

Kommunikation och Tjänster i Öppna Nät

Slutrapport 1996-1999

*Kristina Höök, Sverker Janson, Annika Waern,
Markus Bylund, Joakim Eriksson, Fredrik Espinoza,
Niclas Finne, Tomas Olsson, Andreas Rasmusson,
Kent Saxin-Hammarström och Mark Tierney*

Innehållsförteckning

KOMMUNIKATION OCH TJÄNSTER I ÖPPNA NÄT	1
SLUTRAPPORT 1996-1999	1
INNEHÅLLSFÖRTECKNING	2
INTRODUKTION	4
Perioden 1996-1999	5
GRÄNSSNITTSAGENTER	7
Agentmetaforen som en gränssnittsmetafor	7
Forskningsfrågor	8
Förteckning över bilagor inom området 'Gränssnittsagenter'	10
DESIGN OCH UTVÄRDERING AV ADAPTIVA GRÄNSSNITT	11
Gränssnittskultur	11
Rätt design?	12
Möjligheter för alla?	12
Förteckning över bilagor inom området 'Design och utvärdering av adaptiva gränssnitt'	13
GRÄNSSNITT FÖR HANDIKAPPADE	14
Förteckning över bilagor inom området 'Gränssnitt för handikappade'	Error! Bookmark not defined.
SAMHÄLLSINFORMATION OCH MEDBORGARDIALOG	17
SICS bidrag till KIMSAC	17
Erfarenheter från KIMSAC	19
Förteckning över bilagor inom området 'Samhällsinformation och Medborgardialog'	20
AGENTBASERAD INTERAKTION I ÖPPNA TJÄNSTEMILJÖER	21
Framtidens nätverk	21
Användaren i centrum	21

En öppen tjänstemiljö	21
Användningssituationer	24
Kontraktmodeller för tjänstesamverkan	30
Resultat	31
Förteckning över bilagor inom området 'Gränssnitt emot öppna tjänstemiljöer'	34
EN ÖPPEN ELEKTRONISK MARKNADSPLATS	35
Sammanfattning	35
Elektronisk handel	35
Byggstenar i en agent-baserad marknadsinfrastruktur	36
En prototyp till en agent-baserad marknadsinfrastruktur	38
Slutsatser och framtida inriktning	40
Förteckning över bilagor inom området 'Öppen elektronisk marknadsplats'	41
SÄKER INTERNETHANDEL MED AUTONOMA AGENTER	42
Social Kontroll för Säker Internethandel	44
En personlig säkerhetsassistent för säker internethandel	47
Sammanfattning	49
Förteckning över bilagor inom området 'Personlig säkerhet'	50
PERSONLIG KOMMUNIKATION	51
En vision om ett virtuellt bibliotek: Ett "Virtual Community Library"	52
En arkitektur för ett "Virtual Community Library"	53
En prototyp av den personliga biblioteksagenten	55
En simulator av ett decentraliserat rekommendationssystem	55
Forskningsfrågor	56
Förteckning över bilagor inom området 'Personlig kommunikation'	57
DEN KOMMANDE PERIODEN (1999-2002)	58
REFERENSER	59
Links	62
FÖRTECKNING ÖVER BILAGOR	64

Introduktion

Den här slutrapporten bygger på erfarenheter och resultat från det KFB-finansierade projektet "kommunikation och tjänster i öppna nät", i fortsättningen kallat IntAge (en förkortning av begreppet 'Intelligent Agents'). När projektet påbörjades 1996 var Sverige på väg in i IT-samhället. Sedan dess har vårt samhälle definitivt förändrats. För Sverige uppskattades 25-30 procent av befolkningen i åldrarna 16-79 år regelbundet använda Internet i februari 1997¹, det vill säga ungefär 2 miljoner personer. Ökningstakten är hög. Andelen användare som har köpt en vara eller tjänst över Internet uppskattas till ungefär 25 procent i Sverige. Detta är högt i jämförelse med övriga världen - globalt ligger siffran på knappt 20 procent, och då drar USA upp snittet väsentligt. I Sverige har vi dessutom ett ovanligt högt antal användare av olika mobila enheter (mobiltelefoner, bärbara kalendrar, bärbara datorer).

Internet ger användare helt nya möjligheter. Vi kan jämföra priserna på varor, utnyttja tjänster från hela världen och kommunicera med andra människor och grupper som vi annars inte hade kunnat nå. De bärbara prylarna möjliggör dessutom att vi kan göra det från många olika platser – inte endast när vi sitter framför en dator vid ett skrivbord. Intelligenten och funktionaliteten ligger numera i nätet och dess mjukvara snarare än i de olika prylarna vi använder. Men vem är det som styr den här utvecklingen? Är den driven av teknikutvecklingen? Eller är det den information och de tjänster som företagen tillhandahåller som formar framtiden? Hur kan den vanliga användaren påverka och utnyttja utvecklingen?

Vi tror att det är de begrepp vi använder, metaforer och beskrivningar av de tekniska möjligheterna, som mer än något annat styr utvecklingen. I den utveckling vi ser idag ser vi spåren efter de ideer och metaforer som först dök upp i forskningslaboratorier runt om i världen. Det gäller till exempel de direkt-manipulerande gränssnitten, numera standard, som först utvecklades i forskningslaboratorier på Xerox. Utvecklingen idag mot små bärbara prylar som kan kommunicera med varandra, formades av visioner från Negroponte och Key i början av 1990-talet. Deras visioner var inte i första hand teknologiska, utan handlade om användning och funktion. Detta visar att ett nytt synsätt öppnar nya möjligheter, som i sin tur genererar ny funktionalitet.

I det här projektet har vi valt en sådan metafor, ett synsätt, på informationsteknologin, nämligen *agent*-perspektivet. Med ett sådant perspektiv kan man se användare, företag, organisationer, dataprogram, och data-processer som agenter. Dessa agenter utnyttjar alla en gemensam, global, infrastruktur, i vilken de kan samarbeta eller konkurrera. Ur användarens synvinkel blir det inte längre *en* dator, *en* maskin, som vi kommunicerar med, utan istället interagerar vi med andra människor eller grupper av människor, antingen direkt eller indirekt genom deras agenter, den information och de tjänster som dessa erbjuder.

Informationshantering blir med detta synsätt inte ett anonymt förmedlande av objektiv information eller tjänster, utan ett socialt nätverk av subjektiv information som placerats där av olika agenter. Att navigera genom en sådan informationsrymd blir en social process där vi väljer vad vi ska läsa, vilka varor vi ska köpa, etc., beroende på vem som placerat tjänsten där och vilka andra som använder den.

Agent-perspektivet ger inte bara användarna ett nytt sätt att se på system, det ger också systemutvecklarna ett nytt sätt att se på mjukvaruutveckling. Genom att se olika aktörer som agenter, möjliggör vi ett generaliserat synsätt på information och tjänster som deltagare i *elektroniska marknader*. Även om det skett mycket forskning på distribuerade system är det först helt nyligen som man insett att de olika systemen faktiskt representerar olika individer och organisationer. Så istället för att endast se det som att alla delarna i det distribuerade systemet snällt samarbetar för det allmännas bästa, ser vi nu mer komplexa situationer där vi

¹ Christer Sturmark, "Utkast till ett svenskt ramverk för elektronisk affärsverksamhet", rapport för IT-kommissionen.

måste bygga in mekanismer för säkerställa ett förtroende för att delarna kommer att göra det de ska.

Att ha ett agent-perspektiv kan också ses som en vision om en högre grad av demokrati. Mjukvaru-agenter kan löpa runt i de globala näten och utvärdera tusentals erbjudanden om t.ex. försäljning av en bil och hundratals rekommendationer från diverse olika källor. Det betyder att både små och stora distributörer får lika chanser. Dessutom blir distinktionen mellan konsument och producent suddig: vem som helst kan erbjuda en tjänst, och relationen mellan aktörer blir en av ömsesidiga tjänster snarare än en producent/konsument relation. Vårt yttersta mål med projektet är alltså inte i första hand att studera effekterna av informationsteknologin, utan att utveckla verktyg och metaforer som kan påverka utvecklingen i en mer demokratisk riktning.

Perioden 1996-1999

Vårt projekt hade som mål att utforska och utveckla det som kallas agentsystem. Genom att samla kompetensen i HUMLE² och ISL³ laboratorierna på SICS kunde vi täcka in flera aspekter av agentsystem – alltifrån användaraspekter på gränssnittsagenter till multiagentsystem där agenterna i första hand kommunicerar med varandra.

När vi startade 1996 fanns det i princip inga kommersiella agent-system i drift – varken gränssnittsagenter eller multiagent system. Sedan har det kommit flera exempel på bådadera. Den mest kända gränssnittsagenten är Microsoft Assistant som ger hjälp i Microsoft Office. Den kom ut med Office 97. Ett flertal andra gränssnittsagenter och varianter på temat finns på nätet, t.ex. Jennifer James⁴, Catz and Dogz⁵, PPP Persona⁶.

På multi-agent sidan har ett flertal system skapats och tagits i drift. I synnerhet har agenter för elektronisk handel fått kommersiell användning, exempelvis Goodguy, BargainFinder och liknande system. Dessa börjar nu också bli kända hos den bredare allmänheten. Den 22 juni hade Aftonbladet en bilaga om handel på nätet där agentsystem omnämns⁷. Gemensamt för de flesta sådana system är dock att de egentligen inte är oberoende system som tillhör användaren, utan de är tjänster som erbjuds av företag som av olika skäl vill knyta upp användarna till deras system.

Den utveckling vi ser fanns alltså i någon mening förutspådd i vår ansökan. Det är dock fortfarande förhållandevis enkla system som gjorts tillgängliga hittills och långtifrån alla problem är lösta. Vi har inte heller ännu sett konsekvenserna av den här utvecklingen för samhällsutveckling, skydd för den individuella konsumenten, påverkan på användarna, etc.

Resultat

Under de tre år projektet pågått har vi attackerat en mängd problem relaterade till olika aspekter av agent-teknologier och konsekvenser av desamma. Vi har lyckats skapa ett större team med forskare än de som är direkt finansierade av KFB genom att dra till oss industri-pengar, EU-pengar, NUTEK-pengar, etc. Sammanlagt har c:a 13 forskare från ISL och c:a 11

² HUMLE: HUMAN-computer interaction and Language Engineering laboratory.

³ ISL: Intelligent Systems Laboratory

⁴ <http://www.extempo.com/>

⁵ <http://www.petz.com/>

⁶ <http://persona.dfki.de:8080/>

⁷ <http://www.aftonbladet.se/it/9906/21/fynd2.html>

forskare från HUMLE varit inblandade under de tre åren, därav hör 9⁸ doktorander och 3⁹ seniora forskare till den stabila kärnan i detta team.

Arbetet har centrerats kring ett antal tekniska, användnings- och konsekvensorienterade frågor:

- 1) Hur kommer gränssnittsagenter att uppfattas av och påverka användare?
- 2) Hur ska gränssnittsagenter presentera sig för att bli tillgängliga för alla?
- 3) Hur kan vi tillåta att nya agenter kommer till/försvinner utan att det påverkar systemet i sin helhet eller gränssnittet?
- 4) Hur kan en användare "äga" informationen om sig själv?
- 5) Hur kan vi garantera säkerheten i ett multiagent system?
- 6) Hur kan man använda multiagentsystem för att förverkliga elektroniska marknader?
- 7) Hur ska agenterna förhandla med varandra?

Frågeställningarna och de resultat vi nått fram till beskrivs i mer detalj nedan. Vi har valt att bifoga de flesta av de rapporter vi producerat i ett appendix, eftersom vi inte kan redovisa alla insikter och resultat i den här sammanfattningen. Sammantaget bör detta ge en god bild av (vad vi tycker är) ett lyckat projekt.

Resten av sammanfattningen är uppdelas i följande områden:

- Gränssnittsagenter
- Design och utvärdering av adaptiva gränssnitt
- Gränssnitt för handikappade
- Samhällsinformation och medborgardialog
- Agentbaserade Gränssnitt emot Öppna Tjänstarkitekturer
- En Öppen Elektronisk Marknadsplats
- Säker internet handel med autonoma agenter
- Personlig Kommunikation
- Den kommande perioden 1999-2002

⁸ Markus Bylund, Joakim Eriksson, Fredrik Espinoza, Niclas Finne, Tomas Olsson, Andreas Rasmusson, Lars Rasmusson, Kent Saxin Hammarström, Mark Tierney

⁹ Kristina Höök, Sverker Janson, Annika Waern

Gränssnittsagenter

Annika Waern och Kristina Höök

Människor är i grunden sociala varelser. Vi tycker om att umgås, att titta på andra människor, att tillskriva andra människor, djur och saker mänskliga egenskaper och intentioner. Man kan ju fråga sig vad som händer när vi, dessa sociala, interaktions-intensiva, emotionella mänskliga varelser sätter oss framför en dator? Jo, naturligtvis reagerar vi på datorn som om den är något slags varelse. Vi vet ju visserligen att den bara är en dum maskin, men icke desto mindre relaterar vi till den, ger den mänskliga känslor, tror att den har intentioner etc. [Reeves och Nass, 1996]. Hittills har detta snarast setts som ett problem: ett bra gränssnitt har varit ett som ger ett tydligt intryck av att vara verktyg snarare än varelse. Men frågan är om det inte är dags att vända på problemställningen. Är det verkligen enbart negativt om vi uppfattar ett datorsystem som en varelse, eller är det något som man kan dra nytta av i utformningen av datorgränssnitt? En av de saker som vi ägnat oss åt inom IntAge-projektet är att utforska hur varelseliknande gränssnitt kan användas, och vad som händer med oss mänskliga användare när vi sätts framför denna typ av system.

Agentmetaforen som en gränssnittsmetafor

När datorgränssnitt visar upp varelser snarare än saker, så brukar dessa kallas för gränssnittsagenter. Vid en första anblick är skillnaden mellan verktygsgränssnitt och agentgränssnitt kanske inte så stor: den handlar ju bara om vilka bilder som används för att representera funktionerna i ett program. Men vårt sätt att relatera till gränssnittsagenter skiljer sig från vårt sätt att relatera till andra gränssnittsobjekt. Det gör att de kan ge naturliga metaforer för nya funktioner.

Först och främst så kan man föra en dialog med en varelse. Det gör man inte med 'saker'. Många av de saker man vill göra med en dator kräver språklig interaktion. I direktmanipulerade gränssnitt går det bara att göra saker 'här' och 'nu' - man kan klicka på objekt som syns på skärmen, och 'hantera dem', utföra aktioner på dem som har en omedelbar effekt. Ska man skapa ett nytt objekt, eller ge en order som ska utföras om en stund, måste direktmanipulationen kompletteras med språk. En annan sak som är svårt att göra med direktmanipulation är urval - sökning är ett bra exempel. Alla datorgränssnitt innehåller därför språkkomponenter, oftast i form av menyer och dialogrutor.

Ett system som innehåller mycket språklig interaktion uppfattas inte som ett rent verktyg. Det gäller i synnerhet om dialogen sträcker sig över flera interaktioner, eller om språket är fritt. Det här kan vara ett skäl till att 'normala' gränssnitt undviker långa dialoger och flexibelt språk, till priset av en begränsad och oflexibel interaktion. I stället för en sådan modell kan man i stället välja att komplettera direktmanipulation med en agent som kan förstå (ett nästan) naturligt språk, och som används för att utföra uppdrag åt användaren. Ett tidigare projekt [Bretan 1995] undersökte denna sätt att använda gränssnittsagenter. Ett annat exempel på en gränssnittsagent för interaktion i naturligt språk har utvecklats i ett projekt på Harvard [Rich och Sidner 1997].

De flesta system som använder naturligt språk i sin interaktion gör det alltså oftast för att åstadkomma delegering: användaren delegerar ett uppdrag åt agenten, som den får sköta själv. Det finns flera andra sätt att delegera uppdrag till agenter än att prata med dem. Ett exempel är agenter som tittar på när du sköter en viss uppgift, t.ex. sorterar din epost, för att så småningom ta över uppgiften och sköta den hel- eller halvautomatiskt [Shardanand och Maes 1995]. Delegering är en typisk funktion för gränssnittsagenter: det är svårt att hitta någon bra verktygsmetafor alls för delegering. (Vi delegerar ju t.ex. inte till en postlåda att vidarebefordra vår post, vi delegerar uppgiften till postverket.) En av de saker vi kan delegera till en agent är att representera oss gentemot andra användare eller organisationer. En agent kan till exempel hålla reda på en personlig profil om dig, och distribuera valda delar av den

informationen till andra agenter när den får. Den kan också visa en 'avatar' i ett fleranvändarmiljö, en ikon som på något sätt föreställer dig, i syftet att visa för andra användare att du också 'är där' och går att kontakta.

Den intressantaste användningen av gränssnittsagenter är kanske att de gör det möjligt att ha en känslomässig interaktion med datorn. Dels relaterar vi känslomässigt till agenter, men det är också naturligt att agenter själva reagerar känslomässigt. Det är svårt att tänka sig att ett 'vanligt' program, till exempel en ordbehandlare, skulle reagera på om du som använder det är glad, stressad eller eller just gjort slut med din flickvän. Men av en gränssnittsagent förväntar vi oss snarare sådana reaktioner - det är naturligt för en agent både att visa känslor och att reagera på känslomässiga yttringar från användaren. Ibland är just känsloupplevelsen själva vitsen med en tillämpning. Spel, chat och mycket av allt 'nätsurfande' har som sitt huvudsakliga mål att ge underhållning. I spel- och lärande-program används agenter flitigt för att göra programmen mer underhållande. Det finns också elektroniska djur (t.ex. tamagotchin) där hela funktionen är just att likna varelser.

Gränssnittsagenter blir av dessa skäl allt vanligare i underhållnings- och reklambranchen. En fördel med en agent framför ett traditionellt gränssnitt är att en agent på ett enkelt sätt kan representera sin 'ägare'. Ett exempel på detta är bilförsäljerskan 'Jennifer James'¹⁰. Jennifer gör det mycket tydligt att hon är där för att sälja bilar, och gör det på ungefär samma sätt som en mänsklig (kvinnlig) bilförsäljare skulle göra.

Forskningsfrågor

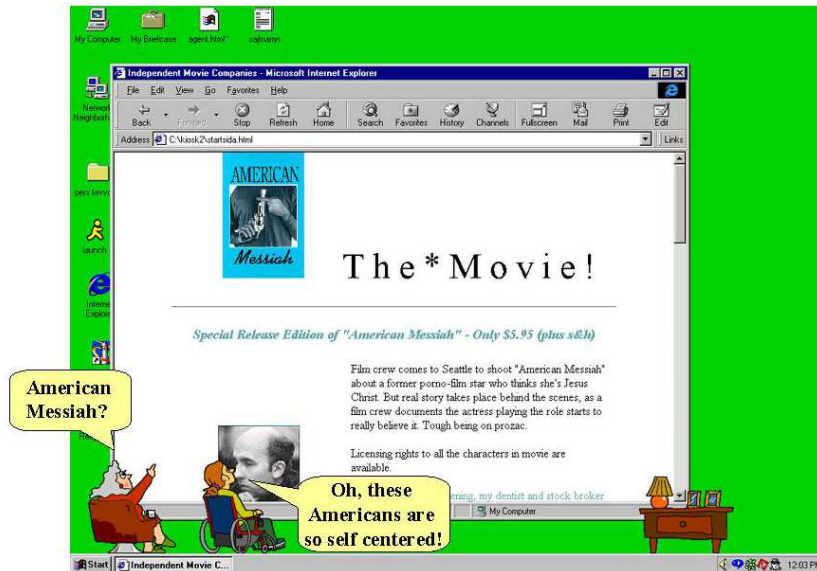
Som forskningsobjekt är gränssnittsagenter fortfarande ganska outredda. Den forskning som har bedrivits har framför allt handlat om att göra agenter så människolika som möjligt, vare sig detta gäller deras språkförmåga, deras förmåga till resonemang, eller helt enkelt deras utseende. Men ur designsynvinkel vet vi mindre. Vi ställde oss tre frågor:

- Hur tekniskt avancerade måste gränssnittsagenter vara - vad måste de kunna göra ?
- Vad bidrar de med - när fungerar gränssnittsagenter och när fungerar de inte?
- Hur påverkar de oss - får de oss att använda ett system mera, kan de påverka oss att köpa en sak, etc?

Vad gäller den första frågan så vet vi att om agenten ges ett mänskligt utseende så ökar användarnas förväntningar på intelligens i systemet [Reeves and Nass 1996, Koda and Maes 1996, Sproull et al 1996, King and Ohaya 1996, Höök et al. 1999]. Ju mer antropomorft (människolikt) gränssnittet är desto mer "naturligt" kommer användarna att reagera på det. Detta kan vara både fördelaktigt och till nackdel beroende på om förväntningar kan infrias av systemet eller ej. Däremot behöver antagligen agentens resonemangsförmåga antagligen inte vara lik människors. Vår förmåga till empati gör att vi 'läser in' egenskaper i en agent - vi lurar tro att den är mer mänsklig än den är.

Den andra frågan kan inte besvaras på något enkelt sätt. Delvis beror det på vilka egenskaper man vill uppnå. Det kan till exempel vara så att det tar längre tid att göra något med en agent, men att det är roligare. Det kan också vara så att man lär sig mer med agenten än utan. Dels kan det bero på vilken attityd användaren har till datorer: en användare som vill se datorer som rena verktyg kan uppleva gränssnittsagenter irriterande.

¹⁰ <http://www.extempo.com/>



Figur 1. Agneta och Frida kommenterar vävsidor.

På SICS har vi jobbat med de två första frågorna genom att utveckla ett exempel på gränssnittsagenter, 'Agneta och Frida' [Persson 1999] (se figur 1). (Den här demonstratorn utvecklades delvis inom IntAge-projektet, samt inom EU-projektet Persona.) 'Agneta och Frida' är två damer som sitter på skärmen och tittar på när man surfar på WWW, ungefär som om de satt framför sin TV i vardagsrummet. De ger roliga och cyniska kommentarer till vad de ser, och de kan också berätta en egen historia, ungefär som en såpopera. 'Agneta och Frida' använder ingen inbyggd intelligens alls. De kan inte förstå naturligt språk, de resonerar vare sig logiskt eller känslomässigt, och de har ingen modell av användarens känslor. Deras dialog är skriven för att passa ihop med vissa webbplatser. Däremot är de mycket skickligt utformade för att få oss att relatera till dem empatiskt. En studie av Agneta och Frida [Höök et al 1999] visade också att folk uppfattade dem som riktiga personer. Användarna kunde efteråt beskriva deras situation, karaktärer och inbördes relation (exempelvis: 'det är en mamma och hennes dotter som sitter och tittar på TV').

Hur uppfattade då folk 'Agneta och Frida'? En oväntad effekt var att man reagerade mycket starkt på dem - antingen positivt eller negativt. Vissa personer tyckte de var roliga, medan andra bara blev irriterade. Det fanns ett samband mellan datorvana och upplevelse: personer med stor vana vid datorer tenderade tycka sämre om 'Agneta och Frida' än personer med lite datorvana. Det här visar tydligt att frågan om när gränssnittsagenter passar är en fråga om kultur och vana. De personer som hade stor vana av datorer hade också vant sig vid att uppfatta webbsurfning som en ganska humorlös och instrumentell uppgift. I den bilden passade 'Agneta och Frida' inte in!

Den tredje frågan, hur gränssnittsagenter påverkar oss, har vi inte arbetat med ännu. Det verkar uppenbart att de borde göra det. Det är ju till exempel så, att vi inte gärna åter på en tom restaurang eller handlar i en tom butik. Samma fenomen tycks uppträda i elektroniska miljöer: i spel används ibland agenter bara för att få en miljö att verka 'bebodd'.

