



PROJEKTREDOVISNING/SLURAPPORT

Charlotte Magnusson

Utveckling av en social simulator



Centrum för rehabiliteringsteknisk forskning
Lunds tekniska högskola

Projekt: Utveckling av en social simulator

Projektnr:99/0481

Projektansvarig: Marjatta Oredsson, Föreningen Autism, Skåne

Projektledare: Charlotte Magnusson, Certec, Avd för rehabiliteringsteknik, Inst för Designvetenskaper, LTH

Bakgrund

För personer med autism eller Asperger syndrom är det ofta svårt att läsa av, förstå och hantera sociala situationer [1],[2],[3]. Och eftersom sociala beteenden så ofta är en väl integrerad och snudd på omedveten del av vårt eget beteende så har vi neurotypiska* människor ofta svårt att förstå och hantera social samvaro som innefattar personer med dessa funktionshinder. En person med högfungerande autism/Asperger syndrom kan givetvis *lära sig* sociala regler genom att observera vårt sociala samspel (observera att det brukar vara *de* som lär sig *våra* regler och inte *vi* som lär oss hur *de* vill ha det). Men eftersom den automatik med vilken vi själva lär oss sociala beteenden inte fungerar för en person med autism så tar denna inläring lång tid. Dessutom finns det så mycket underförstådda budskap i sociala situationer att det kan vara svårt att rätt tolka det som sker och därmed lära sig rätt samband och regler.

Givetvis gäller även det omvända. Det sociala tänkande hos neurotypiska personer arbetar ofta på ett omedvetet plan vilket gör det svårt att tolka en del av det som en person med högfungerande autism/Asperger syndrom gör. Sådana tolkningar kräver ofta en medveten ansträngning, och en neurotypisk person hamnar lätt fel när det inbyggda tolkningsmaskineriet trots allt automatiskt går in.

Den enklaste lösningen på detta kan tyckas vara att alla inblandade frågor så fort de inte förstår något, men det är ofta svårt att gå in och fråga om sociala regler/socialt tänkande i en faktisk situation, eftersom man just genom frågandet visar att man inte behärskar det sociala samspelet [4]. Om frågan ställs till fel person kan denne dessutom på mer eller mindre otrevliga sätt utnyttja den svaghet som på så sätt blottas.

Vad gäller den tekniska bakgrunden till projektet, så har vi på Certec, avdelningen för Rehabiliteringsteknik, Institutionen för Designvetenskaper vid Lunds Tekniska Högskola, sedan länge arbetat med metodutveckling, synliggörande av kunskap och kunskapsöverföring [5]-[14].

Syfte

Mot ovanstående bakgrund var därför syftet med projekt *Utveckling av en social simulator* att utveckla ett datorbaserat verktyg där sociala regler och samband synliggörs

* ordet *neurotypisk* myntades av personer med autismspektrumstörningar som en beteckning på människor med genomsnittliga, normalfungerande nervsystem, dvs motsatsen till de som är neurologiskt atypiska, så som *ex* människor med autism.

så att man på egen hand kan gå in och utforska dem. Målet var att inom projektets ram få fram en fungerande prototyp av programvaran.

Genomförande

Det var från början klart att det verktyg som utvecklades skulle använda text, bild och ljud. Från början var detta ett rent tekniskt övervägande – text, bild och ljud är förhållandevis enkelt att hantera, samtidigt som det med hjälp av text, bild och ljud går att synliggöra väldigt mycket.

Under hösten 2000 utvecklades så en första prototyp av programmet. Denna första prototyp byggde på att användaren hela tiden hade 2-3 fasta val att välja mellan. Vid de första utvärderingarna (utförda inom projektgruppen) av prototypen stod det dock klart att 2-3 fasta val kändes ganska begränsande, och att programmet helt enkelt blev lite tråkigt att använda. Samtidigt var det klart att det med denna programstruktur blev möjligt att gå in ganska djupt i den sociala situationen (bild 1). Till prototyp 1 fanns även knutet ett redigeringsprogram i vilket man grafiskt kunde bygga upp datastrukturen (bild 2).

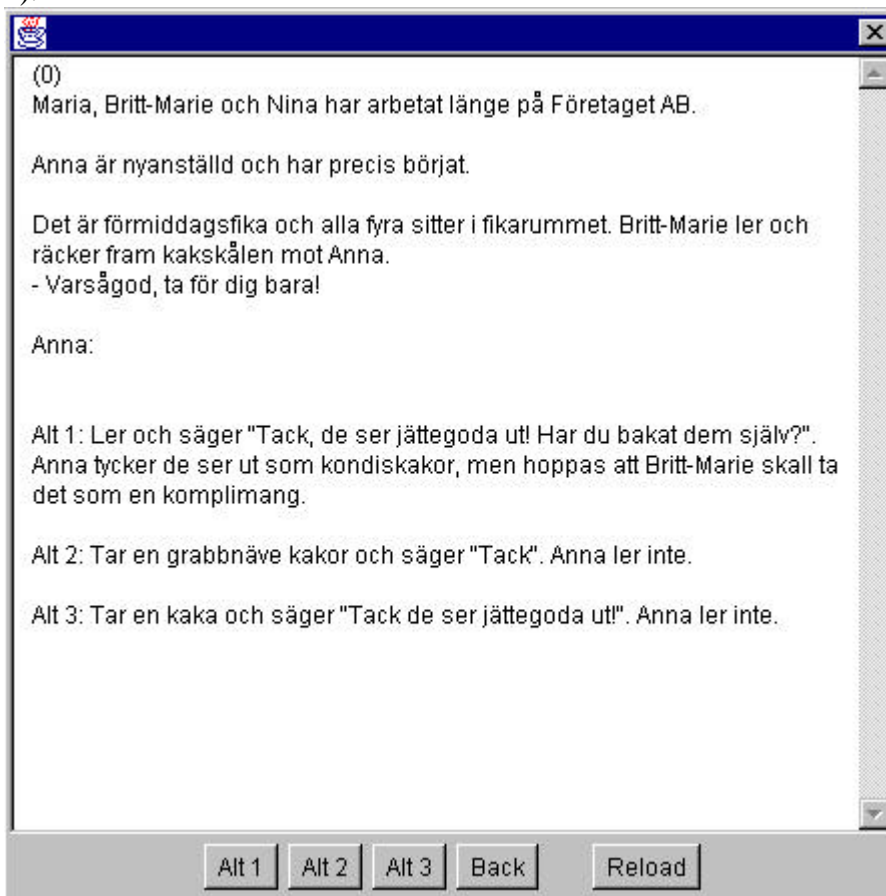


Bild 1. Första testversionen med fasta val.

