ett ”pling-ljud” och ser ut som ett felmeddelande. Det var missvisande (eftersom den faktiskt talar om för användaren att uppdraget nu är utfört och klart) och förvirrade användaren. Den irritation som detta gav upphov till gjorde att vi ändrade återkopplingsmetoden.

Talgränssnittet fick beröm men underkändes också delvis då det inte verkar fungera klanderfritt för användaren. Det är dock för tidigt att uttala sig om detta beror på systemets ostabila teknik, eller, mer allmänt om detta sätt att styra en teknisk apparat, som dessutom rör på sig, var för oprövat för användare. Alla våra användare kunde hantera och hade erfarenheter av ett fönsterbaserat grafiskt gränssnitt. Däremot hade ingen av våra försökspersoner tidigare använt ett interaktivt system som bara styrdes med tal – att jämföra både interaktionsmetoderna på lika villkor skulle följaktligen vara fel redan pga av de olika nivåer av tidigare erfarenhet som folk kan förväntas ha.

Det krävs också flera försöksomgångar för att få en bättre förståelse av hur talgränssnitt påverkar människa-robotinteraktionen. Det verkade t.ex. mycket enkelt och användbart, när det fungerade som planerat. Folk var verkligen imponerade och såg roboten med detta talade dialoggränssnitt som ett väldigt intressant redskap. Samtidigt verkar de flesta användare ha svårt med talträningen som vi i dagens version kräver för att ställa in systemets röstigenkänningsmodell. Även här finns antingen behov av förbättrade metoder eller så används en tek-

nisk lösning som innebär att man slipper en uppträning av talsystemet helt, dvs. man använder sig av ett taligenkänningssystem som är talaroberoende.

Som ett sista argument för en fortsatt försöksserie kan man påpeka situationen som vi provade: Folk fick en relativt kort introduktion och lärde sig systemet och de olika gränssnitten. De fick uppdrag som de skulle lösa med roboten, en hittills okänd apparat. Bara några få omgångar kördes och dessa få omgångar skulle betygsättas. Denna förenkling av försökssituationen behöver inte motsvara hur folk kommer att använda sig av roboten och de olika gränssnitten om de skulle kunna få tillgång till roboten under längre perioder. Kanske är det då jobbigare att öppna det grafiska gränssnittet bara för att köra roboten till någon plats? Kanske har man inget behov av att följa robotens rörelse och övervaka den på detta sätt när man har kört roboten framgångsrikt under flera veckor? Sådana frågor kan enbart besvaras av försök med en robot som kan försiggå under en längre period.

Som ett nytt scenario fick vi fram skillnaden mellan en *single-user* och ett *multi-user* scenario. I det sistnämnda behöver gränssnittet ha stöd och mekanismer för att hantera delningen av en robot i t ex en arbetsgrupp. Eftersom hjälpmedelsrobotar är så pass nya är det oklart hur *personliga* de kommer att vara samt vilka konsekvenser en möjlig delning mellan olika personer skulle kunna ha. Detta område verkar mycket givande då det kan vara möjligt att robotar kommer att bli sociala agenter som gemensamt utnyttjas t ex på jobbet eller hemma i familjen.

Vi har, som tidigare sagts, bara kunnat göra några initiala observationer och kommer att fortsätta med ytterligare försök under våren 2001.

12.Slutord

Denna rapport har beskrivit ett 2-årigt projektarbete kring utveckling av en rörlig transportrobot som stöd för funktionshindrade i arbetslivet. Designarbetet har omfattat ett antal steg såsom en ingående analys av systemets uppgifter, simulering av användarinteraktionen, programmering av ett generellt styrsystem samt konstruktion och utvärdering av flera gränssnittsprototyper. Projektet har resulterat i en fungerande robotprototyp, CERO, som har börjat utvärderas med användare, vilket har lett till preliminära resultat och slutsatser angående prototypens användbarhet.

Det står redan klart att det krävs betydligt mer omfattande försök för att kunna dra allmänna slutsatser om hur gränssnitt mellan människa och robot bör utformas för användning i denna typ av situationer. Framför allt behövs longitudinella, kvalitativa studier där samma användare får använda roboten i sitt dagliga arbete under en längre tid, samt iterativ förbättring av prototypen i ljuset av de observationer som görs. Här krävs inte bara hänsyn till de funktionshindrade målanvändarnas behov, utan också en analys av hur roboten kan fungera i arbetsgruppen som helhet.