Den allra känsligaste frågeställningen är hur det påverkar oss, att gränssnittsagenter kan 'lura' oss att behandla dem som levande varelser. Vi vet att det är ganska lätt att åstadkomma den effekten. Reeves och Nass [1996] har visat att vi reagerar på ganska enkla signaler - reaktionen är inbyggd i oss på ett fundamentalt och omedvetet plan. Dessutom visar erfarenheterna från leksaker som tamagotchin att det finns en stor marknad för 'datorvarelser'. Men vi vet fortfarande väldigt lite om hur det påverkar vår relation till den information som 'datorvarelserna' förmedlar. Kan de till exempel få oss att ta till oss mer av en reklamtext?

Kan de få oss att köpa en vara som vi annars inte skulle ha köpt? Påverkar de barn mer än vuxna? Alla de här frågorna är fortfarande obesvarade.

Hela idén med att använda gränssnittsagenter väcker mycket starka känslor. Vill vi verkligen att användare ska sitta fångade framför ett datorprogram som spelar på våra känslor, ger sin subjektiva syn på världen, etc.? Som forskare väcker naturligtvis sådana kontroversiella frågeställningar vår nyfikenhet. I vårt nuvarande arbete försöker vi tänja gränserna för vad som traditionellt ses som ett 'bra' eller 'dåligt' gränssnitt, för att i stället förstå de här fenomenen på en djupare nivå [Höök et al. 1999, Persson 1999].

Förteckning över bilagor inom området 'Gränssnittsagenter'

- [1] Höök, Kristina, Sjölander, Marie, Ereback, Anna-Lena, and Persson, Per (1999) Dealing with the Lurking Lutheran view on Interfaces: Evaluation of the Agneta and Frida system, Workshop on Behavior Planning for Life-Like Characters and Avatars, Elisabeth Andre, Angelica de Antonio Jiménez, and Kristina Höök (eds.) In Conjunction with the i3 Spring Days '99, March 9th -10th 1999, Sitges, Spain.
- [2] Annika Waern. "DatorVarelser". Populärvetenskaplig artikel i Framtider nr 2, 1998, institutet för framtidsstudier. Även i boken "The Global Tendency Machine - trycket från framtiden".
- [3] Annika Waern and Kristina Höök. Interface Agents: A new metaphor for human-computer interaction and its application to Universal Access. i ed. C. Stephanidis, 'User Interfaces for All', Lawrence Erlbaum, under utgivning.

Design och Utvärdering av Adaptiva Gränssnitt

Kristina Höök

När vi startade projektet 1996 hade vi en hel del kunskaper om hur man kan bygga så kallade intelligenta eller adaptiva gränssnitt ur en teknisk synvinkel, men väldigt få sådana system hade nått marknaden och var därmed inte testade av stora grupper användare. Flera intelligenta gränssnitt höll på att tas fram som kommersiella tillämpningar av bland andra Microsoft. Vi valde därför att ägna en del arbete i projektet åt att diskutera och analysera hur intelligenta gränssnitt ska se ut för att bli användbara, accepterade och spridda. På en grov nivå kan man dela in problematiken i två delar: problem som uppkommer eftersom intelligenta gränssnitt bryter mot den existerande "gränssnittskulturen", samt problem som beror på hur de skapats och vilka mål de uppfyller. Vi sammanfattar en liten del av vår analys nedan, en mer fullständig beskrivning finns i [Höök, under utgivning] samt i den tutorial som Kristina Höök givit på konferenserna IUI'98, IUI'99, samt British HCI'98 [Höök 1998].

Förutom de problem vi måste attackera framöver, finns det naturligtvis stora möjligheter. Annika Waern och Kristina Höök har visat på hur gränssnittsagenter öppnar upp en designrymd som kan vara till stor hjälp när man vill göra gränssnitt för alla ("design for all") [Waern och Höök, 1999, under utgivning]. Vi diskuterar även detta nedan.

Gränssnittskultur

Gränssnittsagenter kan användas för att anpassa systemet till en viss användare [Horvitz et al 1998]. Man tänker sig att problem såsom informationsöverflöd, att ge rätt sorts hjälp vid rätt tillfälle, kognitiv överbelastning, etc., skulle kunna lösas om systemet försökte anpassa sig till användaren. Det är tankarna bakom blanda andra Microsoft assistant - hjälparen i Microsoft Office paketet [Horvitz et al 1998].

Flera forskare har kritiserat den här ansatsen, bland andra Ben Shneiderman och Lucy Suchman [Shneiderman, 1997, Suchman, 1997]. Den här kritiken var delvis välmotiverad. Den är baserad på att tidiga AI-modeller av intelligens var alltför naiva. De förutsatte i alltför hög grad att människors hjälpbehov alltid är desamma, så att man genom att studera vad människor vill göra skulle kunna förutsäga alla användningssituationer och möjliga dialoger. Sådana system är kostsamma att utveckla, och blir rigida och svåra att förändra. De utvecklas inte i samklang med hur användningen av dem förändras och utvecklas.

Vidare oroar sig dessa kritiker för att intelligens i systemen skulle kunna innebära att ansvarsfördelningen mellan system och människa inte blir uppenbar. Om systemen blir aktiva och gör saker på ett intelligent och självständigt sätt - vems är då ansvaret när det går fel?

Slutligen anser bland andra Schneiderman att dessa system bryter mot de fungerade användbarhetsprinciper vi skapat för "vanliga" gränssnitt.

I [Höök, under utgivning] diskuterar vi den här problematiken och vi visar på kopplingen till vad vi kallar "gränssnittskultur". Så länge som vi betraktar gränssnitt som passiva verktyg genom vilka vi utför våra arbetsuppgifter så upplevs naturligtvis mer aktiva gränssnitt som okända djur. Men efterhand som vi vänjer oss vid gränssnittsagenter som kan ge oss aktiv hjälp, eller system som kan förstå våra intressen och rekommendera filmer till oss, så kommer vi att inlemma även dessa fenomen i våra förväntningar på ett system. Vi flyttar oss från en verktygssyn på gränssnitt till en mycket mer komplex bild av vad datorsystem är.

Låt oss ta ett par exempel. En viktig designregel för människa-maskin interaktion är att människan ska vara den som tar initiativet till dialogen hela tiden. Människan ska ha kontroll. I ett system skapat av Clanton kallat "You don't know Jack"¹¹ är det i stället systemet som har

¹¹ <http://www.berksys.com/products/jack/jack.html>

hela kontrollen. Systemet imiterar en TV-show, en frågesport, där användaren blir en av deltagarna. Systemet pressar användaren, ställer rappa frågor, om inget svar kommer går det vidare, etc. Fast detta bryter mot alla våra gränssnittsregler är det ett mycket underhållande program, som vunnit pris för sin design.

Ett annat, minde spektakulärt exempel är hur länkarna på webben ändrar färg när vi besökt dem. Detta är en förändring av gränssnittet, ett slags mycket enkel anpassning till användarens beteende. Men det är en förändring som betyder att gränssnittet inte ser exakt likadant ut när vi återvänder till en redan besökt sida. Det betyder att det i någon mån bryter mot regeln att ett gränssnitt ska se likadant ut när man går tillbaka till det - att det ska vara förutsägbart. Efterhand som användarna har vant sig vid att webben uppför sig på det här viset har det blivit en del av vår "web-kultur", och vi kan dra stor nytta av det när vi söker oss tillbaka till redan besökta sidor.

Det vi kallar direktmanipulation är ett slags gränssnittskultur med sina egna designregler. Dessa regler är inte för en gång allom givna utan är skapade och vi har vant oss vid dem. De har blivit en kultur som vi förstår. Webben har en annan uppsättning regler som inte alltid ser exakt likadant ut som direktmanipulations-gränssnittet och återigen har vi vant oss vid den och betraktar brott mot den kulturen som oanvändbara gränssnitt.

Aktiva, intelligenta, gränssnitt kommer antagligen att genomgå samma utveckling. När vi vant oss vid dem kommer vi att anse att det finns vissa designprinciper som de måste följa för att vara användbara. Poängen är att dessa designprinciper är inte grundade i att människor en gång för alla fungerar på ett visst sätt, beroende på vår fysionomi eller hur vår hjärna är uppbyggd. De fungerar därför att vi vant oss vid dem.

Naturligtvis finns det vissa principer för hur vår kognition fungerar som vi måste ta hänsyn till. Till exempel drar rörliga eller blinkande element i ett gränssnitt alltid till sig vår uppmärksamhet - det har att göra med hur vår perception fungerar. Men ovanpå det påverkar våra tidigare erfarenheter, vår kultur, i allra högsta grad vad vi upplever som användbart, vackert eller funktionsdugligt.

Rätt design?

För att skapa acceptans för aktiva, intelligenta gränssnitt krävs det dock att de faktiskt är till nytta och att de presenteras på ett sätt som användare förstår och kan relatera till. De måste i någon mån bygga på den existerande kulturen för att kunna ta steget vidare.

Ett stort problem med de intelligenta gränssnitt som vi ser idag är att de skapats av forskare inom området 'artificiell intelligens', forskare som i första hand velat utveckla intelligensen i systemet och först i andra hand varit intresserade av hur den kan utnyttjas för att förbättra interaktionen. Det saknas en hel del arbete på designprinciper för dessa nya gränssnitt. Precis som vid all nytveckling av teknik, saknas det också kostnadseffektiva metoder för utveckling av intelligenta gränssnitt, samt metoder, tekniker och verktyg för att underhålla systemen.

Möjligheter för alla?

Vi vet att vi i framtiden måste bygga system för en mycket större och mer diversifierad användargrupp än vad man normalt gör idag. Det kommer inte längre att vara möjligt att prata om den "normale" användaren. Vi har tittat lite på om gränssnittsagenter kan användas för att göra system användbara för en mycket vidare grupp användare.

Det som gör gränssnittsagenter intressanta i det här sammanhanget är att de öppnar helt nya designmöjligheter. En del av dem skulle kunna användas för att nå användargrupper som annars har svårt att relatera till datorer och datorsystem. Andra är mer knutna till nya användningsområden, där vi behöver utvidga vår syn på datorer att inte längre bara handla om

verktyg, utan också om underhållning, social interaktion, etc. [Waern and Höök 1999 under utgivning].

Förteckning över bilagor inom området 'Design och utvärdering av adaptiva gränssnitt'

- [1] Kristina Höök, Designing and Evaluating Intelligent User Interfaces, in proceedings of IUI'98, IUI'99 and British HCI'98, 1998 and 1999.
- [2] Höök, K. (1998) Evaluating the Utility and Usability of an Adaptive Hypermedia System, in Journal of Knowledge-Based Systems, Volume 10, issue 5, 1998.
- [3] Kristina Höök. Steps to Take Before IUI Becomes Real. in Interacting with Computers, forthcoming.

Gränssnitt för Handikappade

Kent Saxin-Hammarström och Kristina Höök

I denna aktivitet har vi studerat hur interaktion med MUD-miljöer kan handikappanpassas. Den centrala tesen i detta arbete har varit att hur information presenteras inte bör ensidigt bestämmas av informationsproducenten, utan ska kunna ställas in av den enskilde användaren. En uppsättning standardkonfigurationer får gärna finnas, men det slutgiltiga valet skall ligga hos konsumenten. Denna ledande princip anser vi kunna vara av nytta för alla användare, men den är helt nödvändig för funktionshindrade användare, eftersom det är osannolikt att en producent skall lyckas förutsäga alla individuella användares behov. Det bästa man skulle kunna hoppas på i det fallet är minsta gemensamma nämnaren – en design för 'genomsnittsanvändaren' snarare än för alla användare.

I nära samarbete med projektet Convene har vi utvecklat idéer om hur multimodalitet (eller *intermodalitet*) kan användas som ett sätt att hjälpa funktionshindrade användare att använda s.k. MUD-system. Detta projektsamarbete är helt i linje med IntAge-projektets syfte att lämna synen på gränssnittsdesign som en design för en viss användartyp, till att arbeta utifrån perspektivet att alla användare är individer med sina egna egenheter och behov.

Convene-projektet handlar om att förbättra gränssnitten till textbaserade, virtuella miljöer (MUD-system) som används för rekreation och sociala ändamål, speciellt för användare med funktionshinder (en grupp som utgör 10-20% av världsbefolkningen). MUD-system är nätverksbaserade system (framför allt via Internet) som sedan tidigt åttiotal använts för avkoppling och fritid, spel och annan social interaktion. En indikation på deras popularitet och stora mängd användare är att det finns tusentals sådana system tillgängliga via Internet. MUD-system är potentiellt mycket användbara "sociala verktyg" för de som är hindrade att delta helt och hållet i sociala relationer och fritidsaktiviteter. Hitills har de dock inte uppmärksammats vad gäller att göra dem lättare att använda. Över huvud taget finns mycket litet gjort när det gäller icke-instrumentell användning av datorer, vilket tycks oss något märkligt med tanke på hur äldre kommunikationsteknologier används, jämför till exempel med telefonen.

Studier

Convene-projektets mål har varit att ta reda på vilka specifika problem funktionshindrade användare stöter på i dessa miljöer och hur dessa kan lösas. Hitintills har två pilotstudier gjorts i detta syfte: En med en grupp användare med en specifik form av muskelförtvinning som leder till dålig finmotorik och en med en grupp blinda användare. Båda dessa studier indikerar att det primära problemet i mjukvarugränssnittet är av samma art som för användare utan funktionshinder, nämligen informationsöverflöd. Skillnaderna är kvantitativa snarare än kvalitativa, det vill säga de är likartade, men förvärras av funktionshinder. Ett uppenbart undantag är när en sensorisk eller motorisk kanal hos användaren försämrats till den grad att den inte kan användas eller behöver kompletteras.

Snarare än att ge upphov till kvalitativt annorlunda problem, besannas den gamla sanningen att utvecklingsarbete för funktionshindrade nästan alltid är till nytta för alla användare.

Intermodalitet

Kännetecknande för en MUD, till skillnad från andra former av datorstödd kommunikation som epost och elektroniska anslagstavlor, är kravet på interaktion i realtid, dels mellan flera användare och dels mellan dessa användare och en virtuell värld. Existerande system är inte alls anpassade för användare med extraordinära behov, utom när det skett av en slump (många har t.ex. tillgång till användardefinierade kortkommandon).

En metod för att underlätta anpassning av systemen är att generera interaktionen från en generell, mer grundläggande representation på det sätt som multimodala presentationssystem gör. Ett problem med det angreppssättet är att det mesta arbete som gjorts i fråga om multimodala presentationssystem inte har berört realtidskommunikation. Mest har det handlat om planering och syntes av presentationer utifrån en statisk kunskapsbas och en fråga från användaren eller ett kommunikativt mål från producenten. I MUD-system finns inga sådana mål hos presentationssystemet (utom det att underlätta kommunikationen). I stället tillkommer en annan uppgift: att analysera information till en form som lättare lånar sig till olika modaliteter.

För att förbättra dessa gränssnitt har vi valt att koncentrera oss på multimodalitet, i den form vi döpt till *intermodalitet*. Med intermodala system menar vi modulärt utformade system med en arkitektur som underlättar översättning av information från en modalitet till en annan, som fungerar bättre för användaren i en viss situation. Tanken är att man efter behov enkelt skall kunna stoppa in nya moduler för att analysera och presentera de modaliteter som behövs. Initialt har vi valt att arbeta med ren text, eftersom det är den dominerande interaktionsformen i MUD-system. I princip tar vi ett flöde av text från MUD-världen, t.ex. rumsbeskrivningar, dialoger mellan användare, händelser i världen och delar upp det i olika kanaler: Rumsbeskrivningskanalen, dialogkanalen osv. Användaren kan sedan själv välja om varje enskild kanal skall presenteras som ljud, text, bilder eller annat. Exempelvis skulle rumsbeskrivningar kunna presenteras i normal textform, medan dialoger presenteras via talsyntes och världsgeografin (hur rummen är förbundna) som en karta.

Denna ansats ger två fördelar.

- Den tillåter kompensation för nedsatt funktion genom att byta modalitet efter behov. I stället för att skapa ett helt nytt system för varje grupp av funktionshinder räcker det att kombinera rätt delar och ev. nyskapa någon enstaka modul,
- Arkitekturen bör reducera användarens kognitiva belastning genom att låta information presenteras i komplementära kanaler (t.ex. dialoger i tal och beskrivningar på en Braille-skärm).

Båda dessa egenskaper torde vara användbara även i andra former av datorförmedlad kommunikation där innehållet är delvis stiliserat, t.ex. epost, elektroniska anslagstavlor och s.k. "chat"-system.

Förutom användarstudierna pågår arbete med en prototyp för att illustrera intermodalitetsprincipen. Hitintills har grundarkitekturen och en textextraktionsmodul implementerats. Kvar att göra är en eller ett par presentations- eller inmatningsmoduler innan principen kan demonstreras. Projektet har en praktisk och problemstyrd, snarare än teknologistyd, inriktning: Vi vill skapa en mjukvaruarkitektur som tillåter att man skapar system som gör existerande miljöer mer tillgängliga med de resurser en användare normalt har till förfogande. Vi har alltså koncentrerat oss på det ramverk som behövs för att sätta samman existerande tekniker – som Internet, talsyntes och informationsextraktion – snarare än innovativa tekniker.

Delar av arbetet har publicerats i form av dels ett anförande vid konferensen "User Interfaces for All" i Strasbourg 1997 [Saxin-Hammarström 1997] och dels som ett kapitel i antologin "Users in Action!", 1999, redigerad av Britt Östlund och utgiven av KFB. Vidare har informella, inbjudna presentationer hållits vid Hjälpmedelinstitutets Informations- och Demonstrationsdagar 1998, Riksförbundet för Rörelsehindrad Barn och Ungdoms IT-ombudsmannamöte 1997, samt i en seminarieserie vid Tema K, Linköpings universitet. En mer omfattande presentation av projektet kommer att föreligga under hösten 1999 i form av en licentiatavhandling (Kent Saxin Hammarström).

Förteckning över bilagor inom området 'Gränssnitt för handikappade'

Kent Saxin Hammarström, Höök, Kristina and Ereback, Anna-Lena, Convene - MUD interfaces for disabled users, in the ERCIM workshop "User Interfaces for All", edited by Stephanidis, C., D. Benyon, N. Carbonell, P.L. Emiliani, G. Faconti, A. Kobsa, A. Waern, M. Wilson (1997) 3rd ERCIM Workshop on User Interfaces for All. Obernai, November 2-4 1997. INRIA/ERCIM.

Kent Saxin Hammarström, Anna-Lena Ereback, Kristina Höök, and Marie Sjölander. Convene - MUD Interfaces for Disabled Users, in ed. Britt Östlund "Users In Action!" (1999), KFB-Rapport 1999:8.

Samhällsinformation och medborgardialog

Annika Waern, Fredrik Espinoza och Mark Tierney

Området 'Samhällsinformation och medborgardialog' fanns med i den ursprungliga ansökan, men finansierades inte inom projektet på grund av begränsade projektmedel. Däremot deltog flera personer från IntAge parallellt i det EU-finansierade projektet KIMSAC [Charlton et al 1997], som handlade om dessa frågor.

KIMSAC (Kiosk-based Multimedial Service Access for Citizens) var ett europeiskt samarbetsprojekt som löpte under åren 1996-1998. Deltagande var företag och forskningsorganisationer från Sverige, Irland, Frankrike, England och Italien. Fältförsöken gjordes på Irland. Syftet var att utveckla ett agentbaserat informationssystem för arbetssökande. Systemet kunde presentera ett personligt anpassat urval av information om både lediga tjänster, studieplatser samt arbetslöshetsunderstöd. Systemet testades i praktiken under två perioder, och var då utplacerat på socialbyråer och arbetsförmedlingar.

SICS bidrag till KIMSAC

SICS insatser i KIMSAC handlade framför allt om tre olika frågeställningar:

- På vilket sätt kan gränssnittsagenter användas i denna typ av informationssystem?
- Hur kan informationsagenter från olika organisationer fås att samarbeta och dela information?
- Hur kan användarens interaktion med flera olika informationsagenter samordnas och individanpassas på ett enhetligt sätt?



Figur 2. En bild av gränssnittet för KIMSAC-systemet.

Gränssnittsagenter i KIMSAC

Tidigare erfarenheter av gränssnittsagenter har visat att de inte är så uppskattade när de uppträder som rena 'hjälpredor'. Ett bra exempel på detta är det lilla gemet i Microsoft Office, som de flesta stänger av nästan genast. Det beror på två saker: dels är det generellt sett svårt att ge folk rätt hjälpmeddelande, dels stör det alltid folk när det kommer ett hjälpmeddelande, även om det är rätt hjälp. Det senare eftersom när man behöver hjälp, så är man extremt koncentrerad på det man håller på med, och meddelandet blir därför störande. Det finns studier som visar att aktiva hjälpsystem är illa omtyckta även om de faktiskt hjälper folk att göra saker fortare eller med färre fel! Det här har inte så mycket att göra med om man använder en gränssnittsagent eller bara en dialogruta, men situationen blir inte bättre av att man introducerar en agent.

En av avsikterna med KIMSAC var att försöka hitta en modell för hjälpagenter som fungerade. Tidigt i projektet tog vi fram flera olika designförslag för gränssnittsagenten. Den modell som vi slutligen använde var en där agenten var ganska stiliserad. I den första versionen av KIMSAC gav agenten användaranpassad hjälp *hela tiden* på en avgränsad del av skärmen (se figur 2). Det här var ett viktigt designval, eftersom det gjorde att användaren inte blev störd eller överraskad av hjälpmeddelandena. Nya användare använde sig mycket av hjälptexterna, medan mer vana användare kunde ignorera dem. Den här modellen för hjälp verkar fungera bra just för kioskbaseade system, eftersom varje användare använder systemet ganska få gånger och ganska kort tid varje gång.

Samverkan mellan tjänster

Den agentarkitektur som vi använde i KIMSAC var inte helt 'öppen'. Eftersom alla agenter i KIMSAC utvecklades inom projektet så antog vi att agenter var 'altruistiska' - de tog inte betalt för sina tjänster och de hjälptes alltid åt när de kunde. Det här förenklade sättet att se på agentmiljöer är ganska vanligt inom agentforskningen. Det reducerar problemet med agentkommunikation till en fråga om att se till att agenter kan hitta varandra och utbyta information. Det här sättet att se på agentkommunikation, *informationskommunikation*, har

dominerat både det USA-baserade forskningsnätverket KSE (Knowledge Sharing Effort) och det europabaserade initiativet FIPA (Foundations of Intelligent Physical Agents)¹².

Det centrala problemet för informationskommunikation är att agenter som skickar information emellan sig måste ha ett gemensamt sätt att tolka informationen. En agent kan skicka information som en annan agent inte alls förstår och kan agera på ('Mitt selknä har gått sönder'). Alternativt kan samma term kan betyda olika saker för olika agenter ('Vägen delar sig i en gaffel'). För att få informationsagenterna som vi utvecklade i KIMSAC att kunna prata med varandra, måste vi alltså först reda ut vilka begrepp som användes inom de båda informationsområden som vi arbetade med (arbetsförmedling och arbetslöshetsunderstöd/socialbidrag). Det här visade sig vara ett mycket svårt problem, vars lösning inte i första hand är teknisk utan kräver en kombination av mänsklig och maskinintelligens. Vi diskuterar detta i mer detalj under rubriken 'erfarenheter från KIMSAC'.