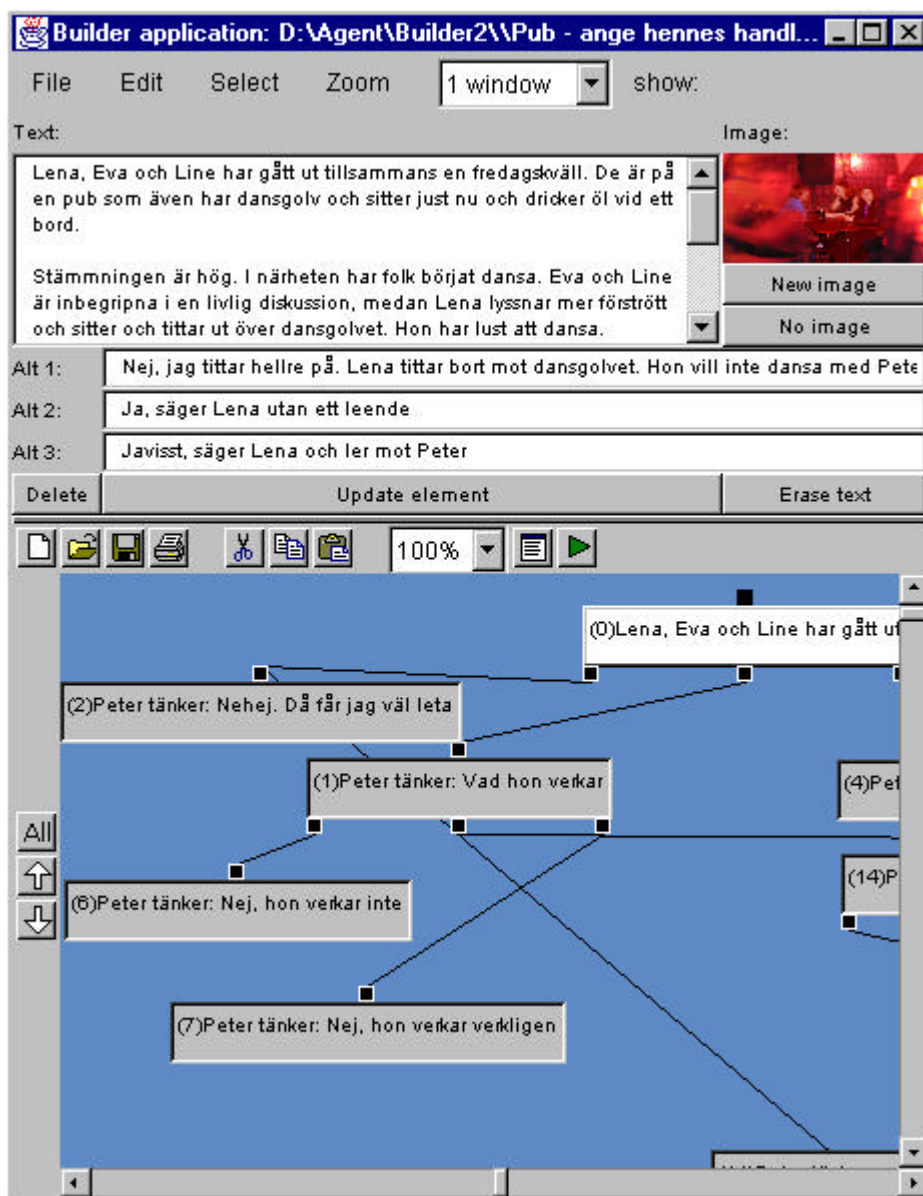


Bild 2. Redigeringsprogrammet till prototyp 1.

Steg två blev sedan att se om det överhuvudtaget var rimligt möjligt att inom projektets ram ta fram ett program som istället byggde på att användaren var i princip fri att själv välja handlingar. En prototyp som innehöll en enkel textigenkänningsmodul (Appendix 1, Enkel textigenkänning) togs fram, och den testades både på neurotypiska personer och på medlemmar i föreningen Autism i Skåne.