Ett problem i utvecklingsarbete av denna typ, som också avspeglas i rapporten, är att flera olika verksamheter normalt måste pågå parallellt. Vidare inträffar ofta viktiga moment och framsteg i en annan ordning än den planerade. Till exempel har arbetet med systemkonstruktion och med gränssnittsdesign till stora delar skett samtidigt och under livliga diskussioner, vilket lett till många revideringar i båda delarna av projektet.

Arbetet har komplicerats av att det saknas etablerade metoder för användarorienterad utveckling av rörliga robotsystem, men kan förhoppningsvis bidra till en utveckling av sådana metoder. Området har stora utmaningar, då det för närvarande råder en diskrepans mellan de löften som formuleras på robotikens område, och förekomsten av praktikfall där robotsystem faktiskt används och utvärderas på ett systematiskt sätt. En speciell utmaning ligger i den allmänna nyfikenhet som området ofta väcker hos en teknikintresserad allmänhet, samt intresset för robotdesign hos yrkesgrupper som IT-utvecklare, industridesigners och mediaföreträdare.

Forskningsområdet människa-robotinteraktion är i sin linda, och projektet har varit en period av lärande. Vi har redan påbörjat nya studier som fortsättning och uppföljning av projektets idéer. Det står klart att den prototyp vi utvecklat kommer att bli ett viktigt verktyg för en fortsatt utveckling av både användarscenarion och gränssnitt, trots dess begränsningar i många avseenden. En prototyp av denna typ kan ses som ett viktigt instrument för kommunikation kring utformning av ny teknik. Vi kommer att använda den som utgångspunkt för fortsatta diskussioner kring hur framtida möten mellan människor och autonoma rörliga system kan gestaltas, och kring de forskningsproblem som dessa möten aktualiserar.

13.Referenser

13.1B öcker och artiklar

Allwood, J., J. Nivre and A. E. (1991). On the Semantics and Pragmatics of Linguistic Feedback. *Gothenburg Papers on Theoretical Linguistics*(64).

Andersson, M., A. Öreback, M. Lindström and H. I. Christensen (1999). An Intelligent Service Robot. *Intelligent Sensor Based Robotics*. H. I. Christensen, Bunke and Noltemeier, Springer Verlag.

Apple (1992). *Macintosh Human Interface Guidelines*. Reading, Massachusetts, Addison-Wesley.

Bernsen, N. O., L. Dybkjer and H. Dybkjer (1998). *Designing Interactive Speech Systems – From First Ideas to User Testing*. Berlin, Springer-Verlag.

Black, A. W., P. Taylor and R. Caley (2000). Homepage of the festival Speech Synthesis System., University of Edinburgh, Centre of Speech Technology Research. **2000**.

Bolmsjö, G. and H. Eftving (1993). *RAID Robotic End-effector developments*. Proceedings of the 1st TIDE Conference, p. 108 - 112. Brussels, IOS Press.

Clark, H. H. (1996). *Using Language*. Cambridge, Cambridge University Press.

Clarkson, P. J., R. Dowland and R. Cipolla (1997). *The use of prototypes in the design of interactive machines*. Proceedings of the International Conference on Engineering Design, p. 735 - 740.

Dahlbäck, N., A. Jönsson and L. Ahrenberg (1993). Wizard of Oz studies - why and how. *Knowledge-Based Systems* **6**(4): 258-256.

Dallaway, J. L., R. M. Mahoney and R. D. Jackson (1993). *CURL - a robot control environment for Microsoft Windows*. Proceedings of the RESNA 93 Annual Conference, p. 510 - 511.

Diaper, D., Red. (1989). *Task Analysis for Human-Computer Interaction*. Chichester, England, Ellis Horwood.

Dix, A., J. Finlay, G. Abowd and R. Beale (1998). *Human-Computer Interaction*. London, England, Prentice Hall.

Dowland, R., R. Cipolla and P. J. Clarkson (1997). *A prototype for interactive robotics application development*. Proceedings of the fifth International Conference on Rehabilitation Robotics, p. 71 - 74.