Integrerad presentation

Ett problem som visade sig centralt för hela KIMSAC-projektet var att åstadkomma ett enhetligt sätt för agenter att interagera med användare. För att åstadkomma detta utvecklade vi en prototyp för ett interaktionssystem, sicsDAIS. (sicsDAIS har senare vidareutvecklats inom IntAge-projektet och beskrivs i mer detalj i avsnittet 'agentbaserade gränssnitt emot öppna tjänstemiljöer'.) I KIMSAC använde vi sicsDAIS för att ge agenterna en möjlighet att använda gemensamma komponenter för att presentera sin information och interagera med användare. Vi lät också sicsDAIS generera en del av layouten av komponenter. En egenskap i sicsDAIS som visade sig otroligt användbar var att sicsDAIS innehåller en lokal databas som gör att presentationskomponenter kan spara sitt tillstånd och dela information. Detta gjorde det dels möjligt att minimera kommunikationen med det underliggande agentsystemet, och dels gjorde det möjligt att köra åtminstone delar av systemet utan att det alls har nätåtkomst. I praktiken fungerade sicsDAIS som ett sätt att flytta en del av funktionaliteten från 'server'sidan till kioskerna när behov uppstod. En bieffekt av detta var att mindre personlig information skickades över nätverket, något som i viss mån bidrog till säkerhetsaspekter av systemet.

Erfarenheter från KIMSAC

Den allra viktigaste erfarenheten från KIMSAC är att *all* samordning mellan tjänster kräver att tjänsteleverantörerna är överens om vilken information som ska erbjudas användarna och andra tjänster. Detta gäller så väl samordnad presentation, samordning av tjänster i bakgrunden, och enhetlig personlig anpassning. KIMSAC utgick från början ifrån ett antal existerande webbsidor, som innehöll den information som skulle integreras. Men ofta fanns otydliga eller oskrivna krav som gjorde det mycket svårt att avgöra om en viss regel skulle tillämpas på en viss individ. Speciellt besvärligt blev det när man skulle sammanföra information från arbetslöshetskassan och socialbidragskassan. Det kunde till exempel vara nästan omöjligt att avgöra om en viss person var berättigad till ett visst bidrag om han eller hon tog en viss studieplats.

Egentligen förekommer den här typen av problem i alla system som försöker representera lagliga eller administrativa regler och tillämpa dem på individuella fall. Vad som tillkom i KIMSAC var att den här informationen dessutom måste kommuniceras mellan system som tillhörde olika organisationer. I praktiken visade sig detta problem nästan oöverkomligt. Det fanns både för lite kompetens om regelsystemen inom varje organisation, och för lite samsyn mellan organisationerna, för att komma fram till en heltäckande integrerad modell inom ramarna för projektet.

¹² <http://www.fipa.org/>

Det finns bara en riktigt bra lösning på denna typ av problem, och det är att se till att bygga riktigt bra verktyg för att folk inom de berörda organisationerna själva ska kunna underhålla informationsmodellerna. Helst ska dessa verktyg vara så enkla och åskådliga att handläggare ute på kontoren kan bidra med förslag till lösningar eller exempel på situationer när systemen gör fel. Dessutom bör verktygen samverka i ett distribuerat nätverk, så att den gemensamma kompetensen kan tillvaratas. Vi tror att sådana verktyg i sin tur bör utvecklas som tjänster i en öppen tjänstemiljö. Idag finns inga sådana tjänster att tillgå, och det fanns ingen möjlighet att utveckla stödverktyg inom ramarna för KIMSAC-projektet. Det här är en fråga vi vill fortsätta jobba med.

En annan observation var att det fanns en motsättning mellan önskemålen på funktion från de inblandade myndigheterna, och önskemålen från den individuella användaren. Studier av tidigare liknande system har visat att den funktion som uppskattas mest är s.k. 'om så'-scenarior; en funktion som t.ex. visar vad som händer med arbetslöshetsunderstödet om man tar ett deltidsjobb. Det här var en funktion som de inblandade myndigheterna ville lägga restriktioner på. I synnerhet ville man inte att systemet skulle 'tipsa' personer om bidrag som de skulle kunna bli berättigade till. Hellre ville man att systemet skulle uppmuntra folk att söka arbete eller studieplatser.

Återigen är det här inte ett problem som direkt har att göra med tekniken för agentbaserade öppna tjänstarkitekturer; samma problem uppstår även med system för enskilda myndigheter. Men motsättningen blir tydligare i den här typen av integrerade system, eftersom integreringen kan göra det uppenbart hur regler för olika myndigheter interagerar med varandra, både till förmån och till nackdel för individen.

I KIMSAC bestämde vi oss ändå för att ha med 'om så'-scenarior. Vår förhoppning var att om systemet kunde synliggöra alla konstiga interaktioner mellan regelverk för en vanlig arbetssökande, så skulle detta ge ett ökat incitament för de berörda myndigheterna att strama upp och förbättra sitt regelverk.

Förteckning över bilagor inom området 'Samhällsinformation och Medborgardialog'

- [1] P. Charlton, Y. Chen, E. Mamdani, J. Pitt, F. Espinoza, O. Olsson, A. Waern, and F. Somers. 1997. Open Agent Architecture Supporting Multimedia Services on Public Information Kiosks. In Proceedings of the 3rd International Conference on the Practical Application of Intelligent Agents and Multi-Agent Methods, pp. 445-465, London, England, April.

Agentbaserad Interaktion i Öppna Tjänstemiljöer

Fredrik Espinoza, Markus Bylund, Mark Tierney och Annika Waern

Framtidens nätverk

Idag är den dominerande tekniken för distribuerade tjänster World Wide Webb. I framtiden kommer tjänster i nätverk att distribueras i betydligt mer generella och öppna tjänstemiljöer (OSE-Open Service Environment). Webben skiljer sig från en sådan miljö på ett antal sätt.

- Webben är bara delvis öppen
- Webben är producentcentrisk. Det är producenterna av tjänsterna och informationen som bestämmer vad som ska erbjudas och hur det ska erbjudas.
- Webben är inte byggd för att vara dynamisk vad gäller typer av tjänster, användningsplattformar, interaktionssätt, och användare.

Följaktligen finns det många problem med att använda World Wide Web som medium för öppna tjänste miljöer.

- *Informationsproblemet.* För att tjänster ska kunna anpassas och skräddarsys åt användaren behöver de tillgång till personlig information. Hur ska sådan information hanteras och delges tjänster på ett säkert sätt?
- *Ontologiproblemet.* Hur kan oberoende tjänster som samverkar för att lösa uppgifter åt användaren förstå varandra?
- *Apparat- och nätverksberoende.* Hur kan tjänsteåtkomsten flyttas från apparat till apparat och hur kan man behålla delar av tjänsters funktionalitet trots brister i eller avsaknad av ett nätverk?

Vi anser att en fullfjädrad OSE kan lösa dessa och andra relaterade problem.

Användaren i centrum

Det viktigaste målet för vår forskning är att skifta fokus från tjänsteproducenterna till användarna av öppna tjänstemiljöer. Vi vill ge varje användare större kontroll över sin situation inklusive information, kommunikation och tjänster. Vi vill också göra det lätt för tjänster och användare att samarbeta på användarens villkor—inte på tjänsteproducentens villkor.

En öppen tjänstemiljö

I en global nätmiljö som blir mer och mer öppen för mindre aktörer behövs det nya metoder, paradigmer, och arkitekturer, som möjliggör en meningsfull samverkan mellan alla användare.

I denna aktivitet har vi valt att utgå ifrån dagens Internet-teknologi i skapandet av nya, mer användarcentrerade lösningar. Internet håller på att bli en 'de facto' standard för en mångfald av aktiviteter: informationsspridning, informationssökning, handel, och underhållning. Internet har uppenbarligen en stor potential då det gäller att förverkliga dessa.

Eftersom en OSE kommer att behöva hantera stora mängder information och godtyckligt komplexa tjänster måste systemen i sig vara komplexa. Men komplexiteten hos systemen får inte bli till en börda för användaren. Vi behöver nya sätt att utveckla system samt nya sätt för att interagera med dessa komplexa system.

Öppna tjänstemiljöer (OSEer) tillåter komplexa system att byggas och användas inom olika användningsområden. En OSE är en modulär arkitektur som är flexibel, dynamisk, implementationsberoende, och viktigast av allt: öppen.

- Den tillåter att nya funktioner introduceras med hjälp av ett enkelt system för pluggbara komponenter.
- Funktionerna kan förändras under tiden som systemen används.
- Arkitekturen är robust tack vare redundans bland komponenterna.
- Systemet är skalbart eftersom det går att distribuera aktiviteter över federationer av system.
- Komponenter kan byggas för olika plattformar med olika teknologier.
- Tjänster kan nå direkt eller indirekt genom olika användargränssnitt.

I följande avsnitt ska vi vidare utforska den öppna tjänstemiljön. Vi kommer att definiera begreppet OSE, förklara egenskaper och komponenter, ge ett exempel (World Wide Web), och slutligen nämna vilka krav en OSE ställer på infrastruktur och implementation.

En informell definition av OSE

I följande avsnitt ska vi vidare förklara den öppna tjänstemiljön. Vi kommer att definiera OSE, förklara egenskaper och komponenter, ge ett exempel (World Wide Web), och slutligen nämna vilka krav en OSE ställer på infrastruktur och implementation.

Öppenhet. Först och främst är en OSE öppen. Den kan rymma vilkens sorts tjänst eller användare som helst oavsett vilken tjänst det är eller vad användaren har för uppgift. Tjänster och användare kan introduceras när som helst. De kan komma och gå, dvs de är inte nödvändigtvis alltid kopplade till nätverket. Användare kan förekomma i systemet för att lösa en enskild uppgift eller för att lösa flera uppgifter under en längre tidsperiod. Tjänster och interaktionssystem för att använda tjänster i OSEn kan byggas för olika plattformar och apparater.

Distribution. Tjänster och användare kan finnas var som helst i nätverket. Distributionen bidrar till stabiliteten och robustheten eftersom tjänsters funktionalitet kan spridas över nätverket.

Heterogenitet. Tjänster kan erbjudas av personer, institutioner, eller företag. Det finns inga krav eller begränsningar på vad en tjänst måste eller kan erbjuda eller på det sätt som tjänsten erbjuds. Detta betyder att en tjänst kan vara hur komplex som helst (som en personlig banktjänst) eller hur enkel som helst (som en valutaomvandlare). En användare kan inte anta ett speciellt format för tjänsten om inte detta är överenskommet. Användare kan vara människor eller tjänster och användningssätten kan variera i form; allt ifrån en mänsklig användare som når en tjänst genom ett speciellt användargränssnitt till en tjänst som använder en annan indirekt.

Transparens mot utrustning. En tjänst är generell i den mening att den service som erbjuds inte är beroende av den hårdvara som används för att nå tjänsten. Detta innebär att vilken utrustning som helst kan användas för att utnyttja tjänster i en OSE. De apparater som kan tänkas är bland andra: stationära datorer, bärbara PDAer, mobiltelefoner, publika terminaler, etc. Det innebär att en tjänst utåt sett erbjuder samma information i samma format oavsett vilken utrustning som används. I stället är det OSEn som ansvarar för att översätta informationen mellan olika typer av utrustning. Tjänsterna är dock medvetna om vilken utrustningstyp som används, så att de ges möjlighet att anpassa sina tjänster efter detta. Exempelvis kan en tjänst ha olika format på information som skickas till en grafisk terminal och till en WAP-telefon.

Förmåga att utnyttja flera interaktionssätt samtidigt. Tjänster kan användas från olika sorters utrustning samtidigt. Detta leder till ökade möjligheter och dimensioner i interaktionen mellan användare och tjänster. Till exempel kan en användare genom sin mobiltelefon använda naturligt språk via en mobiltelefon, samtidigt som tjänsten svarar genom att visa grafisk information på en skärm. Byten av modaliteter sker naturligt i systemet. Det går också att byta mellan olika instanser av samma apparat utan att tappa sammanhanget, till exempel när en användare byter från datorn på jobbet till datorn hemma. De tekniska problem som vi stöter på i detta sammanhang liknar i mycket de som uppstår vid intermodalitet för handikappinteraktion, se avsnittet om 'gränssnitt för handikappade'.

Koordinering av tjänster. I en OSE kan tjänster koordineras för att lösa uppgifter åt användare. Helt skilda tjänster kan kommunicera, samarbeta, tävla, etc. I vissa fall sköts koordinationen av en tredje part och i andra fall sköter de inblandade tjänsterna koordination själva eller med hjälp av användaren.

Ett exempel—World Wide Web

Ett speciellt intressant exempel på en öppen tjänstemiljö är World Wide Web. Webben är ett system av sammanlänkade webbplatser som används för att publicera information och tjänster. Systemet är baserat på det öppna protokollet HyperText Transport Protocol (HTTP) som används som kommunikationsprotokoll, och HyperText Markup Language (HTML) som används för att beskriva innehåll. Webbserverar publicerar informationen och klienter läser informationen men hjälp av webbläsare. Nya webbplatser kan lätt läggas till genom att sätta upp nya serverar.

Dessvärre lider webben av skalbarhetsproblem. När antalet webbplatser ökar blir systemet svårt att använda. Delvis beror detta på att nättrafiken också ökar och att bandbredden är för låg, men huvudsakligen beror det på att det saknas bra verktyg för att komma åt informationen och tjänsterna. Bland annat är det svårt att hitta det man letar efter. Det behövs en mer avancerad arkitektur för att göra det möjligt att bygga bättre sökverktyg. En möjlig utvidgning är att införa nya sätt att klassificera informationen med avseende på innehåll. Även om detta håller på att införas av World Wide Web Consortium (W3C) med XML (Extensible Markup Language), anser vi att det görs för lite på tjänstenivån. I en sann öppen tjänstemiljö kan man integrera avancerade tjänster för att ge användare flexibilitet och lättare åtkomst till informationen och tjänsterna.

Webben brister speciellt på följande punkter.

- *Brist på individualisering.* Det finns för lite stöd för att skräddarsy tjänster på webben. I princip är det enda sättet att låta en användare identifiera sig för en tjänst, så att den kan anpassa informationen från källan (tjänsten)¹³. Användarens kontroll över denna process är mycket begränsad, och all information som behövs, oavsett hur känslig den är, måste skickas via nätverket till tjänsten.
- *Apparatberoende.* Det finns ingen klar separering mellan innehåll och format för presentation. Det gör att tjänsterna blir beroende av vilken typ av access användaren brukar. Cascading Style Sheets (CSS) hjälper, men dess syfte är att formatera text och det gör det användbart enbart för apparater som hanterar text.
- *Nätverksberoende.* Användarens klient (webbläsaren) är beroende av en koppling till nätverket för att den ska kunna nå tjänster och information. Om kopplingen inte finns kommer all interaktion att avstanna.

¹³ Detta görs oftast genom att tjänsten tilldelar användaren ett användarnamn och ett passord. Vid upprepade tillträden till tjänsten används en *cookie* för att användaren ska slippa uppge namn och passord igen.

- *Begränsat stöd för varaktighet.* Information om användarens interaktion med tjänster sparas inte mellan sessioner (d v s *state*). Därför är det svårt att återuppta en aktivitet om interaktionen bryts av någon anledning eller om användaren vill fortsätta vid ett senare tillfälle.
- *Begränsade möjligheter för samarbete mellan tjänster.* Samarbete mellan oberoende tjänster är i stort sett obefintligt. Om det är möjligt sker det på tjänstens sida, utom räckhåll för användaren, och inte i användarens miljö där det är möjligt att övervaka och kontrollera det som försiggår.

I nästa avsnitt presenterar vi ett antal användningssituationer som syftar till att accentuera de möjligheter och krav som vi har beskrivit ovan.

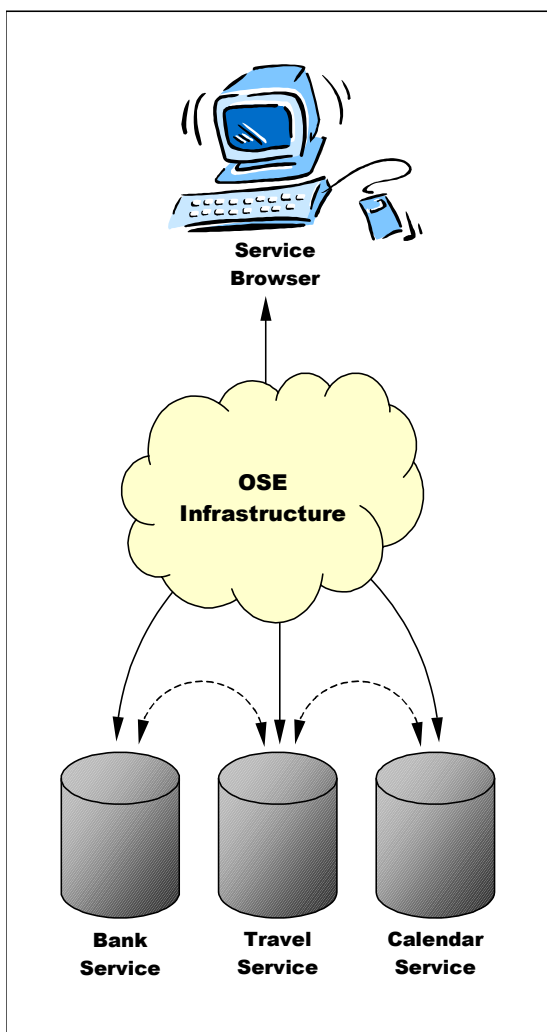
Användningssituationer

Följande tre användningssituationer beskriver några vardagliga situationer som kan uppstå när man jobbar i en uppkopplad värld. De demonstrerar typiska aktiviteter som är möjliga och önskvärda i en OSE. Varje situation beskriver de utmaningar och de problemkategorier som berörs av just det situationen. De utmaningar vi ser kan sammanfattas i följande punkter.

- Kontrollerad utdelning av information om användaren
- Användarassisterad ontologianvändning
- Användarkontroll
- Nätverksberoende
- Apparatoberoende

Följande situationer beskrivs: att köpa en resa, turistinformation, och kalendersynkronisering. Efter varje scenario sammanfattar vi problemen och utmaningarna, samt beskriver våra lösningar.

Scenario 1: Att köpa en resa från en resebyrå



Figur 3. Att köpa en resa från en resebyråtjänst

Som vi har nämnt tidigare i texten anser vi att utvecklingen av nätverksbaserade tjänster bör röra sig mot en användarcentrerad i stället för en producentcentrerad modell. World Wide Web representerar motsatsen: tjänsteproducenter ökar hela tiden gradvis sin kontroll över interaktionen med användare. Interaktionen byts från att vara ett samspel mellan producenten och användaren, till att vara en mera enkelriktad kanallik ström av information från producenten till användaren. Användaren reduceras till att vara en passiv åskådare som matas med information på producentens villkor. Information om användaren efterfrågas av nästan alla producenter och så fort information har delgivits en tjänsteproducent i hopp om att få en bättre tjänst, övergår informationen till att bli tjänsteproducentens egendom. Tjänsteproducenten kan nu göra vad som helst med informationen. Användaren har uppenbarligen förlorat kontrollen.

I detta scenario ser vi ett exempel på en motsatt utveckling där initiativet och kontrollen har flyttats till användaren. Scenariot illustrerar ett antal viktiga egenskaper hos en OSE och en personlig tjänste miljö (PSE—Personal Service Environment¹⁴):

- Användaren är initiativtagare och ansvarig för interaktionen med tjänster. Han eller hon initierar eller övervakar initieringen av alla kopplingar för koordination mellan tjänster. Till exempel: när resebyråtjänsten kontaktar banktjänsten iordningställs ett s k

I vårt första scenario vill användaren via en resebyråtjänst planera och köpa en affärsresa. I sin användarmiljö har hon också tillgång till en banktjänst och en kalendertjänst. Användaren anropar resebyråtjänsten och genomför bokningen av resa samt hyrbil i samarbete med tjänsten. Användarens preferenser angående platstilldelning på flyget eller favoritbiluthyrare får resebyråtjänsten automatiskt tillgång till.

I nästa steg söker resebyråtjänsten i användarens personliga tjänstemiljö efter en tjänst som kan betala för resan. I bästa fall kan tjänsterna hitta varandra, förstå varandra, och komma överens om språket när de kommunicerar. I det här fallet går inte det. I stället får användaren hjälpa till. Resebyråtjänsten ber användaren om hjälp och användaren kopplar ihop den med banktjänsten. Användaren assisterar sedan de två tjänsterna genom att matcha eventuella skillnader i terminologi. Resultaten av interaktionen sparas för framtida bruk.

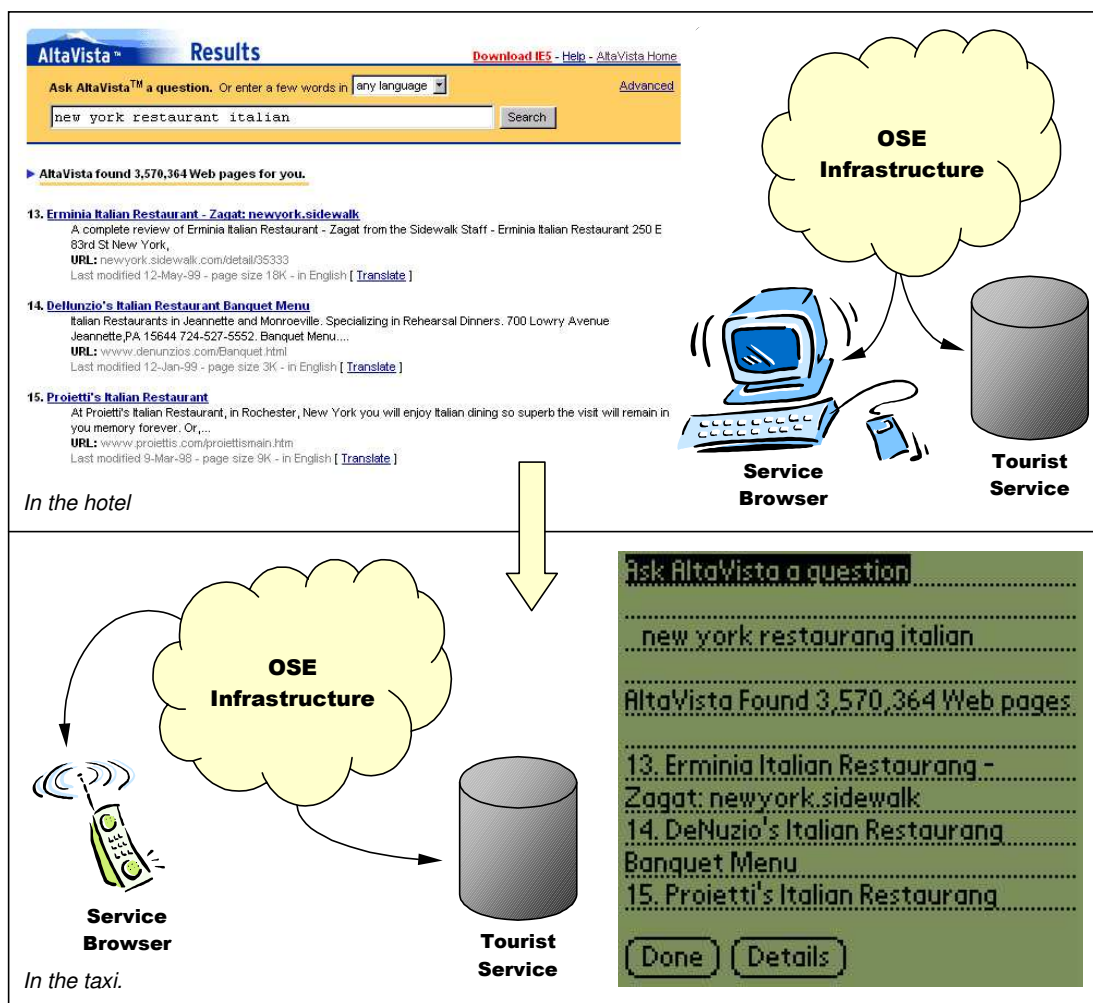
Kalendertjänsten uppdateras med aktuella datum och tider på samma sätt som beskrivs ovan. Återigen är användaren inblandad, denna gång när hon lägger till ytterligare information om kontakter och telefonnummer som har med resan att göra.