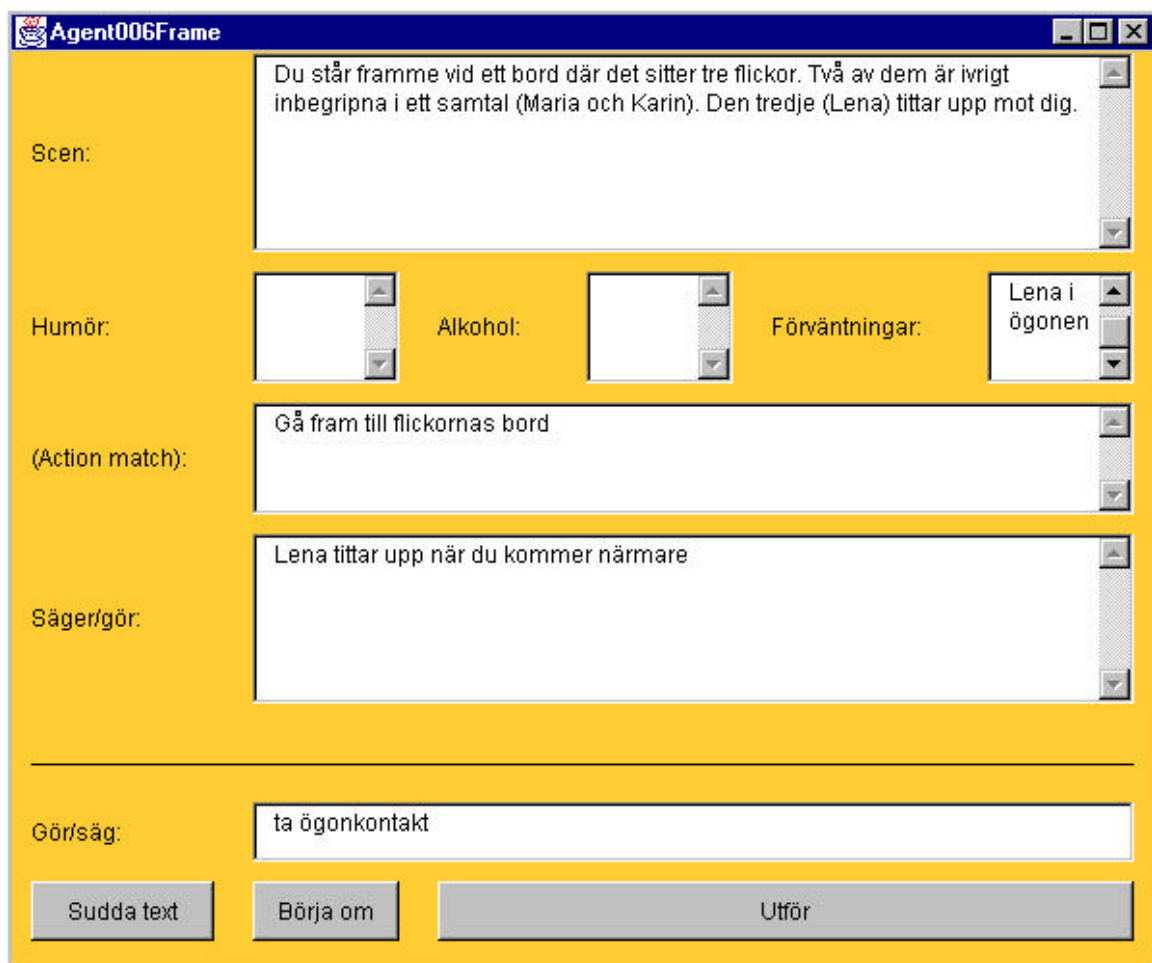


Bild 3. Det textbaserade gränssnittet hos prototyp nummer 2.

Vid testen identifierades ett antal problem:

- Användaren kunde lätt fastna (datorn förstår givetvis inte allt man skriver, och hittar man inte rätt formulering kommer man ingen vart).
- I en visningssituation kan tider (t.ex hur mycket tålamod en person i scenen har) bli för korta.
- Textigenkännigen klarar bara en mening åt gången.
- Man kan inte backa.
- Gränssnittet är svårt att förstå utan hjälp (detta var ett förutsett problem).

De övergripande resultaten av testet var ändå uppmuntrande – trots de uppenbara bristerna i den grafiska utformningen av gränssnittet (bild 3) visade sig programvaran vara både motiverande och så som vi bedömde det tillräckligt fungerande. Intressant att notera var att det tråkiga gränssnittet verkade påverka så lite som det gjorde – programmets responser var vid det här tillfället uppenbarligen mer väsentliga för upplevelsen än utformningen av själva skärmbilden.

Nästa steg blev därför att arbeta vidare med ett program som fungerade enligt de principer som visat sig användbara i prototyp nummer två, utan att för den skull släppa möjligheten att kunna använda fasta val.

Så här långt hade de testscenarier som tagits fram till prototyp 2 kodats in för hand. Det visade sig tidigt i arbetet att om vi ville ha scenarier med någon högre grad av komplexitet så blev detta arbetssätt snabbt omöjligt. Innehållet i ett scenario blir helt enkelt snabbt för komplext för att överblickas. Det grafiska gränssnitt som tagits fram till prototyp 1 såg vid första påseende ut att vara svårt att anpassa till den ganska komplexa datastruktur som krävdes för att ge den sociala simulatören tillräcklig flexibilitet. För prototyp 3 togs därför ett nytt redigeringsprogram fram parallellt.

Den datastruktur som blev resultatet efter utvärderingen av prototyp 2 beskrivs i Appendix 2, Datastruktur. Prototyp 3 byggde sedan på denna datastruktur, och kunde hantera text, bild och ljud. De tidigare "fasta" scenarier som tagits fram gick också att köra via prototyp 3. Skärmbilden i prototyp 3 syns i bild 4.