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

Eftving, H. (1999). The useworthiness of robots for people with physical disabilities, Certec: Department of Design Sciences, Lund University.

- Eftring, H. and K. Boschian (1999). *Technical Results from Manus User Trials*. Proceedings of ICORR'99, Sixth International Conference on Rehabilitation Robotics, p. 136ff. Stanford, USA.
- Ejiri, M. (1995). *Towards Meaningful Robotics for the Future*. Proceedings of the International Workshop on Biorobotics: Human-Robot Symbiosis, p. 5 - 6. Tsukuba, Japan, Elsevier Press.
- Engelhardt, K. G. and R. A. Edwards (1992). Human-robot integration for service robotics (Chapter 16). *Human-robot interaction*. M. Rahimi and W. Karwowski. New York, Taylor & Francis: 315-346.
- Fridenfalk, M., U. Lorentzon and G. Bolmsjö (1999). *Virtual Prototyping and Experience of Modular Robotics Design based on User Involvement*. Proceedings of the AAATE99 Conference, p. 308 - 315. Düsseldorf, Germany.
- Hüttenrauch, H. (under utg.). *Wizard of Oz Study in Robotics, Course Report on HMI651: Research methods in Human-Computer Interaction* Stockholm, IPLab, NADA, Kungliga Tekniska Högskolan.
- Johannesen, L. and D. D. Woods (1991). *System, Man and Cybernetics*. IEEE International Conference on Decision Aiding for Complex Systems, p. 1337 - 1341.
- Kawamura, K., R. T. Pack, M. Bishay and M. Iskarous (1996a). Design Philosophy for Service Robots. *Robotics and Autonomous Systems* **18 (1-2)**(International Workshop on Biorobotics: Human-Robot Symbiosis).
- Kawamura, K., D. M. Wilkes, T. Pack, M. Bishay and J. Barile (1996b). *Humanoids: Future robots for home and factory*. First International Symposium on Humanoid Robots, p. 53-62. Waseda University, Tokyo.
- Kirwan, B. and L. K. Ainsworth, Red. (1993). *A Guide to Task Analysis*. London, Taylor and Francis.
- Leifer, L. (1995). *Integrating the Social-Technical Framework of Human Service through the Internet-World-Wide Web*. Proceedings of the International Workshop on Biorobotics: Human-Robot Symbiosis, p. Tsukuba, Japan, Elsevier Press.
- Maulsby, D., S. Greenberg and R. Mander (1993). *Prototyping an intelligent agent through Wizard of Oz*. Human Factors in computing systems, p. 277 - 284.
- Mayhew, D. (1992). *Principles and Guidelines in Software User Interface Design*. Englewood Cliffs, New Jersey, Prentice Hall.
- Murray, S. A. (1995). *Human-Machine Interaction with Multiple Autonomous Sensors*. 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-machine Systems, p. Cambridge, MA.
- Neveryd, H. (1998). *Ultraljudsnavigerande mobil robot för rörelsehindrade personer*, Lunds tekniska högskola.
- Neveryd, H., H. Eftring and G. Bolmsjö (1999). *The Swedish experience of Rehabilitation Robotics*. Proceedings of Workshop on Rehabilitation Robotics, p. 49 - 54. Hoensbroek, The Netherlands.

- Nielsen, J. (1993). *Usability Engineering*. San Diego, California, Academic Press.
- Nielsen, J. and R. L. Mack, Red. (1994). *Usability Inspection Methods*. New York, John Wiley & Sons.
- Oestreicher, L. (under utg.). *A classification scheme for robots in terms of complexity and capability* Stockholm, IPLab, NADA, Kungliga Tekniska Högskolan.
- Severinson Eklundh, K. (1998). *Användaranpassad utformning av självgående fetch-and-carry-enhet i kontorsmiljö*. Projektbeskrivning, Stockholm, IPLab, NADA, Kungliga Tekniska Högskolan.
- Siegel, J. (2000). *CORBA 3 Fundamentals and Programming*, Wiley Computer Publishing Group.
- Smith, S. L. and J. L. Mosier (1986). *Guidelines for Designing User-Interface Software*, Mitre Corporation.
- Sommerville, I. (1996). *Software Engineering*. Wokingham, England, Addison-Wesley.
- Tåqvist, H. (1999). Fetch-and-Carry Robot for Assistance to Humans - A Navigation System and a Human-Following Behavior. IPLab, NADA. Stockholm, Kungliga Tekniska Högskolan.
- Tognazzini, B. (1992). *TOG on Interface*. Reading, Massachusetts, Addison-Wesley.