¹⁴ PSEn är användarens personliga tjänstemiljö som innehåller individanpassade versioner av tjänster.

tjänstekontrakt. Banktjänsten behöver veta saker som vilket konto pengar ska dras ifrån vid betalningar till den aktuella tjänsten. Den behöver också veta om användaren vill kontrollera framtida betalningar till den aktuella tjänsten eller om betalningar upp till en viss gräns ska få ske automatiskt. Användaren bidrar med denna information genom att interagera direkt med banktjänsten vid det tillfälle då tjänstekontraktet sammanställs.

- Användaren interagerar med tjänster på ett sätt som i huvudsak passar användaren—inte på det sätt som passar tjänsteproducenten.
- Information om användaren delas ut till tjänster om det behövs. Utdelningen sker på användarens villkor och under användarens kontroll. I vissa fall är det meningsfullt att dela ut sådan information, t ex när resebyråtjänsten ska boka flygbiljetter och behöver veta vilken sorts plats användaren föredrar.
- Olika tjänster kan samarbeta och koordineras i syfte att nå gemensamma mål åt användaren. En stor del av koordineringen sker inom användarens personliga tjänstemiljö. I många fall är detta möjligt genom att använda gemensamma ontologier (varav en del är dynamiskt sammansatta), och i vissa fall sker det som följd av överenskommelser mellan tjänsteproducenter. I en del fall måste användaren hjälpa till.

Scenario 2: turistinformation



Figur 4. Turistinformation

En man sitter på ett hotellrum i en för honom obekant stad efter ett affärsmöte. Han använder sin e-tjänstbläddrare på sin nätverksanslutna bärbara dator för att leta efter en restaurang till kvällen. Efter att ha provat sig fram ett tag så har han just hittat den perfekta sökordskombinationen när hans kollega knackar på dörren för att tala om att taxin som ska ta dem till stadens centrum väntar vid hotellets reception. Mannen har ännu inte funnit en passande restaurang, men han har kommit en bra bit på väg; han behöver nu bara jämföra det fåtal träffar som sökmotorn funnit. Hans bärbara dator är dock för stor och klumpig för att kunna tas med i taxin, dessutom saknar den en trådlös nätverksanslutning vilket krävs för att han ska kunna utvärdera sökträffarna. Som tur är så har han även en mobiltelefon med inbyggd e-tjänstbläddrare. Med ett fåtal klick på sin bärbara dator så talar han om för sin e-tjänstbläddrare att han inte längre är tillgänglig via sin bärbara dator, utan via sin telefon istället. Dessa kommandon instruerar e-tjänstmiljön att automatiskt flytta interaktionen från den bärbara datorn till mobiltelefonen genom att frysa interaktionen mellan sökmotorn och den bärbara datorn, för att återuppliva den mot mobiltelefonen. Väl i taxin fortsätter mannen sin sökning efter en lämplig restaurang exakt där han slutade på hotellrummet, om än med ett något begränsat användargränssnitt.

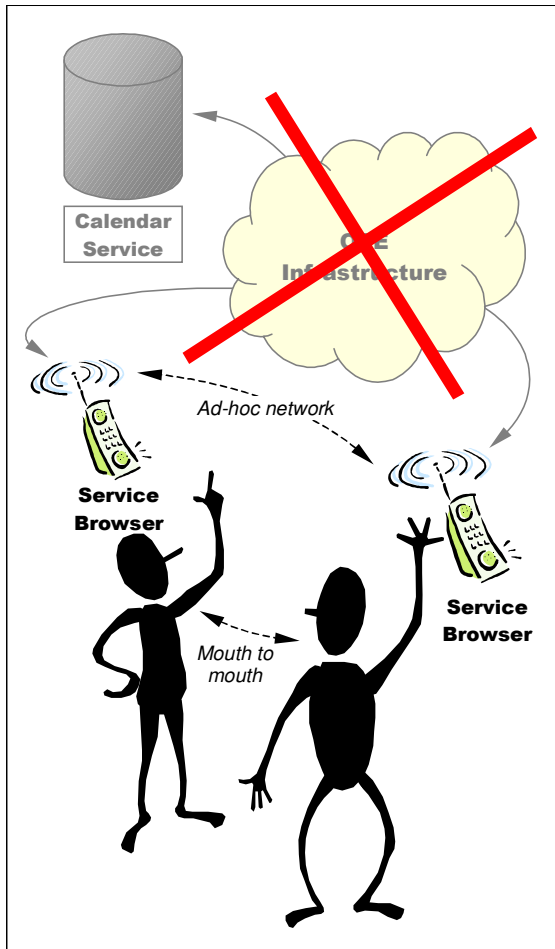
Ovanstående vision illustrerar hur interaktionen mellan användare och tjänst, förmedlad via en apparat x, kan fortsätta utan förlust av interaktionsinformation, trots att

interaktionsmedium byts till en apparat y. Vi kallar detta för apparatberoende, och utmaningen i att åstadkomma det består av två delar, nämligen:

- vilket stöd krävs från tjänstemiljön för att det ska gå att utveckla tjänster som är oberoende av den typ av apparat som används vid interaktionen, och
- hur kan interaktionsinformationen isoleras som ett led i att göra interaktionen flyttbar från apparat till apparat, utan att försvåra utvecklingen av enskilda tjänster.

På dagens World Wide Web är interaktionen med de flesta tjänsterna begränsad till apparater med en högupplösande färgskärm, ett tangentbord och en mus. Om en tjänst på webben stödjer fler än en apparattyp så kräver varje apparattyp ett eget användargränssnitt. Ett exempel på en sådan tjänst är Expressen som för närvarande stödjer interaktion via traditionellt webbgränssnitt, 3COMs PalmPilot, Windows CE-apparater och Nokias 9110 Communicator. För att det ska gå att använda Expressens tjänst från dessa apparater så krävs ett (för tjänsten) särskilt utvecklat gränssnitt för varje apparat. Om en ny typ av utrustning introduceras (exempelvis en mobiltelefon som stödjer WAP) så är risken stor att ytterligare ett nytt användargränssnitt måste utvecklas. Om tjänster kunde utvecklas oberoende av den utrustning som de först varit avsedda för, så skulle fler tjänster kunna göras tillgängliga till fler användare med mindre besvär än idag.

Scenario 3: kalendersynkronisering



Figur 5. Kalendersynkronisering

I kalenderscenariot ovan kunde de två kollegorna ha synkroniserat sina kalendrar även utan Bluetooth-uppkopplingen, om de bara kunnat komma åt sina lokala kopior av sina respektive kalendrar. Detta skulle tillåta dem att komma fram till en lämplig tid för ett möte på det gamla pålitliga mun-till-mun-viset.

Ovanstående scenario demonstrerar vikten av att göra det möjligt att utveckla tjänster som inte är helt beroende av en fungerande nätverksförbindelse för att fungera. De flesta tjänster som förmedlas på elektronisk väg är av naturen sådana att det är möjligt att utveckla åtminstone delar av funktionaliteten till att bli oberoende av nätverksuppkoppling. Detta är sant för kalendertjänsten i scenariot ovan, men även tjänster för kommunikation (t.ex. e-post), underhållningstjänster, och till och med tjänster för elektronisk handel innehåller funktionalitet som kan användas 'off-line'. Detta kräver dock att tjänstemiljön tillhandahåller medel och metoder för att köra delar av tjänsten lokalt på de apparater som används för interaktion med tjänsterna. Enskilda tjänster kan även dra nytta av stöd med avancerad cache-hantering av tjänstespecifik information, synkronisering, och annan generell funktionalitet som tjänstemiljön kan bidra med.

Till en början kan det vara svårt att se vad det är som är nytt med kalendersynkroniseringen, förutom möjligtvis användandet av WAP-telefoner och ett ad-hoc-nätverk som en reservlösning när den normala infrastrukturen inte är tillgänglig; kalendertjänster med liknande funktionalitet existerar på marknaden redan idag. Dock, om de erbjuder den beskrivna typen av nätverksberoende med möjlighet att titta i kalendern off-line, så är de inte utvecklade för en OSE och därmed saknar de fördelarna med att tillhöra en öppen miljö. Å

Två kollegor väntar på ett flyg på en flygplats i ett främmande land. De har varit på en konferens tillsammans och de bestämmer sig för att boka ett möte för att jämföra anteckningar. De bär båda med sig sina WAP-telefoner för att kunna komma åt sitt företags kalendertjänst.

Under normala omständigheter så skulle detta göra det enkelt för dem att synkronisera sina kalendrar för att hitta en lämplig tid i deras upptagna scheman. Tyvärr saknar deras mobiltelefonoperatör avtal för 'roaming' med samtliga operatörer i det land de är i, så de kan inte koppla upp sina telefoner till kalendertjänsten.

Som tur är så är deras telefoner även utrustade med Bluetooth för trådlös kommunikation över korta avstånd. Dessutom så är den kalendertjänst som de använder så pass smart att den fungerar (åtminstone delvis) trots att de inte har kontakt företagets kalender-server. Detta tillåter dem att göra en lokal synkronisering av sina kalendrar.

andra sidan, de existerande kalendertjänster som är utvecklade för webbaserad användning, som t.ex. 'Netscape Netcenter's Calendar' för WWW, kan inte göras oberoende av nätverksförbindelse.

Om man studerar nätverksberoende så upptäcker man snart att problemet är större en enbart frågan om det existerar en fungerande nätverksförbindelse eller inte. Det är t.ex. önskvärt att tjänstemiljön stödjer andra nätverkstopologier än TCP/IP, som är det självklara valet idag; ett ad-hoc-nätverk som Bluetooth är ett exempel på ett sådant nätverk. En annan fråga är kvaliteten på förbindelsen som används. Tjänster som förmedlas via WWW kan inte på något sätt ta reda på, och därmed inte heller anpassa sig till, kvaliteten på den nätverksförbindelsen som används för att förmedla interaktionen till användaren. Detta trots det faktum att kapaciteten under normala omständigheter kan variera från 9600 baud när en GSM-telefon används till 100 Mbit/sek eller mer över ett LAN.

Utmaningen som kan identifieras i detta scenario är därmed att skapa den infrastruktur som krävs för att utveckla tjänster till en OSE; tjänster som är oberoende av nätverksuppkoppling och topologi, men samtidigt medvetna om uppkopplingens kvalitet.

Kontraktmodeller för tjänstesamverkan

Ett modell för att beskriva samarbete mellan agenter är som om interaktionen reglerades av kontrakt. Kontraktmodellen kan användas både för att beskriva helt styrda samarbeten, som till exempel ett vanligt kommunikationsprotokoll, och mer komplicerade samarbeten där agenterna först kommer överens om hur de ska samarbeta, och sedan utför en gemensam uppgift. Det finns också andra sätt att beskriva samarbeten mellan agenter. Ett alternativt modell är att se agenter som tänkande varelser, som styrs av sina egna mål och intentioner. Problemet med en sådan modell är att alla agenter måste gå att beskriva på det sättet. Det blir onaturligt för mycket enkla agenter. Kontraktmodellen passar dessutom väldigt bra för öppna tjänstarkitekturer, där man ser varje agent som 'ägd' av någon människa eller mänsklig organisation. Ägaren kan då bestämma över sin agent genom att begränsa vilka kontrakt den får delta i.

På SICS har vi studerat agentkontrakt på två olika sätt: dels har vi tittat på kontraktförhandling i agentorienterade marknadsplatser (detta beskrivs i mer detalj i avsnittet 'en agentbaserad marknadsplats'), och dels har vi tittat mer i detalj på vad ett tjänstekontrakt ska kunna beskriva [Waern 1998]. På hög nivå kan man urskilja fem olika saker som ett kontrakt måste uttrycka:

- Vad vill varje agent ha ut av samarbetet?
- Vilka kompetenser använder agenterna i samarbetet?
- Finns det information som agenterna måste utbyta för att samarbetet ska fungera?
- Hur går samarbetet till i praktiken, vem gör vad när?
- Vilket språk och vilka termer använder agenterna när de utbyter information?

Ibland måste agenter prata explicit om dessa saker när de sätter upp ett kontrakt. En sådan kontraktförhandling sker i så fall inom ramarna för ett överordnat kontrakt. Men oftast förhandlar agenter bara om en del av sakerna - resten är redan bestämt inom ramarna för det överordnade kontraktet, eller helt enkelt i hur agenterna är programmerade.

I helt öppna agentarkitekturer kan vi inte förutsätta någonting om hur kontrakt fungerar i praktiken. En del agenter kommer inte alls att kunna prata med varandra. Andra kommer att tro att de kan ingå ett kontrakt, men misslyckas, till exempel för att programmerarna inte följer någon viss standard. Det kommer också att finnas agenter som 'ljuger' - som går med på kontrakt som de senare bryter. Virus är exempel på det sista. Vad vi har gjort är att vi har analyserat vissa typer av samarbetsituationer och beskrivit de kontrakt som kan behövas i

sådana situationer. Ett sådant exempel är kontrakt för användaranpassningar [Bylund och Waern 1998].

Resultat

I detta avsnitt presenterar vi kortfattat några av våra resultat.

SicsDAIS—SICS Dynamic Agent Interaction System

SicsDAIS-systemet utvecklades som en del i KIMSAC-projektet [Charlton et al. 1997], vars syfte var att tillhandahålla användaranpassad samhällsinformation genom publika kiosker. sicsDAIS [Espinoza 1998, Espinoza under utgivning] (Espinoza, 1998; Espinoza, kommande) är konstruerad som en öppen och flexibel komponent för interaktion med i huvudsak grafiskt interaktiva tjänster. En tjänst är i detta sammanhang en mjukvaruagent som tillhandahåller sin tjänst genom en grafisk delkomponent som i sicsDAIS kallas för *content handler*. *Content handlers* påminner om tjänste-*proxies* i Jini, eftersom de båda förser användaren med ett sätt att interagera med en fjärran tjänst i användarens egna lokala system.

SicsDAIS hanterar presentationen av och interaktionen med content handlers. sicsDAIS koordinerar content handlers från en mångfald av tjänster och möjliggör kommunikation och delande av data mellan tjänsterna.

SicsDAIS-systemet är implementerat som en agentkomponent i Java och det kan kommunicera med andra agenter med KQML (Knowledge and Query Manipulation Language; ett agentkommunikationsspråk [Labrou och Finin 1994]) eller andra protokoll. SicsDAIS distributionen inkluderar mallar som kan användas för att konstruera content handlers.

Som en enskild komponent för interaktion med tjänster fungerar sicsDAIS väl. Systemet har också använts i denna roll i KIMSAC-projektet. SicsDAIS ger en användare en uniform kontaktpunkt gentemot agenter och tjänster och tjänsterna kan samarbeta, koordinera, och dela på data. I en sann öppen tjänstemiljö ställs det dock större krav. Följande punkter beskriver några av dessa:

- I sicsDAIS begränsas interaktionen till traditionell direktmanipulering. För många tjänster räcker detta men i vissa fall krävs det att fler modaliteter tillåts. Speciellt viktigt är det i sådana fall där interaktionen sker genom apparater som har begränsade grafiska gränssnitt såsom bärbara kalendrar eller mobiltelefoner.
- SicsDAIS tillåter inte att interaktionen delas upp i flera parallella kanaler. Man kan exempelvis inte tala med en tjänst genom en mobiltelefon samtidigt som information från tjänsten visas på en datorskärm.

Trots ovan nämnda brister har sicsDAIS visat sig vara ett bra första steg i utvecklingen mot ett interaktionssystem för en öppen tjänstemiljö.

ConCall: Conference and workshop call reminder service

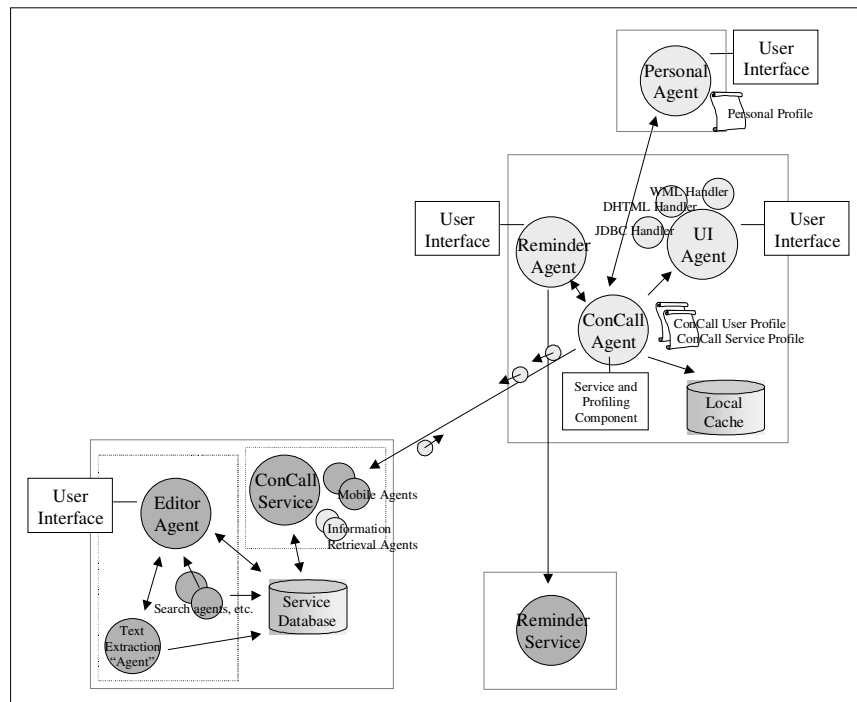
I ett parallellprojekt till IntAge, EdInfo, har vi utvecklat ett exempel på en 'halvt' öppen tjänstemiljö. Huvudmålet för EdInfo-projektet är att skapa en informationstjänst i vilken informationsredaktörer samarbetar med abonnenter för att skapa användaranpassade informationstjänster [Höök et al 1997]. Lösningen i detta projekt har varit att skapa en öppen tjänstemiljö för webb-baserad användaradapterad information med en agentteknologi.

ConCall [Waern et al 1998] är ett agentbaserat system som implementerar idéerna från EdInfo. Systemet stödjer samling, filtrering, och läsning av konferens- och workshop-kallelser, men skulle också kunna användas för kallelser till seminarier, kurser, etc. ConCall-tjänsten nås via webben.

I ConCall kan en användare (en enskild forskare t ex) läsa kallelser och ställa in påminnelser för deadlines. Påminnelser kan fås via e-post eller SMS (Short Message Service) via mobiltelefon, eller kan de läggas in en gruppkalender. För att undvika ointressanta kallelser kan användaren ställa in en användarprofil, en sorts filter, som gör att systemet hämtar ett individuellt axplock av kallelser och organiserar dem enligt användarens preferenser. Denna profil upprätthålls genom en halvautomatisk metod. Användaren har flera möjligheter att ändra i sin profil varav ett är genom att anta en profil som föreslås av systemet och som har skapats av systemet som resultat av användarens interaktion. ConCall erbjuder också användaren att söka i hela databasen av kallelser, som ett alternativt användningssätt.

ConCall-systemet

I ConCall valde vi att realisera en öppen tjänstemiljö med hjälp av teknik för så kallade *mobila agenter*. Mobila agenter är en typ av dynamiska objekt som gör det möjligt att skapa 'kopior' av tjänster och flytta runt dem i ett nätverk. Vi använde tekniken för att skapa personliga instanser av gerella tjänster, och flytta dessa till användarens egen utrustning. En Java-baserad plattform för mobila agenter har använts (den OMG MASIF kompatibla *Grasshopper*-plattformen från IKV++). I projektet utvecklade vi en egen händelsehanterings- och meddelandearkitektur ovanpå denna. KQML valdes som kommunikationsspråk mellan agenter. Detta agentkommunikationsspråk använder vi också för att koordinera händelser mellan agenter. Vidare använder vi dynamisk HTML och Java-applets i användargränssnittet samt JDBC för att hämta informationen som lagras i mySQL-databaser. Figur 6 beskriver systemets arkitektur.



Figur 6. ConCall-arkitekturen

På användarens sida (i övre delen av figuren) har användaren tillgång till en personlig assistent som håller reda på all information som är relevant för just denna användare. Detta inkluderar vilka tjänster som han eller hon prenumererar på, vilka apparater som används, osv. Användare prenumererar på tjänster via sin personliga assistent. Vid prenumerationsstillfället tilldelas användaren de agenter från tjänstesidan som är nödvändiga för att använda sig av tjänsterna.

På tjänstesidan (nedre delen av bilden) samarbetar informationsmäklaragenter med databaser och andra informationsresurser för att kunna erbjuda användare informationstjänsterna. Dessa mäklaragenter har ett antal verktyg för informationssökning, annotering, textextrahering, textanalys, visualisering, etc, till sitt förfogande. Agenter på användarsidan samarbetar med agenter på tjänstesidan för att kunna ge användaren den behövliga informationen. Tjänsteagenterna kan också:

- Hantera speciella agenter som flyttar sig från användarsidan till tjänstesidan för att utföra uppgifter,
- Se till att agenterna som användaren använder sig av är av senaste version,
- Hantera en säkerhets-policy på serversidan,
- Vidarebefordra nya tjänster eller tjänstetillägg till användarsidan, etc.

Arbetet i ConCall-projektet är ett steg mot den öppna tjänstemiljön. Specifikt angriper det adaptiva tjänster. Ett fokus för vidare arbete i denna kontext är att undersöka vilka krav som finns för att göra det möjligt att användaranpassa mobila tjänster till användarens varierande interaktionssätt.

Arbetet med ConCall gav flera viktiga erfarenheter av teknologi för mobila agenter. Den viktigaste erfarenheten var att tekniken inte riktigt är anpassad för implementation av öppna tjänstearkitekturer. Ett problem var att de flesta plattformar endast tillåter vanliga objektanrop och inte mer generell agentkommunikation. I en öppen tjänstearkitektur är detta ett nödvändigt krav - detta var skälet till att vi var tvungna att själva implementera händelsehantering. I stället gav plattformen mycket avancerat stöd för att skapa, flytta och ta bort agenter, ett stöd som vi inte alls utnyttjade till fullo, eftersom objekten i en tjänstearkitektur rör sig på ganska enkla och förutsägbara sätt. Slutsatsen blir att tekniken för mobila agenter skapades med ett annat syfte i åtanke, det att bygga distribuerade men ändå sammanhållna tillämpningar, och därför inte uppfyller kraven på en grundteknik för öppna tjänstemiljöer.

Koordinering av automatiska anpassningar

När utvecklingen går från singulära datorsystem till OSE:er så blir uppgiften att anpassa funktionalitet, dialoger och gränssnitt till enskilda användare verkligt komplex. Det finns flera skäl till det.