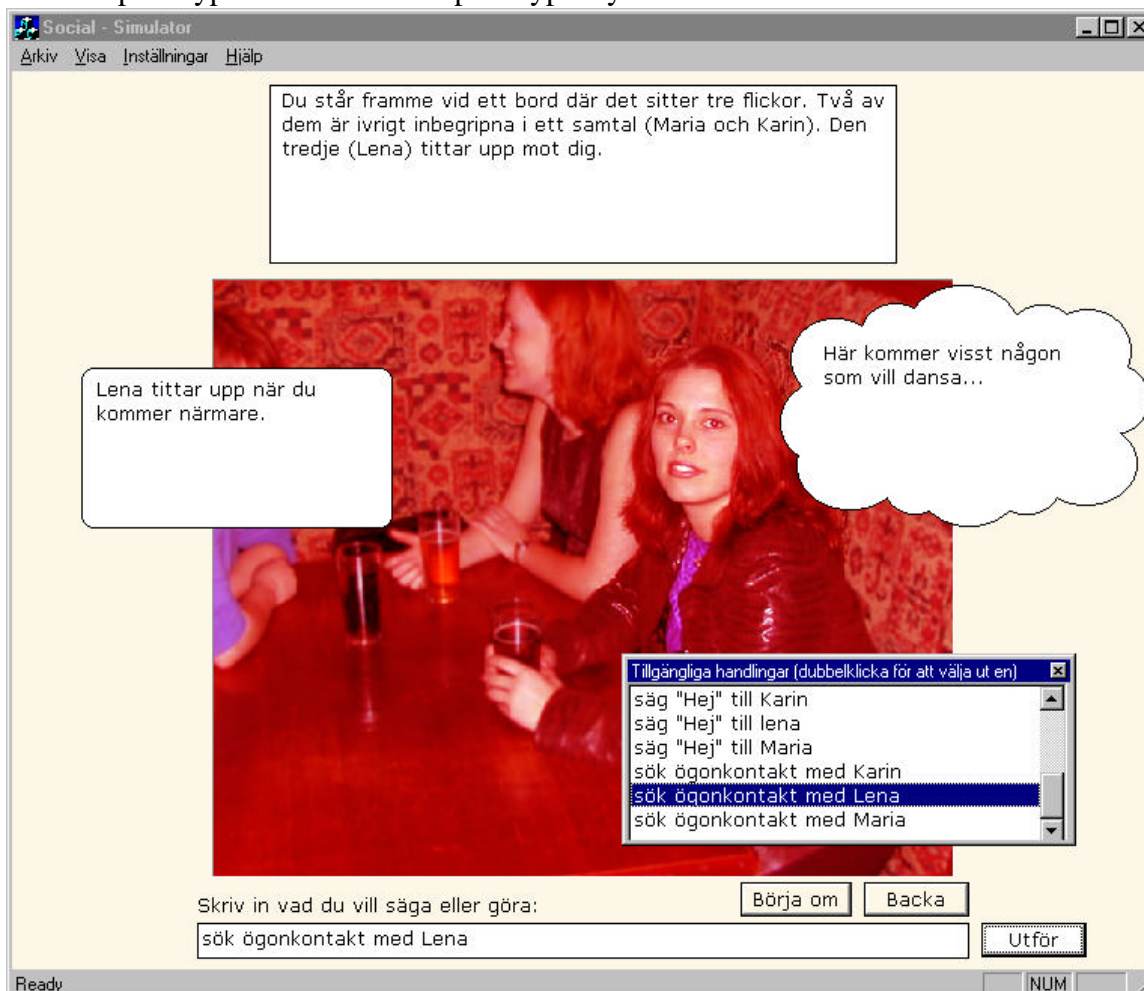


Bild 4. Skärmbild i prototyp 3.

Överst på skärmen syns en text som beskriver situationen. Det finns en bild mitt i fönstret, och det kan finnas ljud som spelas upp. I rutan till vänster kommer olika former av textresponser från programmet/personer i scenariot. I tankebubblan (som användaren själv väljer att plocka fram) syns texter som beskriver vad olika personer tänker. Längst ned finns en rad där användaren själv skriver in vad han/hon vill säga eller göra. För att hjälpa användaren med förslag på handlingar finns det möjlighet att plocka fram en förslagslista (i bild 4 syns den under tankebubblan). Användaren kan hela tiden backa tillbaka och se vad alternativa handlingar ger för utfall. Programmet stödjer enkel animation – det finns möjlighet att som en del av en respons visa upp en bild som försvinner efter ett kort tag. I responsen kan även ingå att ljud spelas upp. Varje scen kan generera tidsstyrda händelser – låta personer tappa tålamodet om man dröjer för länge, eller helt enkelt ge användaren lämpliga tips.

För att komma åt de problem som identifierades vid testen med prototyp nummer 2 har alltså följande åtgärder vidtagits:

- Användaren fastnar – det finns idag en list med handlingsförslag.
- Tider kan vara för korta – det går att stänga av tidtagningen (och ljudet med för den delen).
- Man kan backa.
- Gränssnittet har utvecklats.

Det redigeringsprogram som togs fram parallellt hade följande skärmbild (bild 5).