13.2 Webbplatser

Talsyntes och taligenkänning: <http://www.ibm.com>

SICStus Prolog: <http://www.sics.se>

Java <http://www.javasoft.com>

OMG <http://www.omg.com/>

14.Figurer och tabeller

14.1 Figurer

Figur 1. En servicehund i arbete (markerat genom västen) som drar av en strumpa på sin matte.....	8
Figur 2. Tre olika uppgifter som alla innehåller en liknande grundstruktur, och som är olika instanser utav ett och samma grundläggande mönster.....	10
Figur 3. Uppsättningen av ”Wizard av Oz”-försöket.	12
Figur 4. Kontrollrummet från vilket roboten styrs under Wizard-of-Oz-testet. För att hantera roboten under testen krävdes två personer, en som styrde roboten och en som hanterade talsyntessystemet.	13
Figur 5. Användare under ett försök.....	13
Figur 6. Besökare som leker med roboten före ett Wizard-of-Oz-försök.....	14
Figur 7. ”Walkie-talkie”-liknande fjärrkontroll till roboten. Nödknapp och tumstyrd joystick används för att detaljstyra roboten. Övrig kontroll sker genom talad input (mikrofon)...	17
Figur 8. CERO – Co-operative Embodied Robot Operator. En enkel mekanisk docka som med ett enkelt rörelsemönster förstärker känslan av interaktion mellan användare och robot.	18
Figur 9. Vy av robot med transportenhet a) framifrån och b) från sidan.	19
Figur 10. Förslag på utformning av stödhjul., med en bakre glidskena, för att kunna ta sig nerför trösklar.	19
Figur 11. Förslag på alternativ framdrivningsmekanism, baserad på fyra drivande hjul.....	20
Figur 12. Systemöversikt för servicerobotprototyp.....	25
Figur 13. Systemets informationsflöden mellan komponenterna	26
Figur 14. Robotplattform samt robotprototypen med påbyggnad.....	30
Figur 15. Sonarernas placering och ljudpulsernas utbredning.....	31
Figur 16. Uppdatering av målvektor utifrån position.....	32
Figur 17. Upptäckta hinder vid färd mellan en väg och en soffa med bord	33
Figur 18. Uppdatering av riktningsvektor med hänsyn till hinder	33
Figur 19. Vanlig triangulering	34
Figur 20. Ultraljudstriangulering.....	34
Figur 21. Landmärkesinsamling.....	35
Figur 22. Landmärken som roboten lokaliserar sig efter.....	35

Figur 23. Huvudgränssnitt 1.2.1 "Mission"	38
Figur 24. a) använda symboler, b) samma rum, c) teleoperation.....	38
Figur 25. a) Autonom robot, b) tele-presence och interaktion med 3:e person	39
Figur 26. Första prototypen.....	41
Figur 27. CERO-inspirerad prototyp.....	43
Figur 28. Senaste versionen av robotens gränssnitt	44
Figur 29. Den tänkta situationen när talgränssnittet används av en rörlig användare "ute på golvet".....	50
Figur 30. När ett uppdrag identifierats av systemet försöker det få användaren att lägga till fler uppgifter tills informationen är tillräcklig för att uppdraget ska kunna utföras. Därefter rapporterar systemet om hur uppdraget lyckas.....	53
Figur 31. Olika gester som kan utföras med hjälp av CERO.....	55
Figur 32. En konceptuell systemskiss. Det streckade området utgör de delar som är specifika för systemet	57
Figur 33. Karta över miljön för robotförsöket; roboten kan åka på det blå golvutrymmet	60
Figur 34. exempel på videoinspelning med användaren, robot och det grafiska gränssnittet ..	64

14.2 Tabeller

Tabell 1 Operationer systemet behöver klara av.....	23
Tabell 2 Objekt som systemet behöver känna till	23
Tabell 3. Likheter och skillnader mellan olika typer av talade gränssnitt.	56