- *Kort livscykel* – att individanpassa mjukvara tar tid och kräver reella insatser, antingen av användaren eller systemet. När tjänster endast används under en begränsad tid så leder detta till att tjänstespecifika individanpassningar inte kan utföras.
- *Många enheter att anpassa till* – det totala antalet tjänster som användaren kommer att stöta på är stort. Användaren kan inte förväntas utföra anpassningen av varje enskild tjänst manuellt.
- *Skydd av personlig information* – när tjänster endast används vid enstaka tillfällen så kanske användaren saknar det förtroende för tjänsten som krävs för att individuella anpassningar ska kunna utföras.
- *Samma om och om igen* – informationen som behövs för en anpassning är inte alltid tjänstespecifik. Om samma anpassning måste utföras om och om igen av olika tjänster så kommer en stor del av arbetet att bli mångdubblat.
- *Distribution av information* – informationen som styr och kontrollerar en anpassning i en OSE kommer att distribueras mellan ett flertal aktörer (inkl. användaren). Därför behöver anpassningsprocessen koordineras.

För att kunna hantera dessa frågor så föreslår vi införandet av en *anpassningskoordinator* (*Adaptation Coordinator*) som en fristående aktör i processen att utföra individanpassningar i

OSE:er. Detta tillåter att delar av anpassningsprocessen delegeras från enskilda tjänster till anpassningskoordinatören. Fördelar med att göra detta inkluderar följande.

- Delegeringen kan minska tjänsternas behov av att lära sig, representera och resonera om dess användare, en potentiellt väldigt kostsam process.
- Användarmodeller kan baseras på interaktionen mellan användaren och fler än en tjänst, vilket kan leda till rikare modeller än annars.
- Eftersom information om användaren delas mellan flera tjänster så behöver användaren inte mata in samma information flera gånger.
- Information som är viktig för anpassningsprocessen, men som användaren tycker är känslig, kan skyddas från enskilda tjänster.

Inom KIMSAC-projektet utvecklade vi en enkel anpassningskoordinator [Bylund 1999, Bylund och Waern 1998]. Integrationen av en anpassningskoordinator i en verkligt öppen miljö återstår att studera. Detta gäller i synnerhet kraven på förhandling av anpassningskoordination under systemdrift.

Förteckning över bilagor inom området 'Gränssnitt emot öppna tjänstemiljöer'

- [1] Bylund, M. (1999). Coordinating Adaptations in Open Service Architectures. M.Sc. thesis, Uppsala University, Uppsala.
- [2] Markus Bylund and Annika Waern. Service Contracts: Coordination of User-Adaptation in Open Service Architectures. Journal of Personal Technologies, Volume 2, Issue 4, 1998.
- [3] Espinoza, Fredrik (1998) sicsDAIS: Managing User Interaction with Multiple Agents, Licentiate thesis at the department of Computer and Systems Sciences, October, 1998.
- [4] Espinoza, F. (to appear). sicsDAIS: A Multi-Agent Interaction System for the Internet. Paper presented at the WebNet 99-World Conference on the WWW and Internet, Hawaii.
- [5] Annika Wærn. Service Contract Negotiation - Agent-Based Support for Open Service Environments. In Proc. 1998 workshop on Distributed Artificial Intelligence, at the 4th Australian conference on Artificial Intelligence (AI'98), Brisbane, Australia, July 1998. Revised version in 'Multi-Agent Systems: Theories, Languages and Applications'. ed. C. Zhang and D. Lukose, Lecture notes in Artificial Intelligence 1544, 1998.
- [6] Annika Waern, Mark Tierney, Åsa Rudström and Jarmo Laaksolahti. ConCall: An information service for researchers based on EdInfo. SICS Internal Report T98:04, 1998.

En Öppen Elektronisk Marknadsplats

Sverker Janson, Joakim Eriksson och Niclas Finne

Sammanfattning

Internet beskrivs ofta som en elektronisk marknadsplats, och visst kan vi se framför oss hur vi vandrar mellan försäljarna på nätet och tittar på deras utbud. Men där stannar likheten. Nätet är inte ett litet marknadstorg. All världens handel skall så småningom ske på detta torg och det är en övermänsklig uppgift att vare sig som säljare eller köpare överblicka dess helhet.

För att dra full nytta av nätet behöver vi avancerade verktyg som hjälper oss att automatiskt överblicka, utvärdera och agera på stora mängder komplex information. Det mest lönsamma sättet att tillhandahålla sådana verktyg är idag att erbjuda dem i form av tjänster, portaler och auktionstjänster, som kan ägas och kontrolleras av en stark aktör. Men denna utveckling leder inte till en effektiv global marknad där alla kan dra nytta av nätet på lika villkor.

Vi har i denna aktivitet studerat principer för utformningen av en öppen, agent-baserad, marknadsinfrastruktur, som gör det lätt för alla att automatisera sin verksamhet på en global marknad. Vi har tagit fram en grundläggande modell och prototyper som visar att modellen är praktiskt tillämpbar.

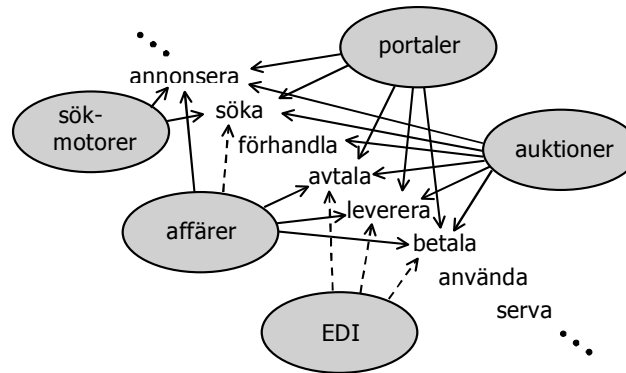
I de följande avsnitten ger vi en översikt av våra resultat. De beskrivs mer utförligt i ett antal bifogade artiklar [Rasmusson och Janson 1999, Eriksson et al. 1996a, 1996b, 1997, 1998, 1999, Eriksson och Finne 1997], vilka även ger referenser till annan närliggande forskning. Avslutningsvis pekar vi ut möjliga inriktningar för vidare arbete för att skapa större flexibilitet, öppenhet och utvecklingsbarhet i en tänkt framtida elektronisk marknadsinfrastruktur.

Elektronisk handel

De nya kommunikationsmöjligheter som Internet gett oss har skapat förhoppningar om en global perfekt marknad, som effektivt allokerar varor och resurser efter tillgång och efterfrågan. Men vi klarar inte som människor av att tillräckligt snabbt överblicka, utvärdera och agera på de stora mängder information som omsätts i en global marknad. En sådan marknad kräver automation.

Handel omfattar aktiviteter som annonsering, sökning, förhandling, avtalsslut, leverans, betalning, användning och service (se figur 7). Vi fokuserar i detta arbete på annonsering, sökning, förhandling och avtalsslut, de aktiviteter som utgör kärnan i en marknad. Hur kan de automatiseras?

Webben i kombination med sökmotorer stöder global annonsering och sökning, men med mycket dålig precision. Information på nätet är avsedd för mänskliga ögon och kan ligga dold i bilder och de flesta behov låter sig inte fångas i några enkla sökord. Webben ger heller inte någon grund för sökning med specifika kriterier – t ex den billigaste bilen som uppfyller vissa villkor. Utveckling av metadata, t ex baserat på XML, kan förbättra precisionen. Det krävs att marknadens aktörer kan enas om standarder för att representera sådan information och att de kan se att det ligger i deras intresse att tillhandahålla information på detta sätt.



Figur 7. Elektronisk handel

Vissa framsteg kan göras även utan metadata. Ett flertal tjänster på nätet sammanställer information från ett antal försäljningsställen till en sammanslagen "katalog". Affärsmodellen för sådana prisjämförelsetjänster är att leda in kunder till köp och att få betalt av anslutna företag. De går en balansgång mellan att tillhandahålla ett värde för sina användare och att inte motverka intressen för de anslutna företagen. Jämförelser sker ur ett "neutralt" perspektiv. Om informationen var standardiserad skulle det vara en enkel uppgift att göra sådana jämförelser och de skulle inte behöva utföras av mellanhänder med provision. Varje aktör kunde själv, med enkla program, utföra sådana jämförelser ur sitt eget perspektiv.

Andra tjänster, t ex auktionstjänster, tillhandahåller även förhandlingsmekanismer. Om denna trend att utöka funktionaliteten i tjänster tas längre kommer enstaka tjänster att stödja hela marknader, globalt åtkomliga men centraliserat ägda och kontrollerade marknadsplatser. Detta strider helt klart emot idealet om fria marknader.

Men automatiserade marknader behöver inte stödjas av centraliserade tjänster. Den utvecklingen är en följd av att det är lätt att tjäna pengar på att äga och kontrollera en tjänst. Det är också möjligt att automatisera handel på ett decentraliserat sätt, där vi kan agera som självständiga aktörer och samverka direkt över Internet utan onödiga mellanhänder.

Byggstenar i en agent-baserad marknadsinfrastruktur

Vi har studerat principer för utformningen av en agent-baserad marknadsinfrastruktur, som gör det lika lätt för alla att automatisera sin verksamhet. Varje aktör representeras av agenter, program de själva äger och kontrollerar, inte av en central tjänst ägd och kontrollerad av en annan aktör. Aktörernas agenter kommunicerar med varandra över nätet. Målet för marknadsinfrastrukturen är att tillhandahålla minsta möjliga gemensam nämnare som möjliggör enkel automation av marknadsaktiviteter. Aktörerna skall vara fria att ta tillvara sina egna intressen och fria att anta alla olika roller på en marknad, köpare, säljare och olika typer av mellanhänder.

Den gemensamma nämnaren vi föreslår är ett gemensamt språk för handelskommunikation, i vilket meddelanden utväxlas mellan agenter på i princip samma sätt som elektronisk post mellan människor. Språket har två komponenter: ett intressespråk och ett interaktionsspråk.

Intressespråket MIF (Market Interest Format) bygger på strukturerade dokument som representerar mängder av kontrakt. Ett intresse beskriver den mängd av kontrakt som en aktör säger sig vara (potentiellt) intresserad av att ingå.

Interaktionsspråket MIL (Market Interaction Language) tillhandahåller ett grundläggande ordförråd för annonsering, sökning, förhandling och avtalsslut. Meddelanden i MIL uttalar sig om intressen: "detta är mitt intresse", "har du ett intresse som överlappar med mitt?", "detta intresse är ett anbud", "jag accepterar detta alternativ i ditt anbud", osv.

MIL och MIF är formella språk; här ges för enkelhets skull tolkningar på svenska. En detaljerad specifikation återfinns i [Eriksson et al. 1999].

Exempel: Meddelanden i MIL/MIF.

Man kan i MIL/MIF uttrycka meddelanden av typen:

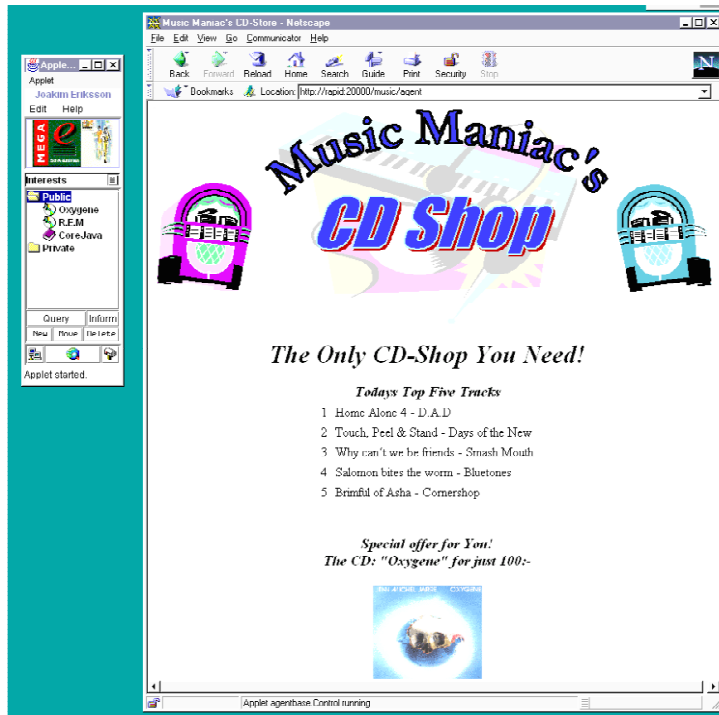
- Jag önskar köpa en röd Mazda-626 kombi –98.
- Pelle vill sälja pennor för mindre än 10 kr.
- Åsa vill köpa en pizza inom en halvtimme.
- Jag köper penna #42 för 6.50 kr.

och så vidare.

Exempel: Dialog i MIL/MIF.

Följande exempeldialog mellan agenter kan uttryckas i MIL/MIF och kan hanteras automatiskt av de inblandade parternas agenter:

- A frågar (genom sin agent) en mäklartjänst om adresser till agenter som annonserat intresse av att sälja kylskåp
- Mäklartjänsten svarar att B och C vill sälja kylskåp
- A ber B och C om offerter på Electrolux 3117B
- B offererar ett Electrolux 3117B för 3000 kr
- C offererar ett Electrolux 3117B för 2500 kr
- A informerar B om Cs bättre offert
- B matchar Cs erbjudande
- A accepterar Bs nya offert och tackar nej till Cs



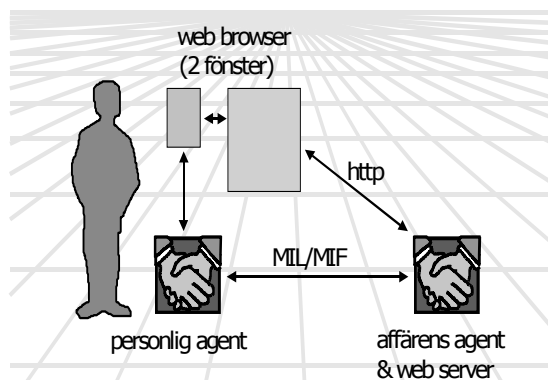
Figur 8. Den personliga assistenten WebButler (lilla fönstret) intill ett fönster med en besökt CD-affär stödd av MerchantAssist.

En prototyp till en agent-baserad marknadsinfrastruktur

MarketSpace – agent-stödd Internet-handel

Baserat på MIL/MIF har vi utvecklat en prototyp för agent-baserad handel av konsumentvaror på Internet i samarbete med Telia Research AB. Den består av:

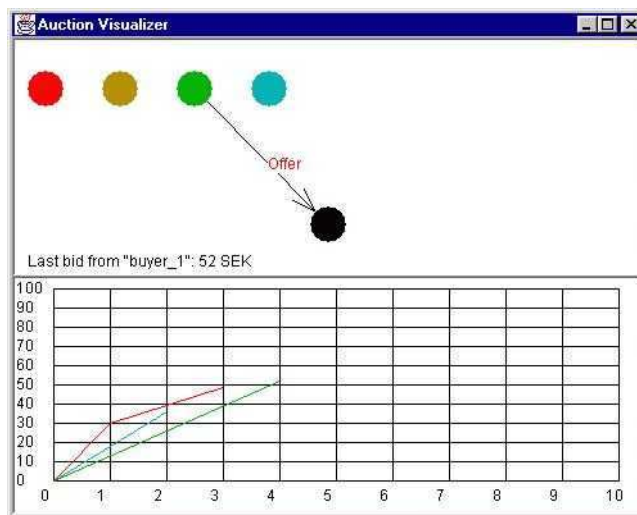
- En personlig-assistent-agent (WebButler) som låter användaren beskriva intressen, interagera med andra agenter och dra nytta av agent-förstärkta tjänster (affärer)
- En handels-assistent-agent (MerchantAssist) som bl a utnyttjar kommunikation med kundernas agenter till att anpassa web-presentationen till kundernas intressen (se figur 8).
- En katalogtjänst (AgentFind) som låter kunder och försäljare, via sina agenter, registrera sina intressen och hjälper dem hitta aktörer med matchande intressen.
- En tjänst för att beskriva intressen. Intressen sänds till och sparas i den personliga agenten.



Figur 9. Integrering av web-baserad handel med agenter

Agent-stödet i MarketSpace är ett komplement till, och är integrerat med, vanlig web-baserad Internet-handel. Kommunikationen mellan agenter, web-läsare och web-server sätts upp så att den personliga agenten ser vilka sidor som besöks och kan interagera med den besökta tjänstens/affärens agent.

Användare kan också delegera uppgifter till agenten som då utförs helt automatiskt, t ex att sammanställa prisuppgifter eller t o m att köpa den billigaste varan som motsvarar ett visst intresse.



Figur 10. Ett auktions-exempel som följer med AgentBase-verktygslådan

AgentBase – en verktygslåda för utveckling av agent-baserade system

Ett sidoresultat från utvecklingen av MarketSpace-prototypen är verktygslådan AgentBase, för implementering av agent-baserade system i Java och i Prolog. AgentBase har används vid utvecklingen av två prototypsystem i samarbete med Telia Research.

- 1) En agent-baserad marknadsinfrastruktur för varor och tjänster, som kompletterar och är integrerad med web-baserad handel (MarketSpace)
- 2) Ett agent-baserad workflow-system som stöder dynamisk generering av arbetsflöden från en marknad av tillgängliga aktiviteter och kompetenser (AMIS)

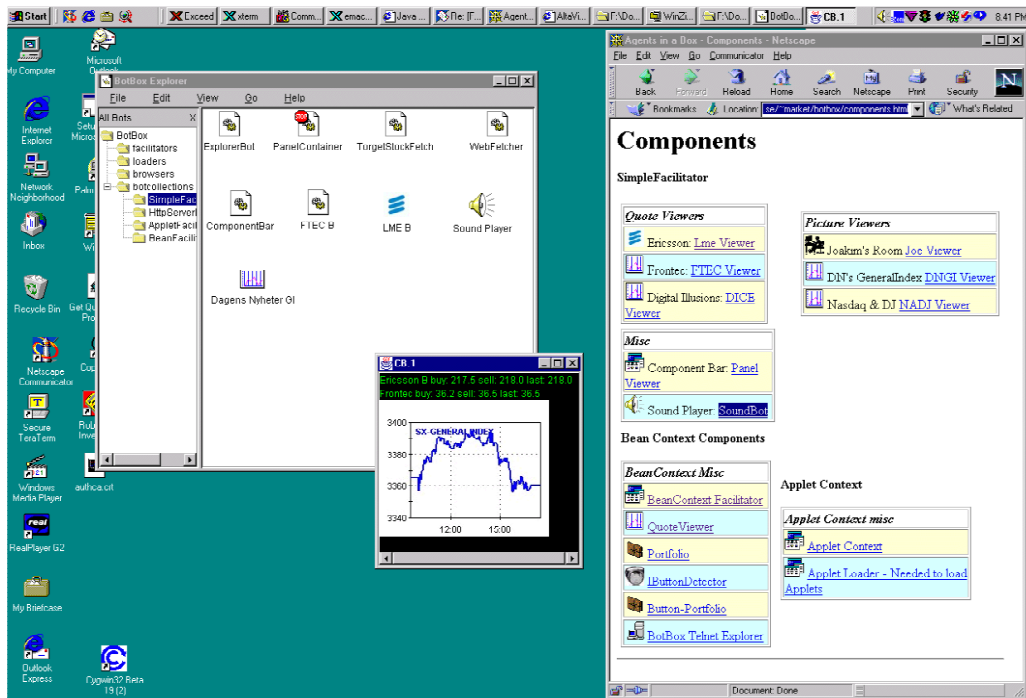


Figure 11. The BotBox Internet service platform.

BotBox – en plattform för Internet-tjänster

Ytterligare ett sidoresultat från utvecklingen av MarketSpace är en generell arkitektur och plattform för Internet-tjänster som körs lokalt på användarnas egna datorer, varav personliga agenter är ett special-fall. BotBox (figur 11) gör det lätt att installera och hantera en personlig uppsättning tjänster.

Slutsatser och framtida inriktning

Vi har i vårt arbete studerat principer för agent-baserade marknadsinfrastrukturer som gör det möjligt för alla, stora som små aktörer, att utnyttja nätets och automatiseringens fördelar, som idag endast tillkommer stora aktörer. Vi har i detta första skede starkt betonat enkelhet vilket lämnar utrymme för utvidgningar och förfiningar i ett flertal riktningar. Vi kommer i vårt fortsatta arbete att närmare studera följande delproblem.

Intressen med preferenser

Intressespråket MIF gör det möjligt att uttrycka intressen som mängder av (potentiellt intressanta) kontrakt. Inga preferenser mellan kontrakt kan anges. Detta är ett medvetet designval i detta skede eftersom intressen inte skall behöva avslöja mer än vad som behövs för att hitta andra aktörer med motsvarande intressen. Preferenser behöver då inte kommuniceras till andra aktörer.

Men för att kunna delegera förhandling till en mellanhand, t ex en mäklare, vill vi kunna uttrycka våra prioriteringar. Vi avser därför att studera representationer vilka kan uttrycka varierande grad av utilitet eller säkerhet (certainty) i intresset.

Öppenhet för interaktionsprotokoll

Interaktionsspråket MIL tillhandahåller en grundläggande vokabulär för interaktion i en marknad. Men protokollen (t ex olika auktionsformer) måste vara kända av agenterna i förväg och MIL kan inte heller uttrycka varje tänkbart meddelande i marknadsinteraktion. Det är önskvärt att dels kunna uppdatera protokoll helt dynamiskt, dels kunna införa nya protokoll med nya egenskaper.

Vi avser att studera ett generellare ramverk där protokoll kan utväxlas mellan agenter. Problemet blir dels att kunna hantera protokoll som dynamiska komponenter, dels att kunna lita på att protokoll korrekt tar tillvara egna intressen.

Öppenhet för intresseformat

Givet att arkitekturen utvidgas till att kunna hantera dynamiska komponenter, kan intressespråket MIF generaliseras till en intresse-gränssnitts-standard. Dynamiska komponenter kan införa ny funktionalitet. Vi avser att längre fram studera även denna generalisering.

Förteckning över bilagor inom området 'Öppen elektronisk marknadsplats'

- [1] Lars Rasmusson and Sverker Janson. Agents, self-interest and electronic markets. Knowledge Engineering Review, Vol. 13:2, 1999, 1-8.
- [2] Joakim Eriksson, Niclas Finne, and Sverker Janson. Information and interaction in MarketSpace - towards an open agent-based market infrastructure. In Second USENIX Workshop on Electronic Commerce, USENIX, Nov 1996a.
- [3] Joakim Eriksson, Niclas Finne, and Sverker Janson. Information and interaction in MarketSpace and their implementation in Prolog. Paper presented at LPNET'96 (1st Workshop on Logic Programming Tools for Internet Applications), ICLP'96, Sept 1996b.
- [4] Joakim Eriksson, Niclas Finne, and Sverker Janson. Surfing the market and making sense of the web. Paper presented at Programming the Web -- In Search of APIs, WWW5, Paris, May 1996c.
- [5] Joakim Eriksson and Niclas Finne. MarketSpace: an open agent-based market infrastructure. Master's Thesis. Computing Science Department, Uppsala University and Swedish Institute of Computer Science, 1997.
- [6] Joakim Eriksson, Fredrik Espinoza, Niclas Finne, Fredrik Holmgren, Sverker Janson, Niklas Kaltea, and Olle Olsson. An Internet software platform based on SICStus Prolog. Paper presented at Logic Programming and the Web, WWW5, San Jose, 1997.
- [7] Joakim Eriksson, Niclas Finne, and Sverker Janson. To each and everyone an agent: augmenting web-based commerce with agents. In Proceedings of the International Workshop on Intelligent Agents on the Internet and Web, World Congress on Expert Systems, Mexico City, March 1998.
- [8] Joakim Eriksson, Niclas Finne, and Sverker Janson. SICS MarketSpace: an agent-based market infrastructure. In First International Workshop on Agent-Mediated Electronic Trading, AMET-98, Selected Papers, Lecture Notes in Computer Science 1571, Springer-Verlag, 1999.