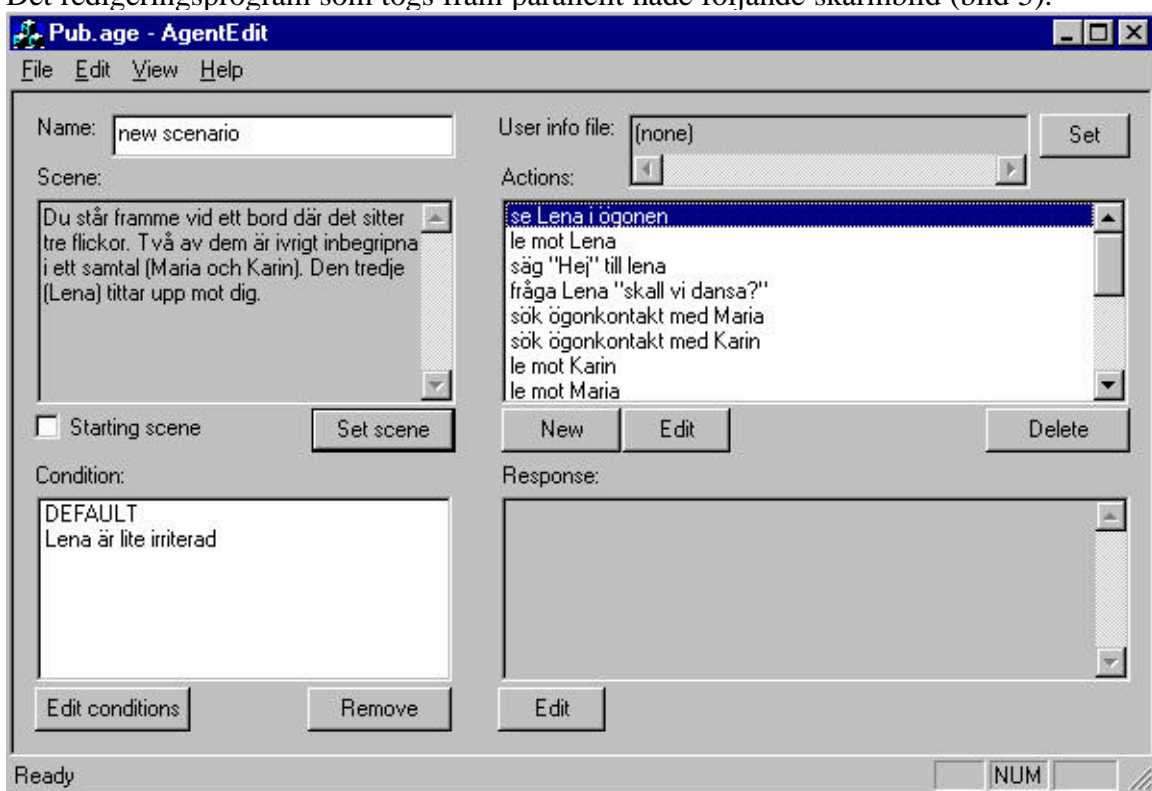


Bild 5. Redigeringsprogrammet till prototyp 3.

För att kunna släppa redigeringsprogrammet var det uppenbart att det skulle komma att behövas en ordentlig hjälpfil. Under utarbetandet av denna blev det allt mer tydligt att programmets utformning var olämplig. Det visade sig också direkt i användarförsök att detta gränssnitt var i princip omöjligt att handskas med för alla utom den som hade full förståelse för datastrukturen. Detta är i sig en intressant observation – att skapandet av en hjälpfil i sig kan innebära en kritisk granskning av ett program som kan vara väl så givande som riktiga test (även om man givetvis inte kan avvara testen).

Ett bättre redigeringsprogram var uppenbart av nöden. Vi beslöt att återgå till ett grafiskt gränssnitt liknande det som utvecklats till prototyp 1 (bild 2) – preliminära försök visade att detta trots allt gick att utforma på ett någorlunda överskådligt sätt. Samtidigt anpassades både detta och den sociala simulatoren till att integrerat kunna hantera också den först provade typen av scenarier – de med ett fåtal fasta val.

Den slutliga prototypen av den sociala simulatoren fick i princip samma utseende som visades i bild 4. Dessutom kan programmet nu hantera fasta val (se bild 6).



Bild 6. En scen med fasta val.

Redigeringsprogrammet har numera följande gränssnitt (bild 7).

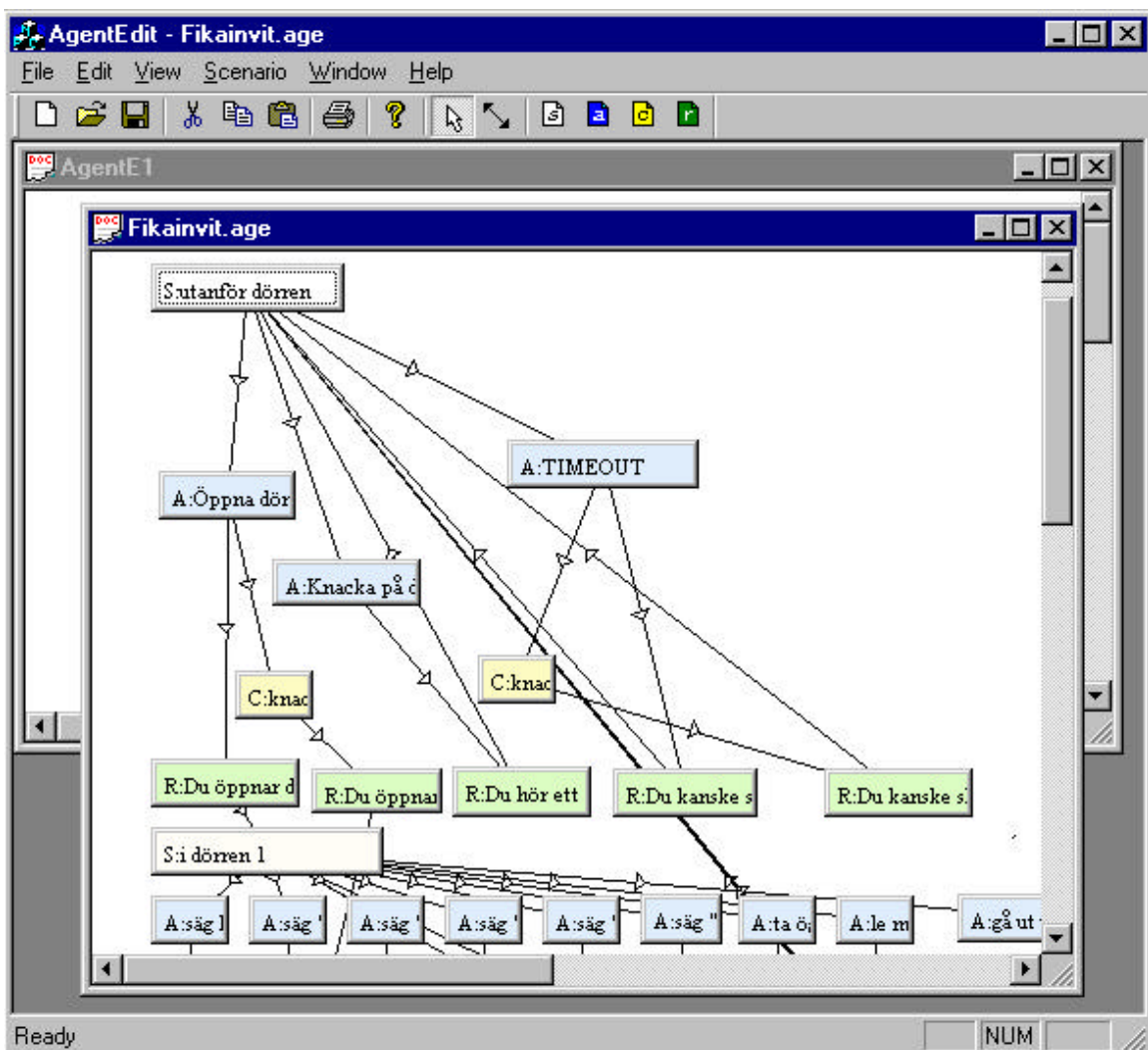


Bild 7. Den slutliga prototypen av redigeringsprogrammet.