Säker internethandel med autonoma agenter

Andreas Rasmusson, Lars Rasmusson och Sverker Jansson

Vi har studerat hur internet handel på öppna Internet-liknande marknader kan göras säkrare för agenter (personer eller program). I de följande avsnitten ger vi en översikt av våra resultat. En utförligare beskrivning av resultaten och deras relation till annan pågående forskning finns i de bifogade artiklarna [L Rasmusson 1996, A. Rasmusson 1996, L Rasmusson och Jansson 1996, A Rasmusson och Jansson 1996, Rasmusson et al 1997].

I vår vardag kommer vi göra många, ofta små, mer eller mindre automatiserade transaktioner. Agenter, våra eller andras, kan köpa och filtrera information åt oss, ibland även försöka förekomma våra önskningar genom att leta reda på information vi ännu inte vet att vi vill ha. Vi behöver kunna känna oss trygga att system som hanterar personlig information eller gör ekonomiska transaktioner är stabila i avseendet att friheten att lägga till nya tjänster inte gör det lätt att utföra bedrägerier eller informationsstölder.

På en global marknad som Internet finns det många personer som vi skulle vilja kunna göra affärer med. Men hur vet vi att den andra parten inte försöker lura oss? I samhället förlitar vi oss vanligtvis på att rättsväsendet kan hjälpa oss om ett kontraktsbrott sker, men på internet är det inte lika enkelt. Öppenheten i systemet gör det mycket svårt att med säkerhet spåra vilken person som ligger bakom en viss identitet.

Säkerhetsproblematiken i öppna marknader skiljer sig en hel del från traditionell datasäkerhet där man främst försökt hindra information att hamna i orätta händer. På en öppen marknad däremot, *vill* man att agenter och personer skall samverka. Det man vill hindra är istället att informationen missbrukas av andra. Frågeställningar som blir aktuella är:

- Förtroende för andra agenter. Eftersom pengar är inblandade finns det motiv för personer att manipulera de andra deltagarna. Det stora antalet deltagare gör dessutom lurendrejerier med låg avkastning per person lönsamma. Den centrala frågan blir därför hur agenterna skall kunna avgöra vilket förtroende de kan ha för andra agenter.

Om det är alltför svårt att finna pålitliga affärskontakter kan det mycket väl bli så att endast väletablerade företag blir de som kan dra nytta av den globala marknaden.

- Skydd av privat information. Om andra agenter skall kunna hjälpa oss på något meningsfullt sätt behöver vi kunna ge dem möjlighet att använda privat information som t.ex favoritresmål eller musiksmak. Detta medför en risk att ett virus eller illasinnad agent stjälar, manipulerar eller förstör den privata informationen.

Eftersom vi *vill* kunna dra nytta av programmen behövs det stöd för att se till att informationen inte missbrukas.

- Önskat illasinnade agenter. Vi förväntar oss att de agenter som företräder oss skall agera rationellt, dvs nyttomaximerande för oss. Ett ineffektivt program som löser en uppgift med stor resursförbrukning kan med denna syn upplevas som illasinnat, även om dess skapare aldrig avsåg detta. Faktum är att det till och med kan ses som rationellt att begå brott om det inte finns någon risk inblandad¹⁵.
- Vem skall ställas till svars? Om ett säkerhetssystem byggs utifrån att det hela tiden skall gå att ställa någon person till svars för agenternas handlingar räcker det med att en enda agent lyckas ta sig runt systemet för den skall kunna orsaka stor skada. I praktiken

¹⁵ Idén om brott som rationellt beteende har behandlats i ett nummer av Economic Times [1995].

är det omöjligt att garantera att vem som helst, världen över, är den den utger sig för att vara.

Även om ett program och en person kan identifieras som orsakande av en skada är det inte enkelt att avgöra vem som ytterst är skyldig. Programmet kan innehålla fel eller blivit avsiktligt eller oavsiktligt manipulerat av ett annat program.

- Vad är en identitet? Till skillnad från det vanliga samhället är det väldigt lätt att vara anonym på Internet. Det går alltid att skaffa sig en ny identitet någon annanstans eller, om man är ett program, skapa en kopia av sig.

Varför skulle då någon vilja ha något att göra med ett program eller en person vars intentioner det inte går att veta någonting om? Svaret är att man förstås hoppas att man själv skall ha nytta av programmet eller varan.

Ansatsen vi utforskat är man inte förbjuder en okänd agent att komma åt personlig information förrän den betar sig på ett oförväntat eller illasinnat sätt. Det går inte att helt komma ifrån det risk-moment som finns i att ge okända agenter rättigheter att använda information som de potentiellt kan missbruka. Målet är istället att nå en balans mellan säkerhet och förväntad nytta.

Vi har undersökt två sätt att hantera risker i öppna agentsystem där målet är att underlätta samverkan mellan okända parter.

- genom att samverka i ett system där förtroende och rykte är fundamentala byggstenar i systemet
- genom att övervaka okända program på en persons dator och ge beslutsstöd för användaren

Dessa metoder komplementerar varann genom att de övervakningen gör det möjligt att upptäcka brott och rykte är en mekanism för att föra tillbaka informationen till de andra agenterna i systemet.

Vi kallar den säkerhet sådana system ger för "mjuk" säkerhet. Med det menar vi att rättigheter till samverkan ges till okända agenter men risk- och nyttohänsyn tas med i bedömningen. Med "hård", traditionell, säkerhet menar vi följdaktligen metoder där beslut om tilldelade privilegier inte omutvärderas.

Nedan beskriver vi i mer detalj det arbete som vi gjort om agentsäkerhet i öppna system. Under rubriken 'social kontroll för säker internethandel' beskriver vi simuleringar av sociala kontrollmekanismer som ryktesspridning och skvaller och hur dessa påverkar sammansättningen av goda/elaka agenter i en marknad. Under rubriken 'om när det är rationellt att fuska' beskriver vi en arkitektur för en säkerhetsassistent vars uppgift är att ge användare beslutsstöd vid hantering av okända program på sin egen dator, t.ex s.k mobila agenter.

Social Kontroll för Säker Internethandel

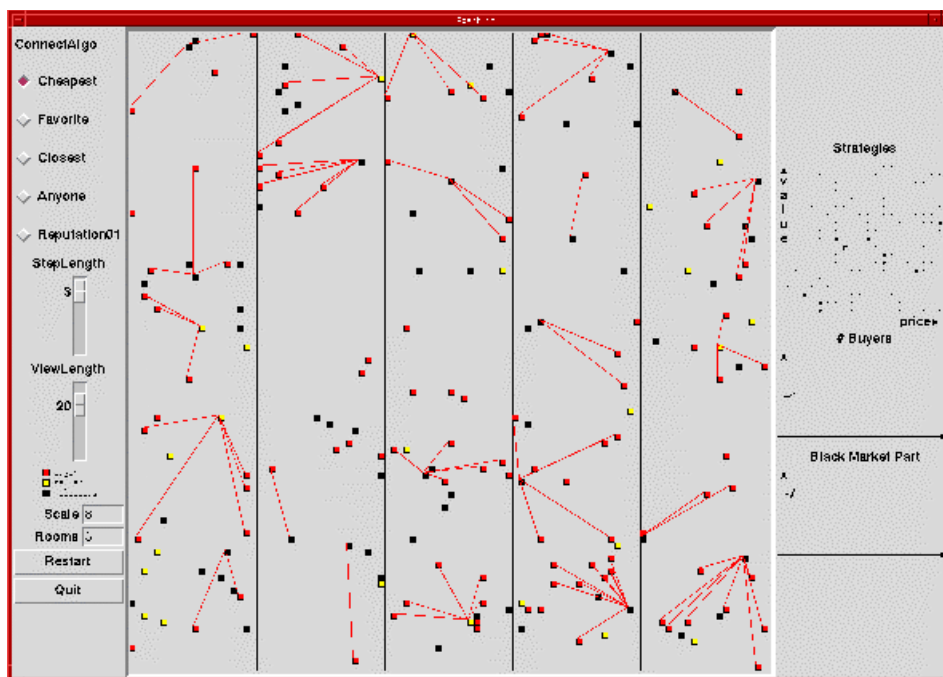


Figure 12. En simulering av en marknadsmiljö med ryktesmekanismer

I ett system med social kontroll hanteras säkerheten helt av deltagarna i systemet. Genom att undersöka de grupp beteenden som uppstår ifrån agenternas helt lokala avväganden försöker vi finna kriterier för individernas beteenden som gör det möjligt att bygga system som minskar inflytandet av agenter som identifierats som illasinnade.

Agenterna i en öppen marknad har mål som delvis är i konflikt med varann. Varje agent vill ju att varan skall säljas så dyrt/billigt som möjligt. Det finns inget entydigt mått som säger vad som är bäst i en sådan population av agenter. Dels bör marknaden vara stabil och leva länge, dels bör det finnas få illasinnade agenter eller de bör i allafall inte delta mycket i marknaden. I en effektiv marknad bör det även finnas en korrelation mellan pris och kvalitet. Ytterligare egenskaper att ta hänsyn till är hur drastiskt systemets sammansättning förändras. Under vissa parametrar går en stor andel av de "nyttiga" agenterna bankrutt innan de illasinnade agenterna slås ut.

Simuleringar gjordes på olika marknadsmodeller och med olika interna agentmodeller för resonera om incitamenten för de olika sociala mekanismerna (se figur 12).

I simuleringarna visualiserades olika parametrar för att ge en intuitiv förståelse för hur marknaden beter sig under olika förutsättningar. Målet var att undersöka vilka *kvalitativa* skillnader variation av en viss parameter ger samt att undersöka vilka parametrar som är mest signifikanta. "Sanna" optimala värden på de olika parametrarna kan förstås inte utläsas ur några rent simulerade modeller.

Om när det är rationellt att fuska

Agenter som fattar beslut om ifall de skall samarbeta eller lura varann ställs inför ett beslutsproblem som kallas "Fångens dillema". I fångens dilemma vet agenterna att båda parter tjänar mest på om båda uppfyller sitt avtal, men också att de förlorar mer på att själv, ensidigt uppfylla avtalet än om båda bryter det. Spelteoretiska analyser [Rasmusen 1989] visar att om man vet att man kommer ställas inför samma beslutsproblem med samma person igen så är det en bra strategi att samverka. Däremot, om man aldrig ses igen är det "bäst" att bryta avtalet.

I stora marknader, eller marknader där man kan vara anonym, är att chansen att man träffar på samma agent igen är liten. I sådana fall blir bedrägeri det rationella beteendet. Om vi skall kunna bygga agenter som utför komplexa uppgifter som kräver samarbete måste mekanismer byggas in i marknaden som gör att det blir rationellt att välja att samverka.

Eftersom analysen av rationellt beteende bygger på resonemang om vedergällning blir agenternas identiteter det värde de sätter på spel när det deltar i ett samarbete. Liknande fenomen kan ses idag i och med den ökade vikten som läggs vid tydliga varumärken. Varumärken "byggs upp" under en lång tid för att sedan användas som medel och argument för samarbete med andra. Det sker även att varumärken säljs mellan företag, t.ex. volvo.

Varumärken är dock inte den enda aspekt vi använder för att välja vem vi samverkar med. I den vanliga världen spelar även saker som språk, geografisk belägenhet eller personliga kontakter in. Möjligheter att bedöma varor efter andra varor än priset är viktiga för att göra det möjligt för nya agenter att kunna komma in på marknaden.

Simulering

Varje agent producerar varor med ett visst pris/kvalitetsförhållande. Produktionskostnaden för varorna beror på deras kvalitet, men säljarna kan välja vilken marginal de vill ha.

Som exempel på en annan egenskap än pris för att avgöra vem man skulle handla med delade världen delades in i zoner/rum. Agenter var endast synliga för andra agenter inom en zon. Agenternas rörelse mellan zonerna och antalet zoner varierades.

Olika strategier för att avgöra vem en agent skall handla med testades. Agenterna kunde välja t.ex. vem som helst, den närmsta, den senaste bra eller den som rekommenderades av en lokal rekommendationsagent i zonen.

Rekommendationsagenterna gör det möjligt för agenterna att sprida rykten eller skvaller mellan varandra. När agenterna kan välja vem de skall samverka med baserat på motpartens rykte skapas en marknad där identiteten blir en värdefull tillgång eftersom det knyts till värdet av framtida affärer. Detta är ett exempel på hur mjuka säkerhetsmekanismer kan användas för att minska den förväntade förtjänsten av avtalsbrott.

Några Erfarenheter

Mer komplicerade modeller är lätta att bygga. T.ex. kan agenterna göra överväganden om förtjänsten av att rapportera till rekommendationstjänsten eller där de har möjlighet att bilda karteller genom att kommunicera utanför marknaden. Alltför komplicerade modeller gör dock endast att de fundamentala beteendena i simuleringen blir svåra att utläsa. Poängen med simuleringen är i stället att lyfta fram vissa egenskaper hos öppna system på ett tydligt sätt.



Figure 13. Olika köparstrategier gör olika strategier fördelaktiga för säljarna. I den vänstra bilden ser köparna enbart till priset vilket gynnar säljare av varor med lägre kvalitet. Till höger visas prisbilden när köparna ber om rekommendationer.

Några typiska fenomen som kan ses ur simuleringen är:

- Om agenterna rör sig snabbt mellan zonerna kommer de hela tiden träffa på nya köpare och säljare. Inga av strategierna var då tillräckliga för att hindra de dåliga agenterna från att ta över marknaden.
- Om alla agenter befinner sig i ett (stort) rum och använder samma strategi uppstår oftast ett monopol. I en monopolsituation blir agenterna känsliga för om säljaren ändrar sin strategi, eftersom det inte finns några konkurrenter. Om det finns flera rum är det lättare för flera säljare att överleva. Detta står dock i konflikt med iden om Internet som en global marknadsplats och gör det inte längre möjligt att garantera ett prismässigt "optimalt" val.
- I marknader där de köpande agenterna inte har något att tjäna på att de andra köparna får ta del av information om en varas kvalitet, t.ex informationsmarknader för böcker eller musik, är exemplet med lokala rekommendationsagenter tänkbart. I de simuleringarna slogs för det mesta de dåliga säljarna ut.

En personlig säkerhetsassistent för säker internethandel

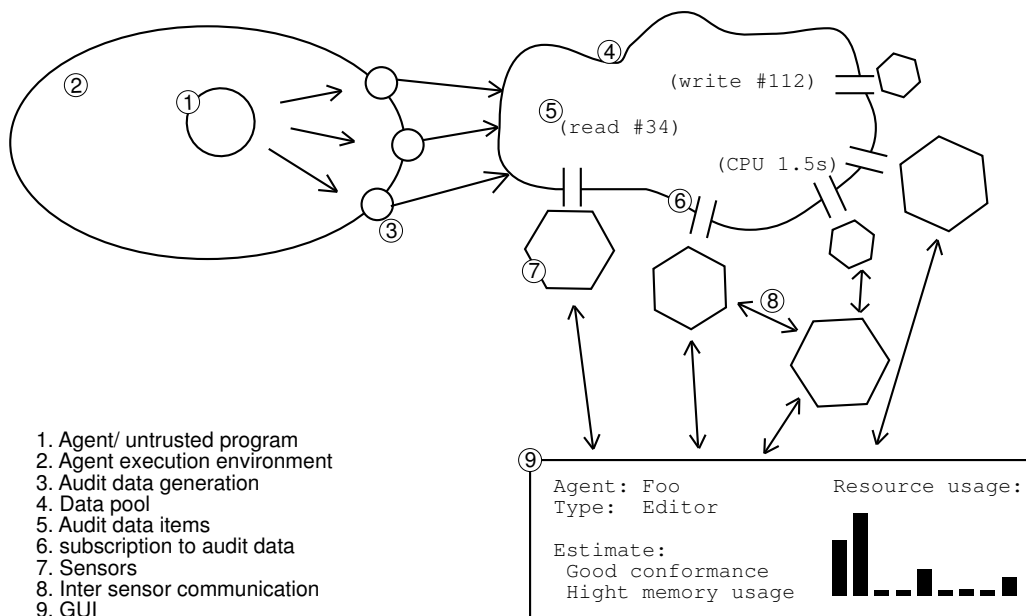


Figure 14. En systemarkitektur för en personlig säkerhetsassistent.

I en öppen agentmarknad kommer många tjänster innebära att slutanvändarna laddar hem program från ett företag. Programmen kan t.ex presentera information eller bevaka informationstjänster åt användarna. Sådana program, s.k mobila agenter, behöver komma åt information om användare för att kunna anpassa sig efter användarnas önskemål. Om inte, kommer de program som kan spridas endast kunna ha mycket knappa funktioner och vara av begränsad nytta.

Det saknas idag skalbara metoder för företag att bygga upp förtroende för att deras program inte missbrukar informationen på användarnas datorer. Den vanligaste metoden, att signera program, fungerar endast om den som signerat programmen har ett tillräckligt känt varumärke. Ett program som är signerat av en för oss okänd person ökar inte vår tilltro till programmet, oavsett antalet bitar som användes vid signeringen. Även vår tilltro till kända programleverantörer är idag i praktiken blind om vi inte kan försäkra oss om att programmen beter sig som förväntat¹⁶.

Vi har undersökt en ansats där varje användare har en personlig säkerhetsassistent som övervakar programmen och varnar för oförväntade eller farliga beteenden. Genom att utgå från programmets faktiska beteende, inte avsändarens eller någon extern certifieringsorganisations¹⁷ rykte, kan användaren väga in den förväntade nyttan av programmet när han avgör vilka rättigheter det skall ges.

Egenskaper hos assistenten som kommer vara särskilt viktiga är hur väl den kan hjälpa oerfarna användare och hur snabbt den kan anpassa sig till nya hot och nya program.

¹⁶ Både Microsoft och Netscape har på senare tid fått kritik för att deras produkter innehåller funktioner som kränker den personliga integriteten.

¹⁷ Som nämns i avsnittet *En Öppen Elektronisk Marknadsplats* är det önskvärt att söka decentraliserade lösningar för att minska risken för partiska bedömningar och fördröjningar som beror på att endast ett fåtal parter kan publicera nya tjänster.

Vad saknas i traditionella säkerhetsansatser

Traditionell datasäkerhet har gått ut på att begränsa tillgång till eller dölja information. De viktigaste ansatserna går ut på att skilja på programmens och användarens rättigheter som är en stor källa till dagens problem med virus. I säkra programspråk (t.ex java), Capability-system eller med kryptering hindrar man programmen att kringgå sina tilldelade rättigheter. Dessa metoder ger oss goda möjligheter att specificera och kontrollera vilka rättigheter ett program har. Vad de inte kan ge hjälp med är hur, när och varför programmet verkligen behöver resurserna och vilka konsekvenser detta kan få.

En annan ansats, att formellt bevisa att ett program uppfyller en specifikation, används för att garantera att delar av programmet är ofarliga och används t.ex i Javas säkerhetsmodell. Komplet analys av ett programs beteende är dock mycket resurskrävande [Cohen:88], särskilt eftersom vi behöver ta hänsyn till programmens interaktioner med varann. Dessutom är ett program endast skadligt om det verkligen utför de farliga delarna. Mycket analysarbete har annars gjorts i onödan.

Eftersom det inte går att helt avgöra ett programs farlighet utan att ta hänsyn till under vilka omständigheter det körs är det mest besläktade datasäkerhetsområdet 'Intrusion Detection' [Lunt 1993, Sundaram 1996]. Målet med Intrusion Detection är att upptäcka interna eller externa användare som missbrukar sina rättigheter eller otillåtet använder systemet, ofta genom att leta efter avvikelser från förväntat beteende. Traditionellt utgår man från att det är människor som missbrukar systemet, men många av metoderna kan även användas för att övervaka program.

Säkerhetsassistenten

Säkerhetsassistentens uppgift är att underlätta och automatisera tilldelandet av rättigheter baserat på kunskap om vad programmet förväntas göra för användaren. Assistenten behöver ha en modell över vad programmet förväntas göra samt vilka rättigheter som givits till andra program och kunna resonera om vilka konsekvenser utdelning av nya resurser kan få.

När ett nytt program startas kan användaren i dialog med säkerhetsassistenten ställa in övervakningen av potentiellt farliga beteenden. Genom att, baserat på programmets förväntade arbetsuppgift, begränsa vad det får göra minskas chansen att fördolda aktiviteter passerar oupptäckta. Övervakningen ger användaren mjuk säkerhet.

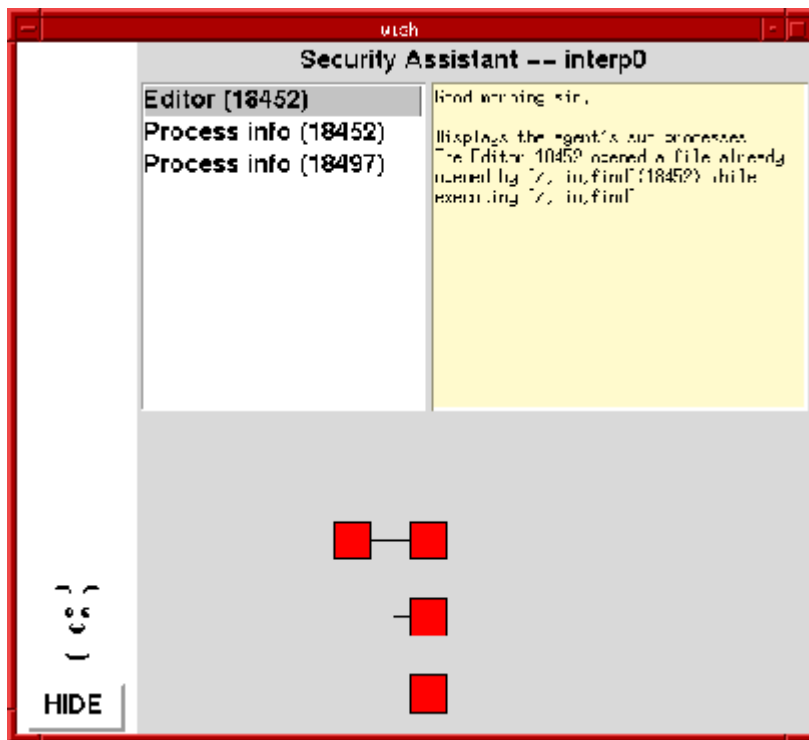
Ett viktigt scenario är hur assistenten övervakar tidigare okända program. Om användaren kan beskriva vilken *sorts* program detta program förväntas vara kan avvikelser från denna sorts beteende rapporteras. Vi tror att dessa sorter bör specificeras endast vagt för att inte begränsa möjligheterna för programmerare av vare sig applikationer eller sensorer.

Erfarenheter

Vi har implementerat prototyper för att undersöka olika sätt och möjligheter att samla in data från okända program samt för att undersöka vad sensorer kan förhandla för att kunna dela på arbete.

En prototyp övervakar Java-program och utnyttjar det ramverk som finns för att begränsa rättigheterna för små okända program, s.k applets. Framförallt saknas i Javas befintliga säkerhetsramverk möjligheter att övervaka vad appleten gör med resurser som det en gång fått tillgång till och möjligheter att styra hur mycket minne ett program får använda. Vi uppmärksammade också ett av de problem som kan uppstå om man endast tillhandahåller en detaljerad rättighetsarkitektur utan vidare stöd för hantering av rättighetstilldelning. Det fanns, när prototypen byggdes, inga tillgängliga Java-program som använde sig av systemresurser på något avancerat sätt. Det flesta program var endast enkla grafiska komponenter.

De övriga prototyperna utnyttjade i stället på de möjligheter som finns att studera programs beteende i den "vanliga" programkörningsomgivningen.



Figur 15. Gränssnitt till en personlig säkerhetsassistent

Figur 15 visar en assistent som klassificerar program enligt vilken *sort* det har, t.ex om programmet är ett editeringsprogram eller om det filtrerar data osv. Assistenten bedömer sedan programmen efter om deras beteende avviker från det förväntade för den sorten. Vilka programsorter som kommer finnas och hur de kommer övervakas kan ändras och är upp till behoven och fantasin hos framtida programmerare av applikationer och sensorer.

Det finns två huvudsakliga sätt för sensorer att undersöka programmens beteende. De kan antingen upptäcka avvikelser (anomalier) från vad som förväntats av programmen eller de kan försöka upptäcka mönster som svarar mot kända attacker eller risker. Båda ansatserna har sina fördelar och svagheter. Om en anomali upptäcks är det inte alltid lätt att ta reda på vad det övervakade programmet faktiskt försöker ta sig för. Sensorer som letar efter kända attacker riskerar å andra sidan att missa nya, tidigare okända attacker.

Sammanfattning

Utgående från scenariot ovan förutspår vi att öppna agentmarknader med fungerande mjuk säkerhet kommer att få följande karakteristiska drag:

- Systemets öppenhet gör det omöjligt att spåra varje deltagares identitet. Agenterna kommer istället vara anonyma eller själva bära med sig någon form av identitetsbevis.
- Det finns ingen central instans som bedömer eller står som garant för agenternas goda eller dåliga beteende. Det går inte att hindra illasinnade agenter från att finnas i systemet men varje individ kan låta bli att samverka med dem.
- Säkerhet är en vara som kan köpas och säljas på samma sätt som andra varor. Det är upp till varje individ att välja hur mycket som skall investeras i att avgöra hur mycket motparten kan lita på.

- Om en illasinnad agent upptäcks finns det effektiva sätt för denna information att spridas till andra agenter på marknaden.
- Om det kostar att befinna sig på eller synas på marknaden kommer de agenter som inte får några kunder försvinna från marknaden. Under rätt betingelser kommer systemet själorganisera sig till att bestå av flest effektiva eller godartade agenter.

Förteckning över bilagor inom området 'Personlig säkerhet'

- [1] Andreas Rasmusson. Interactive security assistance for end-user supervision of untrusted programs. Master's thesis, Royal Institute of Technology, Department of Computer and Systems Sciences, 1996.
- [2] Lars Rasmusson. Socially controlled global agent systems. Master's thesis, Royal Institute of Technology, Department of Computer and Systems Sciences, 1996.
- [3] Andreas Rasmusson and Sverker Janson. Personal security assistance for secure Internet commerce. In *New Security Paradigms '96*. ACM Press, 1996.
- [4] Lars Rasmusson and Sverker Janson. Simulated social control for secure Internet commerce. In *New Security Paradigms '96*. ACM Press, 1996.
- [5] Lars Rasmusson, Andreas Rasmusson, and Sverker Janson. Reactive security and social control. In *19th National Information Systems Security Conference*. NISSC, 1996.
- [6] Lars Rasmusson, Andreas Rasmusson, and Sverker Janson. Using agents to secure the Internet marketplace: Reactive security and social control. In *Proceedings of Practical Applications of Agents and Multi-Agent Systems 1997 (PAAM'97)*, London, UK, 1997.

Personlig kommunikation

Tomas Olsson, Andreas Rasmusson och Sverker Janson

Inom denna aktivitet försöker vi se användaren som en individ med en egen uppsättning intressen och behov som både kan vara privata och publika. All kommunikation med webbaserade tjänster och verktyg behöver därför vara personligt utformad. Utifrån detta perspektiv har vi designat en agentbaserad arkitektur som vi kallar ett "Virtual Community Library" [Rasmusson et al., 1998]. Den tror vi kommer att tillgodose den enskilde användarens behov på ett bättre sätt än dagens Internet. Dock är den inte konstruerad för att ersätta Internet, utan för att vara ett komplement. Se även det relaterade avsnittet om *Agentbaserade gränssnitt emot öppna tjänstarkitekturer*.

Med Internets tillväxt ökar antalet potentiella informationskällor explosionsartat. Det blir allt svårare för en användare att söka information och att hålla sig uppdaterad om nyttillkommen information. Sökmotorer, portaler och hubbar är alla försök att skapa mer eller mindre personligt anpassade tjänster för att underlätta för användarna att hitta och hålla sig uppdaterade. Dessa kan också ses som ytterligare potentiella informationskällor som användarna måste hitta och hålla reda på. Det måste därför finnas sätt för dessa teknikbaserade webbtjänsterna och webbplatser att tala om att de finns och vad de gör. Detta sker bland annat genom reklam på webbsidor, recensioner i tidningar men också via mun-till-mun-metoden, dvs. att användarna rekommenderar tjänster och webbplatser till varandra.

I vår arkitektur vill vi automatisera spridning av information och informationskällor genom att koppla ihop användarna och andra informationskällor i ett nätverk, automatisera mun-till-mun-metoden [Shardanand och Maes 1995] och göra kommunikationen mer personlig. Användarna kan också vara informationskällor i vår arkitektur.

Ett besläktat projekt är Yenta [Foner, 1997] som helt decentraliserat matchar användare med samma intressen. Ett annat närliggande projekt är ACORN [Marsh och Masrour 1997] som är ett agentbaserat system för att publicera och söka efter dokument. Detta sker med hjälp av smarta dokument som letar efter intresserade användare och smarta sökfrågor som kan leta efter dokument.

En vision om ett virtuellt bibliotek: Ett "Virtual Community Library"

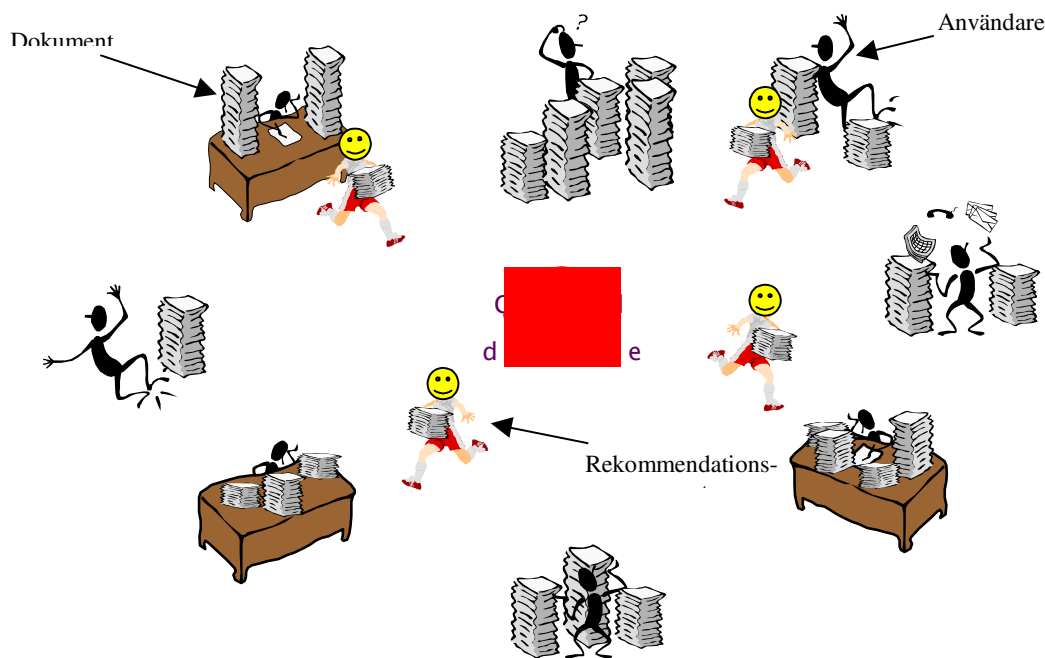


Figure 16: Visionen för ett "Virtual Community Library".

Vi försöker se världen som den är, utan en central databas. Människor arbetar, de läser, skriver och sorterar "dokument" i en miljö formad av dem själva, t.ex. deras hierarkiska filsystem eller e-posthanterare. Vi kallar en sådan miljö för ett personligt bibliotek. Även webbplatser kan ses som "personliga" bibliotek men med ett innehåll som speglar de intressen som dess ägare och användare har.

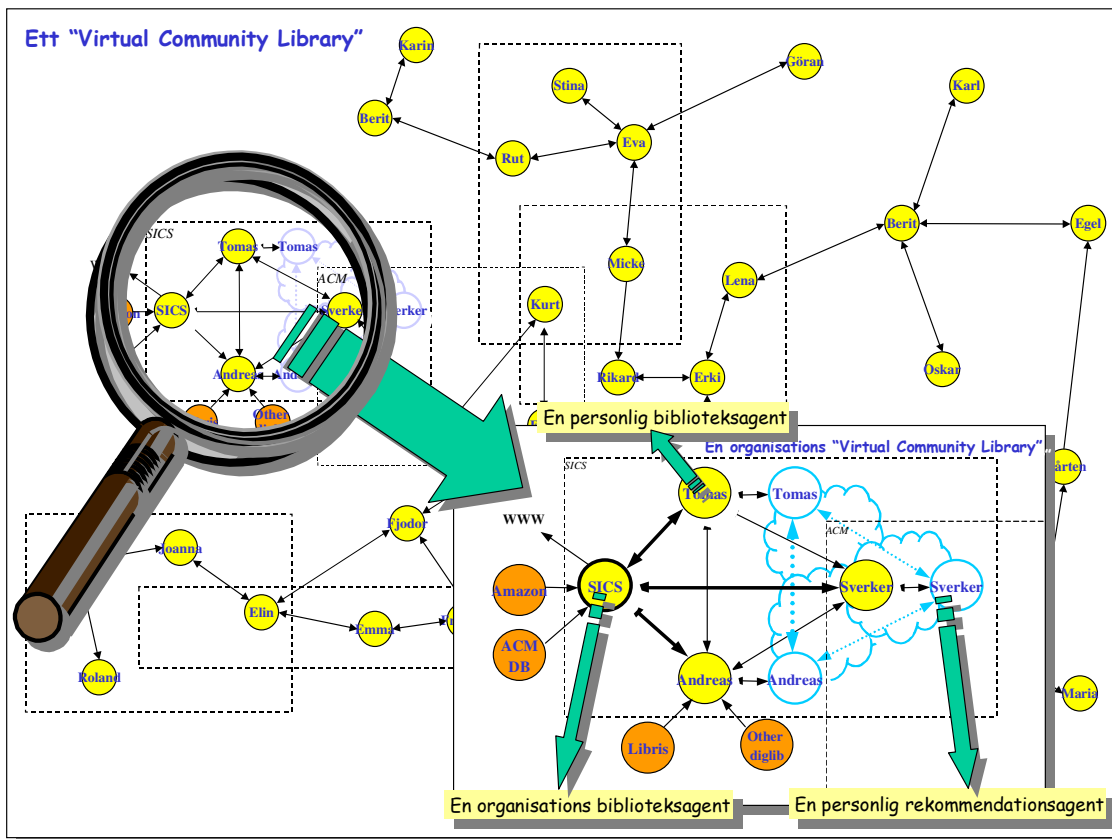
Vad vi vill göra är att koppla ihop alla dessa personliga bibliotek till ett stort virtuellt bibliotek som vi kallar ett "Virtual Community Library" (VCL). I ett VCL ska användarna kunna dra nytta av det arbete som de andra användarna lagt ned på sina personliga bibliotek.

Genom att inte se det virtuella biblioteket som en statisk enhet, utan som ett dynamiskt virtuellt samhälle ("community"), försöker vi få samma egenskaper som i ett riktigt samhälle. I ett riktigt samhälle sprids information genom att man tittar i någons bokhyllor, frågar om något eller berättar för andra om någonting eller söker i en databas. Kontrollen över informationsflödet tillhör de enskilda individerna (åtminstone i ett demokratiskt samhälle) och de kan skaffa information både aktivt (genom att fråga och leta) och passivt (genom att någon berättar).

Idén till VCL är inspirerat dels av Internet och dels av det klassiska bibliotekskonceptet. På Internet är det lätt att publicera dokument. Alla med egen uppkoppling kan starta en webbservare och publicera dokument. Det finns ingen central kontrollnod för informationsflödet och användare kan organisera sin information som de vill med hjälp av enkelriktade länkar.

Ett bibliotek har som mål att ha en komplett samling dokument inom sitt område och göra dokumentsamlingen tillgänglig för sina användare med hjälp av klassificeringssystem och sökverktyg.

Ett VCL försöker förena all dessa egenskaper. Användarna av ett VCL, det vill säga, de som har personliga bibliotek, kan publicera sina dokument i det egna personliga biblioteket och därmed göra det åtkomligt för alla andra användare. De åtkomliga dokumenten utgör det virtuella bibliotekets dokumentsamling.



Figur 17. Ett "Virtual Community Library" där ljusgrå noder är användarnas personliga biblioteksagenter eller organisationers biblioteksagenter (med tjock kant). Mörkgrå noder är webbplatser och webbtjänster medan de ofyllda cirklarna är personliga rekommendationsagenter. Pilarna visar i vilken riktning som information skickas mellan agenterna.

Det finns ingen central nod i nätet, utan alla personliga bibliotek i ett VCL är likvärdiga. Användarna kan organisera och klassificera sina personliga bibliotek på det sätt de själva önskar. Dessutom finns det möjlighet att göra distribuerade sökningar och att prenumerera på information om uppdateringar hos andra personliga bibliotek.

Genom att användarna organiserar sina personliga bibliotek själva så får vi databaser som beskriver användarna. En sådan databas kan sedan användas som användarprofil för att göra kommunikationen med andra tjänster såsom sökmotorer, andra personliga bibliotek, etc. mer personliga. Det skulle t.ex. gå att rangordna en användares sökresultatet utifrån nyckelorden i en viss dokumentkategori eller använda databasen för att rekommendera dokument via en rekommendationsagent.

Eftersom den information som läggs in i det personliga biblioteket är relevant för användaren så motverkar vi tendensen att användare sällan vill lägga in information i ett bibliotek som denne inte är intresserad av.

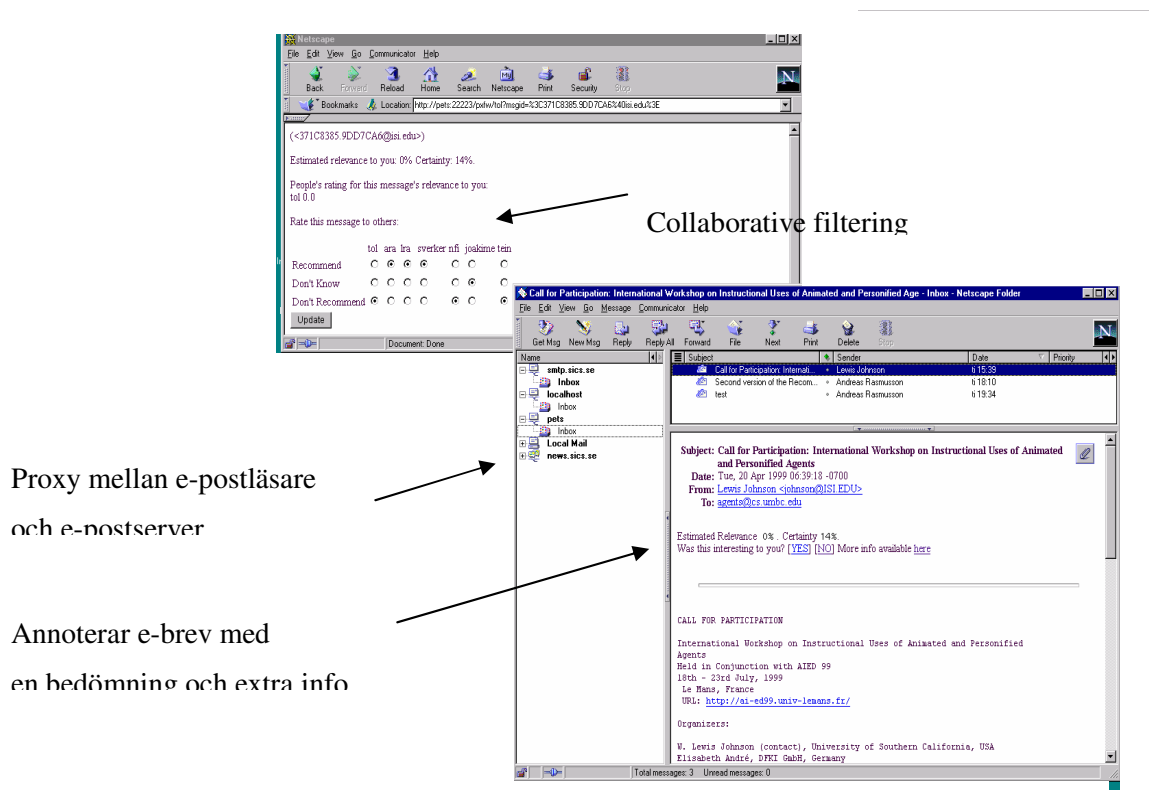
En arkitektur för ett "Virtual Community Library"

Basen för ett VCL är ett fleragentsystem som består av personliga biblioteksagenter och organisationers biblioteksagenter (se figur 17). En organisations biblioteksagent är ett specialfall av den personliga biblioteksagenten som representerar en organisation istället för en person, men under samma villkor. Dessutom kan varje biblioteksagent ha tillgång till en rekommendationsagent som har till uppgift att automatiskt hitta relevant information åt biblioteksagenten, t.ex. dokument till dess dokumentsamling.

I ett VCL representeras en användare av den personliga biblioteksagenten. Den personliga biblioteksagenten underlättar för användaren att skapa ett eget bibliotek utifrån användarens egna preferenser. Dessutom kan den ge personligt anpassad tillgång till olika webbtjänster, t.ex. sorterade resultat från sökmotorer. Användaren kan också prenumerera på nyinlagda eller förändrade dokument från andra personliga biblioteksagenter och låta agenten bevakna förändringar i webbplatser utanför VCL-et. Den ska också kunna utföra distribuerade sökningar i andra personliga biblioteksagenter.

En organisations biblioteksagent har hand om de externt synliga resurserna (via www i figur 17) för en organisation. Det kan t.ex. vara en sökbar dokumentsamling. Den är också ansvarig att informera agenter inom organisationen om delade resurser och användarrättigheter. Agenten är också en personlig biblioteksagent ur organisationens bibliotekaries synvinkel, då den prenumererar på information om dokument publicerade inom organisationen och innehåller därmed organisationens dokumentsamling.

I denna arkitektur så har inget bibliotek någon särstatus i fråga om att innehålla komplett eller korrekt information. Varje enskilt bibliotek är autonomt och fri att förhandla med andra bibliotek för att hitta ny information. Dock kommer stora dokumentsamlingar (institutioner och organisationer) fortfarande ”influera” vilka beskrivningar dokument får eftersom de oftare är anlitade än de vanliga personliga biblioteken.



Figur 18: Prototypen för en personlig biblioteksagent.

Varje biblioteksagent har möjlighet att ha en rekommendationsagent som har till uppgift att automatisera informationsspridningen mellan agenterna. En rekommendationsagent föreslår dokument eller informationskällor till användaren baserat på en implicit eller explicit användarmodell den byggt från användarens interaktion med sin biblioteksagent. För att hitta relevant information härmar den mun-till-mun-metoden och använder s.k. ”collaborative filtering”. Detta innebär att rekommendationsagenterna sänder rekommendationer till varandra och lär sig vilka andra de kan lita på [Olsson 1998]. Flödet av rekommendationer leder till att kluster av användare med liknande intressen uppstår och information kan spridas

snabbt mellan dem. Ett sådant nätverk med rekommendationsagenter kallar vi ett decentraliserat rekommendationssystem (se molnet i figur 17).

En prototyp av den personliga biblioteksagenten

Vi försöker bygga ett system med det personliga biblioteket som en integrerad del av en användares dagliga informationshantering och valde därför e-posthantering som försöksdomän. Inkommande och avsända meddelanden reflekterar en användares nuvarande aktiviteter och kommunikationsmönster. Dessa kan användas för att forma en implicit modell av användarens intressen och sociala nätverk. E-brev sorteras ofta ad-hoc snarare än enligt någon etablerad standard för katalognamn.

Prototypen för den personliga biblioteksagent som vi implementerat använder e-brev vilka är typiska dokument som användare vill mottaga, sända och utbyta. Dokumenten sorteras av användaren in i olika kataloger efter användarens egna preferenser. Det motsvarar något så när en klassificering av dokumenten enligt användarens intresseområden. Utifrån dessa kataloger kan vi lätt skapa tjänster baserade på aktuell arbetskontext, i detta fall den för närvarande valda katalogen. Det går också lätt att integrera externa tjänster genom att t.ex. addera kataloger till användarens kataloghierarki innehållande gränssnitt till de aktuella tjänsterna. I den befintliga prototypen har vi inte tittat på kommunikationen och förhandlingar mellan agenter.

Den personliga biblioteksagenten har vi implementerat som en proxy mellan en e-postläsare och en e-postserver (se figur 18). Genom att avlyssna kommunikationen på applikationsnivå är det möjligt att övervaka några aspekter av användarens beteende utan att ändra på vare sig e-postläsaren eller e-postservern.

Ett enkelt decentraliserat rekommendationssystem

För att testa den ovan beskrivna biblioteksagenten har vi implementera ett enkelt rekommendationssystem. Systemet använder s.k. "active collaborative filtering" [Maltz och Ehrlich 1995] för att filtrera e-postlistor (se figur 20).

Metaforen för detta enkla system är "vänner rekommenderar e-brev till varandra". Varje person känner till ett antal andra människor. Vi antar att varje person känner sig motiverad att spendera tid för att bedöma vad vännerna skulle tycka om ett meddelande. Detta är det väsentliga beteendet när e-brev är rekommenderade och "forwardade" till andra idag. Bedömningen sker i fönstret med "collaborative filtering" i figur 19.

Från dessa bedömningar kan systemet försöka förutsäga om en användare kommer att gilla ett mottaget meddelande eller inte. Systemet försöker dessutom förutsäga hur en användare kommer att bedöma ett meddelande åt en vän och fyller i bedömningen i förväg.

En poäng med denna enkla uppställning är faktiskt att den inte automatiserar rekommendationerna. Agenten fungerar bara som ett stöd i rekommendationsprocessen. Detta gör att det hela tiden är människor inblandade och det minskar risken för att icke-relevant e-post når en användare och ökar sannolikheten för att mycket viktig information verkligen når fram. En fråga är om vi kan lyckas automatisera denna process tillräckligt bra.