Redigeringsprogrammets gränssnitt bygger i denna version på att scener, handlingar, villkor och responser representeras som boxar (boxarnas storlek är anpassningsbar). Flödet i ett scenario anges sedan genom att dessa boxar förbinds med pilar. Initiala användartest visar att detta gränssnitt ger möjlighet åt betydligt fler personer att skapa scenarier. Samtidigt är det ganska klart att uppgiften inte är trivial – ett realistiskt scenario håller förhållandevis hög komplexitetsgrad. För att underlätta för användaren innehåller också denna version av programmet en hjälpfil. I denna hjälpfil finns även övningsexempel som användaren kan gå igenom för att lättare förstå hur programmet fungerar. Redigeringsprogrammet beskrivs närmare i Appendix 3, AgentEdit.

Resultat

Det finns idag en fungerande prototyp av både den sociala simulatorm och ett redigeringsprogram där man skapar scenarier till den sociala simulatorm.

På grund av begränsningarna i projektets resurser (jämfört med ansökan) har flera mer omfattande studier strukits. Samtidigt är det vår uppfattning att resultatet så här långt är lovande. Vid test och visningar har stort intresse för den här typen av verktyg framkommit.

Programvaran går att köra på de flesta Windows-plattformar (95/98/NT/2000) och den kräver ingen extrautrustning. Den finns att hämta gratis på:

<http://www.certec.lth.se/expertsystem/SocialSimulator/index.html>

Det är lite intressant att notera att initiativet uppenbarligen ligger i tiden. Det finns nämligen ett likaledes pågående projekt vid universitetet i Nottingham, "Development of Social Skills amongst adults with Asperger's Syndrome using Virtual Environments" (<http://www.virart.nott.ac.uk/asi/>). Detta initiativ utgår ifrån tredimensionella världar, och utgör ett intressant komplement till den Sociala simulatören, eftersom man kommer åt delvis olika delar av den sociala interaktionen via de olika angreppssätten.

Styrkan hos programvaran som ingår i den sociala simulatören, dess relativa enkelhet (text, bild på jpeg eller bmp format, ljud på wav-format) är givetvis även dess svaghet. Det finns mycket som helt enkelt inte går att representera på denna form. Vi har trots detta valt att hålla fast vid denna approach, och tror att området mår bra av att det finns flera olika kompletterande verktyg.

Långsiktig betydelse

Det är ännu tidigt att uttala sig om projektets långsiktiga betydelse. Det är dock klart att potentialen för användning av ett verktyg för att bearbeta/synliggöra/simulera olika aspekter hos sociala situationer är stor.

Projektet har exempelvis redan väckt intresse inom psykiatrin. Det är för tidigt att uttala sig om exakt hur programmet kommer att användas, men det är klart att alla som är intresserade av problematik kring sociala situationer borde vara betjänta av att det finns den här typen av programvara.

Spridning av resultat

Projektresultaten sprids ännu så länge främst via projektets hemsida <http://www.certec.lth.se/expertsystem/SocialSimulator/index.html>

Fortsatt verksamhet

Projektet har under 2001 erhållit stöd från Vårdalstiftelsen. Det är vår avsikt att gå vidare och söka projektmedel för fortsatt arbete inom projekt social simulator.

Uppföljning och utvärdering

Projektet följs upp kontinuerligt i och med att arbetet kring den sociala simulatorm fortsätter.

Appendix 1, Enkel textigenkänning

Fullständig textigenkänning är oerhört avancerat (i princip kräver det ju att datorn verkligen förstår vad som skrivs) och det finns idag ingen teknisk lösning som fullt ut klarar detta – trots att lång tid och mycket resurser lagts ned på problemområdet. Däremot finns givetvis en mängd lösningar som exempelvis kan hantera text inom en begränsad domän.

En mycket enkel sorts igenkänning bygger på att datorn helt enkelt identifierar nyckelord i meningen. Närvaro av nyckelorden gör att olika responser genereras. Vid starkt begränsade problemdomäner kan det här fungera utmärkt – kan användaren exempelvis i princip bara svara *ja* eller *nej* så fungerar det här förmodligen bra.

Andra sorters igenkänning kan bygga på att datorn istället känner igen typiska mönster i texten. Ett exempel på detta var programmet Eliza. Exempel på den sorts mönster, och responser som genereras i Eliza visas i följande tabell:

Mönster:	Respons:
X always Y	Can you think of a specific example? When? Really, always?
X you Y me	Why do you think I Y you? You like to think I Y you, don't you? Suppose I did Y you, what would that mean?

Detta kan ge upphov till följande dialog:

A: People always stare at me.

B: Can you think of a specific example?

A: For instance you keep staring at me.

B: Why do you think I keep staring at you?

A: Because you hate me.

B: You like to think I hate you, don't you?

Igenkänningen som utvecklades för sociala simulatorm har drag av båda de ovanstående varianterna, samtidigt som den egentligen bygger på igenkänning av hela fraser.

Grundprincipen för igenkänningen är att det för varje handling som kan utföras lagras ett stort antal alternativa sätt att uttrycka samma sak.

För att exempelvis säga "Hej" till en flicka finns bland annat alternativen:

säg hej
säger hej
säga hej
jag säger hej!
jag säger hej till flickan
jag säger "hej!" till flickan
jag säger hej till henne

För den som skapar scenarier är det opraktiskt att behöva skriva in alla fraserna fullt ut – det blir betydligt enklare om man kan dela in fraserna i bitar, och sedan låter man datorn kombinera ihop bitarna och jämföra dem med det som skrivs in av användaren.

Ett möjligt sätt att lagra fraserna ovan är följande indelning:

jag	säg säger säga	”	hej	!	”	till flickan till henne
-----	----------------------	---	-----	---	---	----------------------------

Delarna i detta schema kan sedan kombineras ihop till de önskade meningarna. När programmet körs är grundprincipen att det tar den inskrivna meningens och försöker matcha delar av denna mot de ord som finns lagrade i olika kolumner.

Det finns dock ett problem med detta – om man bara kombinerar ihop alla delarna i tabellen ovan får man även en mängd nonsens (även om det krävs att delarna hamnar i den ordning som specificeras genom tabellens kolumner). Dessutom blir det efterhand svårt att skilja olika handlingar från varandra. Kan man säga "Hej" både till flickan och till pojken t.ex är det ganska uppenbart att det krävs mer än att bara plocka ihop delarna ovan och jämföra – "säg hej" t.ex matchar både att säga hej till flickan och till pojken.

Programmet försöker lösa detta genom att dessutom använda nyckelord. Detta innebär att man i tabellen ovan även anger vilka delar som *måste* finnas med för att den text som skrivs in skall anses stämma med den lagrade handlingen.

I tabellen ovan är *hej* en sådan del. Finns det fler att säga "hej" till kan även den sista kolumnen (till flickan/till henne) behöva markeras som nödvändig.

I de försök som hittills genomförts har denna förhållandevis enkla textigenkänning visat sig fungera fullt tillfredsställande.

Appendix 2, Datastrukturen

Varje scenario i den sociala simulatorm består av ett antal *scener*.

Varje *scen* innehåller:

- handlingar
- villkor

- responser

Dessutom har scenen:

- namn (används bara vid redigeringen)
- text (den text som beskriver scenen för användaren)
- bild
- ljud

Scenen har även något som kallas *timeout*. Detta innebär att efter en tidsrymd som anges för varje scen genereras en så kallad *timeout*-handling. Dvs. det är en handling som inte initieras av användaren utan som genereras i datorn.

Handlingarna i sin tur innehåller dels ett namn (som visas i förslagslistan) och information för textigenkänningen. Saknas textigenkänningsinformation för alla handlingar i en scen visas denna som i bild 6 – dvs det blir en scen med fasta val.

Villkoren anges i form av texter. Dessa texter kan kombineras med logiskt OCH (AND). Detta innebär att det sammansatta villkoret betraktas som uppfyllt enbart när *båda* delvillkoren är uppfyllda. Villkoren styr vilken respons som olika handlingar genererar. Villkoren är inte nödvändiga för scenarioupbyggnaden – däremot förenklar de för den vane scenariobyggaren.

Responserna innehåller:

- responstext (den text som visas i rutan till väster på skärmen)
- tanketext (den text som visas i tankebobblan)
- bild (en bild som visas en kort stund innan scenbyte sker)
- ljud (ett ljud som spelas upp innan scenbyte sker)

Dessutom innehåller varje respons information om vilka villkor som blivit uppfyllda i och med att en respons genererats, samt vilka villkor som skall tas bort som en konsekvens av just denna respons.

Varje respons leder sedan vidare till en ny scen.

Appendix 3, AgentEdit

Det program i vilket man redigerar scenarier till den sociala simulatorm heter AgentEdit.

I detta visas scener, handlingar, villkor och responser som boxar i ett redigeringsfönster (se bild 7 och 8). Flödet i programmet anges genom att boxarna länkas samman med pilar. Vilka handlingar som hör till en scen anges alltså genom att de aktuella handlingarna via pilar sitter ihop med scenen. Vilken respons respektive handling skall leda till (i frånvaro av villkor) anges genom att det går en pil från handlingen till responsen. Finns det villkor som avgör vilken respons som skall genereras går det först en pil från handlingen till villkoren, och sedan från villkoren till responsen. Den scen som responsen leder vidare till anges genom att det går en pil från responsen till scenen. Det

är tillåtet att dra pilarna från responsen tillbaka till samma scen som handlingen hörde till – det kan ju hända att scenen inte ändras vid vissa handlingar.

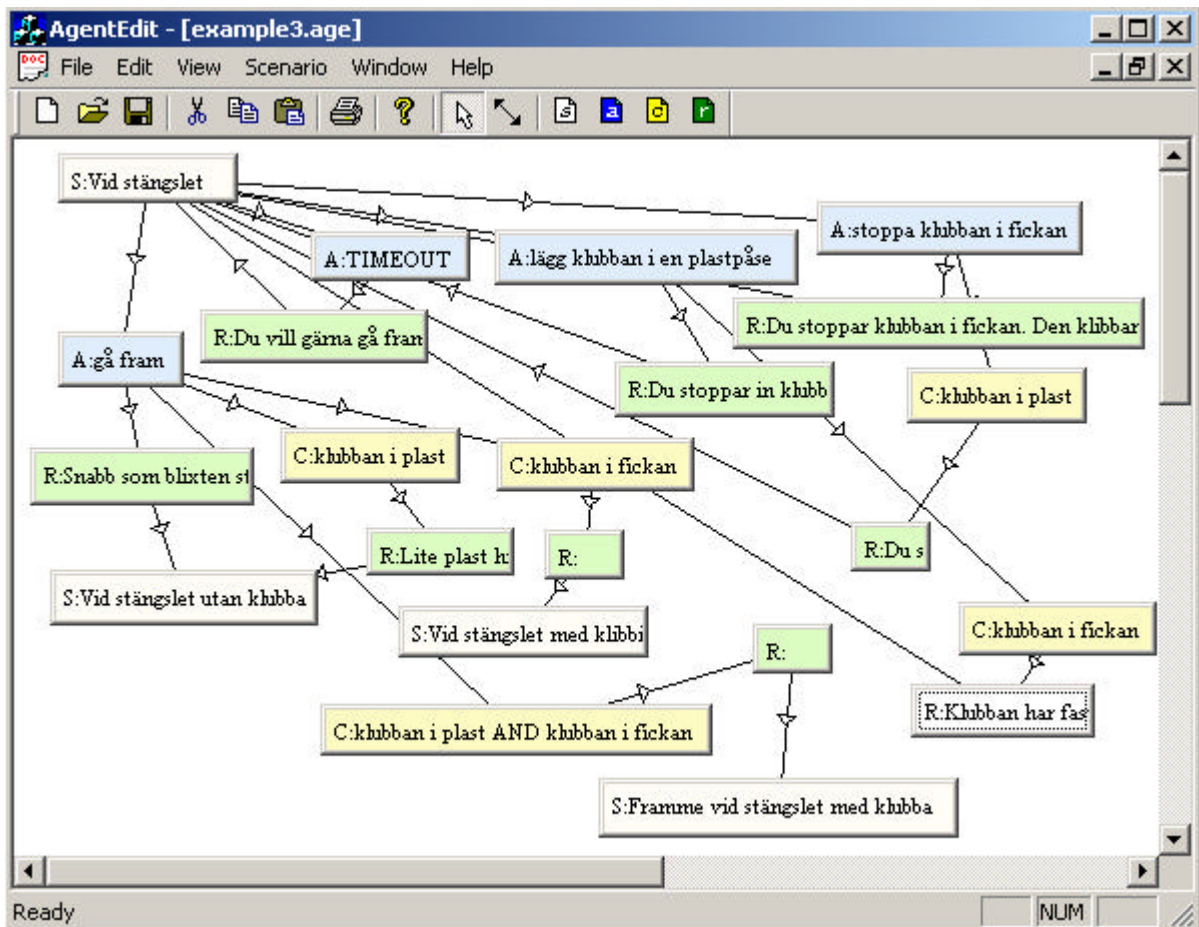


Bild 8, Ett exempelscenario i AgentEdits redigeringsfönster.

För att redigera innehållet i boxarna dubbelklickar man på dem i redigeringsfönstret. Man får då upp dialogrutor i vilka man kan ange olika egenskaper.



Bild 9, En dialogruta där man kan ange egenskaper för en scen.

I bild 9 visas en dialogruta för en scen. Som synes kan man ange namn (det är det namn som syns i boxen i redigeringsfönstret), text, bild, ljud och timeout. Ljudet till en scen kan "loopas" (loop sound). Detta innebär att ljudet spelas om och om igen, vilket kan vara bra för många sorters bakgrundsljud.

På likande sätt finns det dialogrutor för handlingar, villkor och responser.

AgentEdit innehåller även en testmodul där man kan testa sitt scenario. Testmodulen fungerar precis som den sociala simulatormen med ett undantag. Vid utveckling av scenarier är det viktigt att testa mycket så att man täcker in alla handlingar användarna kan tänkas hitta på. För att slippa föra anteckningar vid sådana test finns möjlighet att låta programmet automatiskt föra logg över vad användaren gör. För att ingen skall bli loggad utan att veta om det frågar programmet alltid om det skall logga eller ej:

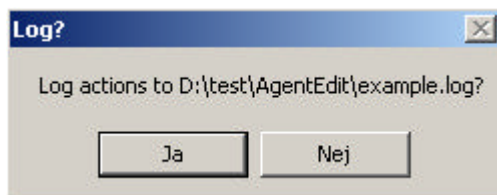


Bild 10. Förfrågan om användarens handlingar skall loggas.

För varje scenario måste man även ange vilken scen som är den scen som inleder scenariot. Och man bör skriva en kort beskrivning av sitt scenario. Denna beskrivning

visas alltid innan ett scenario sätts igång, och där är det bra att ge användaren lite bakgrundsinformation om vad just det aktuella scenariot handlar om.

Referenser

- [1] Does the autistic child have a theory of mind ?, Baron-Cohen,S., Leslie, A M. & Frith, U., Cognition 1985:21, 37- 46
- [2] Autism - gåtans förklaring, Frith,Uta, Liber utbildning 1994
- [3] Stuart Powell and Rita Jordan, Autism and Learning, David Fulton Publishers, 1997
- [4] En riktig människa, Gunilla Gerland, bokförlaget CURA
- [5] Kunskapsbaserat systemet Svarne, beslutsstöd vid våldssituationer, Charlotte Magnusson, Arne Svensk, CERTEC 4:93
- [6] Fria Freja - Del 1. Före datoriseringen, Eve Mandre, Bodil Jönsson, CERTEC 2:96
- [7] Autismen finns där - men det ska gå bra ändå! Del 1. Före Kennies datorisering, Karin Axeheim, Manuela Gustafsson, Anders Gustafsson, Eija Lindqvist, Bodil Jönsson, CERTEC 4:96
- [8] Fria Freja - Del 2 Eve Mandre, Bodil Jönsson, CERTEC 5:96
- [9] Fria Freja - Del 3 Eve Mandre, CERTEC 1:97
- [10] Kunskap på burk - SVARNE, slutrapport, Charlotte Magnusson, Arne Svensk, CERTEC 3:97
- [11] Expertsystem och lärande, Charlotte Magnusson, Certec 4:98, <http://www.certec.lth.se/dok/expertsystemoch/>
- [12] MatteExperten, ett pedagogiskt-tekniskt samarbetsprojekt, Certec 1:2000 , Charlotte Magnusson, Elsa Foisack, <http://www.certec.lth.se/dok/matteexperten/>
- [13] Rehabiliteringsteknologi och design – dess teori och metoder, rapport, Bodil Jönsson, Peter Anderberg, <http://www.certec.lth.se/teoriochmetod>.
- [14] Useworthiness of Robots for People with Physical Disabilities, Håkan Efring, Doktorsavhandling, (disputation 22 september, 1999).