En simulator av ett decentraliserat rekommendationssystem

De grundläggande idéerna för vårt decentraliserade rekommendationssystem bestående av rekommendationsagenter är följande:

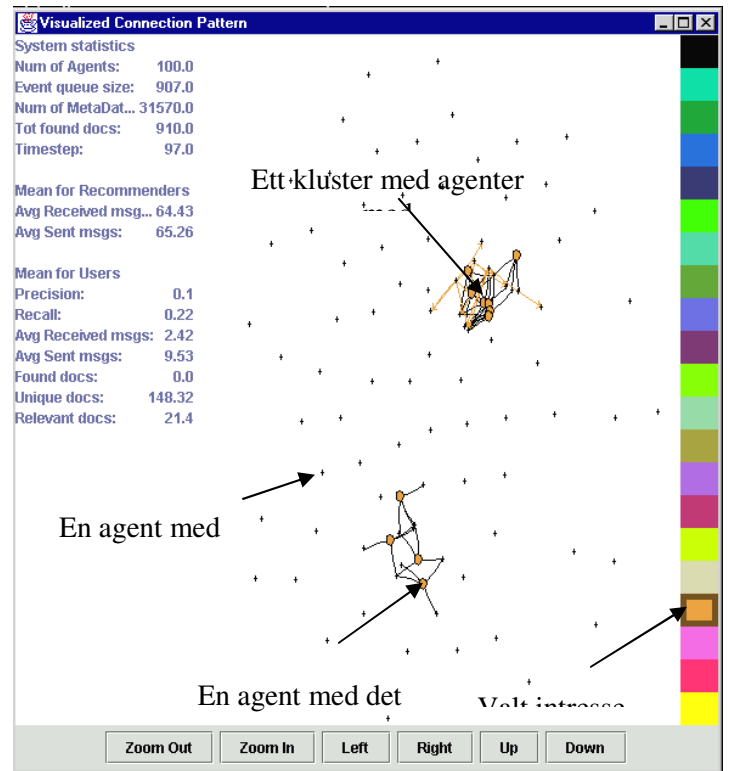
1. Låt varje användare ha en egen rekommendationsagent
2. Koppla ihop dessa agenter i ett nätverk
3. Låt varje agent fungera som en "broker" som bygger modeller av andra agenter (och sin användare)

4. Låt sedan agenterna routa information till de mest intresserade agenterna och användarna.

Vi antar att användarna när det fått en rekommendation kommer att ge någon slags respons. Om rekommendationen är ett dokument antar vi att användaren kommer att klassificera det i t.ex. en kategori. Vi kallar alla dokument som tillhör eller borde tillhöra en av användarens kategorier för användarens intresse.

Ett av målen med denna uppställning är att få rekommendationsagenterna att forma kluster med användare som har överlappande intressen. I klustren kan avståndet mellan agenter som representerar samma intresse vara kort och därmed kommer information att kunna spridas snabbare, nå rätt individer och ge upphov till mindre kommunikation.

Vi har implementerat en simulator av ett decentraliserat rekommendationssystem. Basen är ett fleragentsystem och i figur 20 visas en skärmdump av en sådan simulering.



Figur 20. En visualiserad simulering av ett decentraliserat rekommendationssystem.

Rekommendationsagenterna modellerar andra agenter baserat på de rekommendationer de får från dem. Sedan använder rekommendationsagenterna modellerna för att avgöra om de ska sända rekommendationer till de modellerade agenterna.

Simuleringen i figur 20 visar agenterna som små svarta eller stora grå noder. Stora grå noder är agenter som representerar användare med det intresse som för tillfället är valt att visas. Oriktade svarta kanter visar vilka andra agenter varje agent modellerar. De riktade kanterna (pilarna) visar att det mellan två agenter skickas en rekommendation av ett dokument som tillhör aktuellt intresse. Under simuleringens gång kan man se att agenterna bildar kluster bestående av agenter med samma intresse.

Forskningsfrågor

Vi vill studera hur virtuella samhällen som VCL fungerar. Vilka parametrar styr strukturen hos ett VCL och då i synnerlighet hos det decentraliserade rekommendationssystemet?

Några andra intressanta frågor är:

- Under vilka omständigheter kommer
 - undergrupper formas,
dvs. kommer användare finna varandra.
 - grupper vara stabila eller förändras
dvs. kommer förändrade intressen bryta upp tidigare formade grupper.
- Vilka incitament är det som får användarna att dela med sig av sina rekommendationer?

- Under vilka omständigheter kommer fördelarna för individerna vara så stora att de använder systemet?
- Kan systemet bli effektivare om vi sätter priser på alla rekommendationer och skapar en marknad?
- Hur påverkas systemet av agenter som skickar felaktig information?

Förteckning över bilagor inom området 'Personlig kommunikation'

- [1] Olsson T. 1998. Decentralized Social Filtering based on Trust. In: AAAI-98 Recommender Systems Workshop Papers. Technical Report WS-98-08. pp 84-88. AAAI Press.
- [2] Rasmusson A, Olsson T, and Hansen P. 1998. A Virtual Community Library: SICS Digital Library Infrastructure Project. Research and Advanced Technology for Digital Libraries. Second European Conference, ECDL'98, Heraklion, Crete, 19-23 September, Proceedings. Lecture Notes in Computer Science, Vol 1513. pp 677-678. Springer.

Den kommande perioden (1999-2002)

Inom IntAge-projektet har vi studerat många aspekter på agentteknologi: vilka möjligheter den ger och vilka nya problem som uppstår. Men många frågor är fortfarande outredda. Under de kommande tre åren vill vi fördjupa vår analys inom ett antal frågor som är grundläggande för vårt agent-perspektiv.

Från användarsynvinkel finns det två grundläggande frågeställningar vi vill fortsätta arbeta med.

- Hur kan vi bidra till en mer socialt baserad syn på informationsförädling? Vi vet att när människor söker information, gör de ofta det genom att vända sig till andra människor, snarare än via 'objektiv' information såsom uppslagsverk och kartor. Det här gäller både på det personliga och på det professionella planet. Om ditt barn är rödprickigt ringer du din mamma eller din BVC-sköterska, snarare än slår i hemmets läkarbok. Och när läkaren i sin tur söker efter information om lämplig medicinering, kan hon mycket väl förlita sig på information från en kollega mer än på informationen ur FASS. Även när vi inte direkt frågar efter information, så letar vi efter spår av andra människor. Vi går in på en restaurang där det redan finns mycket folk, till exempel. Så här information finns oftast inte med i vanliga gränssnitt, de är 'döda' omgivningar utan närvaro eller spår av andra människor.
- Vi vill fortsätta arbeta med frågan om hur vi kan använda agentmetaforen i användargränssnitt, s.k. antropomorfa gränssnittsagenter? Vad bidrar gränssnittsagenter med, och hur påverkar de oss?

Ur mjukvaruutvecklings-perspektivet vill vi studera öppna arkitekturer för samarbete och koordinering. I synnerhet vill vi arbeta med stöd för 'vanliga' användare i att utveckla samarbetet mellan tjänster över tiden. I KIMSAC-projektet visade sig detta vara en nyckelfunktion som det inte fanns utrymme att utveckla inom projektet. Många frågor är obesvarade, framför allt när det gäller hur man kan upptäcka att två tjänster borde kunna samarbeta när de inte kan det i praktiken, och hur en vanlig användare ska kunna påverka tjänster så de lär sig samarbeta.

Vi vill också arbeta vidare med sant distribuerade modeller för marknadsplatser och rekommendationssystem. Här finns många obesvarade frågeställningar, framför allt vad gäller studier av mer komplexa modeller av agentsamverkan och kontraktsförhandling.

IntAge är ett projekt som blandar teknikutveckling med användarorienterade, empiriska studier av olika gränssnitts- och systemlösningar. På detta sätt kan vi inte bara bidra till den tekniska utvecklingen, eller förstå dess effekter på den mänskliga användaren. Vi kan påverka utvecklingen att gå i en riktning som är gynnsam för den enskilda människan och för samhället som helhet. Visionen om ett fritt och demokratiskt IT-samhälle med skydd för individen är grunden för vårt arbete. För oss räcker det inte att analysera konsekvenserna av IT-utvecklingen - vi anser att vi måste påverka den.

Referenser

- [1] Brennan, Susan E. and Ohaeri, Justina O. (1994) Effects of Message Style on Users' Attributions toward Agents, Conference Companion, CHI'94, Boston.
- [2] Bretan, I. (1995) Natural Language in Model World Interfaces. Licentiate thesis, Stockholm University, Report no 95-017.
- [3] Bylund, M. (1999). Coordinating Adaptations in Open Service Architectures. M.Sc. thesis, Uppsala University, Uppsala.
- [4] Markus Bylund and Annika Waern. Service Contracts: Coordination of User-Adaptation in Open Service Architectures. Journal of Personal Technologies, Volume 2, Issue 4, 1998.
- [5] P. Charlton, Y. Chen, E. Mamdani, J. Pitt, F. Espinoza, O. Olsson, A. Waern, and F. Somers. 1997. Open Agent Architecture Supporting Multimedia Services on Public Information Kiosks. In Proceedings of the 3rd International Conference on the Practical Application of Intelligent Agents and Multi-Agent Methods, pp. 445-465, London, England, April.
- [6] Clanton, Chuck (1998) An Interpreted Demonstration of Computer Game Design, HCI Lessons from Games, Demonstation at Computer-Human Interaction, CHI'98, Los Angeles.
- [7] Economic Times. "The Economics of Crime", Vol. 4 No 1, 1995.
- [8] Joakim Eriksson, Niclas Finne, and Sverker Janson. Information and interaction in MarketSpace - towards an open agent-based market infrastructure. In Second USENIX Workshop on Electronic Commerce, USENIX, Nov 1996.
- [9] Joakim Eriksson, Niclas Finne, and Sverker Janson. Information and interaction in MarketSpace and their implementation in Prolog. Paper presented at LPNET'96 (1st Workshop on Logic Programming Tools for Internet Applications), ICLP'96, Sept 1996.
- [10] Joakim Eriksson, Niclas Finne, and Sverker Janson. Surfing the market and making sense of the web. Paper presented at Programming the Web -- In Search of APIs, WWW5, Paris, May 1996.
- [11] Joakim Eriksson and Niclas Finne. MarketSpace: an open agent-based market infrastructure. Master's Thesis. Computing Science Department, Uppsala University and Swedish Institute of Computer Science, 1997.
- [12] Joakim Eriksson, Fredrik Espinoza, Niclas Finne, Fredrik Holmgren, Sverker Janson, Niklas Kaltea, and Olle Olsson. An Internet software platform based on SICStus Prolog. Paper presented at Logic Programming and the Web, WWW5, San Jose, 1997.
- [13] Joakim Eriksson, Niclas Finne, and Sverker Janson. To each and everyone an agent: augmenting web-based commerce with agents. In Proceedings of the International Workshop on Intelligent Agents on the Internet and Web, World Congress on Expert Systems, Mexico City, March 1998.
- [14] Joakim Eriksson, Niclas Finne, and Sverker Janson. SICS MarketSpace: an agent-based market infrastructure. In First International Workshop on Agent-Mediated Electronic Trading, AMET-98, Selected Papers, Lecture Notes in Computer Science 1571, Springer-Verlag, 1999.
- [15] Espinoza, Fredrik (1998) sicsDAIS: Managing User Interaction with Multiple Agents, Licentiate thesis at the department of Computer and Systems Sciences, October, 1998.

- [16] Espinoza, F. (to appear). sicsDAIS: A Multi-Agent Interaction System for the Internet. Paper presented at the WebNet 99-World Conference on the WWW and Internet, Hawaii.
- [17] Foner L N. 1997. In: Proceedings of The First International Conference on Autonomous Agents (Agents '97). pp 301-307. ACM Press.
- [18] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. (1998) The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI. Morgan Kaufmann: San Francisco, pp. 256-265.
- [19] Anette Hulth and Anna Jonsson. (1999). An Experimental Digital Library Platform - A Demonstrator Prototype for the DigLib Project at SICS. Department of Computer and Systems Sciences. Stockholm. (Master thesis)
- [20] Kristina Höök, Designing and Evaluating Intelligent User Interfaces, in proceedings of IUI'98, IUI'99 and British HCI'98, 1998 and 1999.
- [21] Höök, K. (1998) Evaluating the Utility and Usability of an Adaptive Hypermedia System, in Journal of Knowledge-Based Systems, Volume 10, issue 5, 1998.
- [22] Kristina Höök. Steps to Take Before IUI Becomes Real. in Interacting with Computers, forthcoming.
- [23] Höök, Kristina, Åsa Rudström, Annika Waern. Edited Adaptive Hypermedia: Combining Human and Machine Intelligence to Achieve Filtered Information. Presented at the workshop "Flexible Hypertext" in Southampton during the Hypertext conference in April 6 - 11th, 1997.
- [24] Höök, Kristina, Sjölander, Marie, Ereback, Anna-Lena, and Persson, Per (1999) Dealing with the Lurking Lutheran view on Interfaces: Evaluation of the Agneta and Frida system, Workshop on Behavior Planning for Life-Like Characters and Avatars, Elisabeth Andre, Angelica de Antonio Jiménez, and Kristina Höök (eds.) In Conjunction with the i3 Spring Days '99, March 9th -10th 1999, Sitges, Spain.
- [25] King, W.J., and Ohya, J. (1995) The representation of agents: a study of phenomena in virtual environments, Proc. of the 4th IEEE International Workshop on Robot and Human Communication, RO-MAN'95, Tokyo, Japan.
- [26] Koda, T. (1997) Agents with faces: A study on the effects of personification of software agents. Master's thesis, MIT.
- [27] Koda, T. and Maes, P. (1996) Agents with Faces: The Effects of Personification of Agents, Proceedings of HCI'96, London, U.K.
- [28] Yannis Labrou, and Tim Finin. A Semantics Approach for KQML - A generic purpose communication language for software agents. in proc. of the third international conference on information and knowledge management, eds N.R. Adam, B. K. Bhargava and Y. Yesha, pages 447-455. New York, Association of Computing Machinery (1994).
- [29] Teresa F Lunt, A Survey of Intrusion Detection Techniques. Computers & Security, Vol. 12 No 4, June 1993, pages 405 – 418.
- [30] Maltz D and Ehrlich K. Pointing the way: active collaborative filtering. (1995) CHI'95 proceedings.
- [31] Marsh S and Masrouy Y. 1997. Agent Augmented Community Information — The ACORN Architecture. In: Proceedings CASCON'97, Meeting of Minds.

- [32] Tomas Olsson. Decentralized Social Filtering based on Trust. In AAAI-98 Recommender Systems Workshop Papers. Technical Report WS-98-08. pp 84-88. AAAI Press, 1998.
- [33] Persson, Per (1999) AGNETA & FRIDA: Browsing, Narrative and Emotion, Workshop on Behavior Planning for Life-Like Characters and Avatars, Elisabeth Andre, Angelica de Antonio Jiménez, and Kristina Höök (eds.) In Conjunction with the i3 Spring Days '99, March 9th -10th 1999, Sitges, Spain.
- [34] Eric Rasmusen. *Games and Information*. Blackwell, Cambridge, MA, 1989.
- [35] Lars Rasmusson and Sverker Janson. Simulated social control for secure Internet commerce. In New Security Paradigms '96, ACM Press, 1996.
- [36] Andreas Rasmusson and Sverker Janson. Personal security assistance for secure Internet commerce. In New Security Paradigms '96, ACM Press, 1996.
- [37] Lars Rasmusson, Andreas Rasmusson and Sverker Janson. Reactive security and social control. In 19th National Systems Security Conference, Baltimore, 1996.
- [38] Lars Rasmusson, Andreas Rasmusson and Sverker Janson. Using Agents to Secure the Internet Marketplace: Reactive Security and Social Control. In Proceedings of the Second Annual Conference on Practical Applications of Intelligent Agents and Multi-Agent Systems, 1997.
- [39] Rasmusson A, Olsson T, and Hansen P. 1998. A Virtual Community Library: SICS Digital Library Infrastructure Project. Research and Advanced Technology for Digital Libraries. Second European Conference, ECDL'98, Heraklion, Crete, 19-23 September, Proceedings. Lecture Notes in Computer Science, Vol 1513. pp 677-678. Springer.
- [40] Lars Rasmusson and Sverker Janson. Agents, self-interest and electronic markets. Knowledge Engineering Review, Vol. 13:2, 1999, 1-8.
- [41] Reeves, Byron and Nass, Clifford (1996) The Media Equation: How People Treat Computers, Television, and New media Like Real People and Places, Cambridge University Press.
- [42] Rich, Charles, and Sidner, Candace L. (1997) Segmented Interaction History in a Collaborative Interface Agent, Proceedings of IUI'97, Orlando, Florida.
- [43] Rudström, Åsa, Waern, Annika, and Höök, Kristina, (1997) Interactive Adaptation of Intranet Newsletters, presented at the workshop "Adaptive Systems and User Modeling on the World Wide Web" held in conjunction with UM97: The Sixth International Conference on User Modeling, Sardinia, 1997.
- [44] Kent Saxin Hammarström, Höök, Kristina and Ereback, Anna-Lena, Convene - MUD interfaces for disabled users, in the ERCIM workshop "User Interfaces for All", edited by Stephanidis, C., D. Benyon, N. Carbonell, P.L. Emiliani, G. Faconti, A. Kobsa, A. Waern, M. Wilson (1997) 3rd ERCIM Workshop on User Interfaces for All. Obernai, November 2-4 1997. INRIA/ERCIM.
- [45] Kent Saxin Hammarström, Anna-Lena Ereback, Kristina Höök, and Marie Sjölander. Convene - MUD Interfaces for Disabled Users, in ed. Britt Östlund "Users In Action!" (1999), KFB-Rapport 1999:8.
- [46] Shardanand U. and Maes P. (1995). Social Information Filtering: Algorithms for Automating "Word of Mouth". Proceedings of the ACM Conf. Human Factors in Computing Systems CHI'95, 210-217, New York: ACM.
- [47] Shneiderman, Ben (1997) Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces, Proceedings of 1997 International Conference on

Intelligent User Interfaces, (eds.) Johanna Moore, Ernest Edmonds, and Angel Puerta, ACM, Orlando, Florida.

- [48] Sproull, Lee, Subramani, M., Kiesler, S., Walker, J. H., and Waters, K. (1996) When the Interface is a Face, *Human-Computer Interaction*, 1996, Vol. 11, pp. 97 - 124.
- [49] Suchman, L. A. (1987) *Plans and Situated Actions: The problem of human-machine communication*, Cambridge University Press.
- [50] Suchman, L. A. (1997) *From Interactions to Integrations*, Proc. of Human-Computer Interaction INTERACT'97, S. Howard, J. Hammond, and G. Lindegaard (eds.), Chapman & Hall.
- [51] Aurobindo Sundaram. *An Introduction to Intrusion Detection*. Crossroads, the ACM Student Magazine, Vol 2 No 4, April 1996.
- [52] Annika Waern. "DatorVarelser". Populärvetenskaplig artikel i *Framtider* nr 2, 1998, institutet för framtidsstudier. Även i boken "The Global Tendency Machine - trycket från framtiden".
- [53] Annika Wærn. *Service Contract Negotiation - Agent-Based Support for Open Service Environments*. In Proc. 1998 workshop on Distributed Artificial Intelligence, at the 4th Australian conference on Artificial Intelligence (AI'98), Brisbane, Australia, July 1998. Revised version in 'Multi-Agent Systems: Theories, Languages and Applications'. ed. C. Zhang and D. Lukose, Lecture notes in Artificial Intelligence 1544, 1998.
- [54] Annika Waern, Mark Tierney, Åsa Rudström, Jarmo Laaksolahti and Torben Mård. "Digital Libraries: Information Broker Roles in Collaborative Filtering". In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, Crete, September 1998, LNCS Vol. 1513, 1998.
- [55] Annika Waern, Mark Tierney, Åsa Rudström and Jarmo Laaksolahti. *ConCall: An information service for researchers based on EdInfo*. SICS Internal Report T98:04, 1998.
- [56] Annika Waern, Mark Tierney, Åsa Rudström, Jarmo Laaksolahti and Torben Mård. *ConCall: Edited and Adaptive Information Filterin*. in proceedings of the 1999 Intelligent User Interfaces Conference.
- [57] Annika Waern and Kristina Höök. *Interface Agents: A new metaphor for human-computer interaction and its application to Universal Access*. in ed. C. Stephanidis, 'User Interfaces for All', Lawrence Erlbaum, forthcoming.
- [58] Annika Waern, Charlotte Averman, Mark Tierney and Åsa Rudström. *Information Services Based on User Profile Communication*. in *Proceedings of the seventh International Conference on User Modelling*, Banff, Canada, forthcoming June 1999. (Produced within the EdInfo project)

Links

- Bluetooth Special Interest Group (SIG) - <http://www.bluetooth.com>
- EdInfo - <http://www.sics.se/humle/projects/edinfo/>
- Expressen - <http://www.expressen.se/>
- Grasshopper - <http://www.ikv.de/>
- Java - <http://java.sun.com/>
- Jini - <http://www.sun.com/jini/>
- Netscape Netcenter's Calendar - <http://home.netscape.com/calendar/>

- Nokia 9110 Communicator - <http://www.nokia.com/phones/9110/>
- PalmPilot - <http://www.palm.com/>
- Symbian - <http://www.symbian.com/>
- The Object Management Group (OMG) - <http://www.omg.org/>
- WAP Forum - <http://www.wapforum.org/>
- Windows CE - <http://www.microsoft.com/windowsce/>
- Foundations of Intelligent Physical Agents - <http://www.fipa.org/>
- Jennifer James - <http://www.extempo.com/>
- Petz - <http://www.petz.com/>
- PPP persona - <http://persona.dfki.de:8080/>
- You dont know Jack - <http://www.berksys.com/products/jack/jack.html>

Förteckning över bilagor

- [1] Bylund, M. (1999). Coordinating Adaptations in Open Service Architectures. M.Sc. thesis, Uppsala University, Uppsala.
- [2] Markus Bylund and Annika Waern. Service Contracts: Coordination of User-Adaptation in Open Service Architectures. *Journal of Personal Technologies*, Volume 2, Issue 4, 1998.
- [3] P. Charlton, Y. Chen, E. Mamdani, J. Pitt, F. Espinoza, O. Olsson, A. Waern, and F. Somers. 1997. Open Agent Architecture Supporting Multimedia Services on Public Information Kiosks. In *Proceedings of the 3rd International Conference on the Practical Application of Intelligent Agents and Multi-Agent Methods*, pp. 445-465, London, England, April.
- [4] Joakim Eriksson, Niclas Finne, and Sverker Janson. Information and interaction in MarketSpace - towards an open agent-based market infrastructure. In *Second USENIX Workshop on Electronic Commerce, USENIX*, Nov 1996.
- [5] Joakim Eriksson, Niclas Finne, and Sverker Janson. Information and interaction in MarketSpace and their implementation in Prolog. Paper presented at *LPNET'96 (1st Workshop on Logic Programming Tools for Internet Applications), ICLP'96*, Sept 1996.
- [6] Joakim Eriksson, Niclas Finne, and Sverker Janson. Surfing the market and making sense of the web. Paper presented at *Programming the Web -- In Search of APIs, WWW5*, Paris, May 1996.
- [7] Joakim Eriksson and Niclas Finne. MarketSpace: an open agent-based market infrastructure. Master's Thesis. Computing Science Department, Uppsala University and Swedish Institute of Computer Science, 1997.
- [8] Joakim Eriksson, Fredrik Espinoza, Niclas Finne, Fredrik Holmgren, Sverker Janson, Niklas Kaltea, and Olle Olsson. An Internet software platform based on SICStus Prolog. Paper presented at *Logic Programming and the Web, WWW5*, San Jose, 1997.
- [9] Joakim Eriksson, Niclas Finne, and Sverker Janson. To each and everyone an agent: augmenting web-based commerce with agents. In *Proceedings of the International Workshop on Intelligent Agents on the Internet and Web, World Congress on Expert Systems*, Mexico City, March 1998.
- [10] Joakim Eriksson, Niclas Finne, and Sverker Janson. SICS MarketSpace: an agent-based market infrastructure. In *First International Workshop on Agent-Mediated Electronic Trading, AMET-98, Selected Papers, Lecture Notes in Computer Science 1571*, Springer-Verlag, 1999.
- [11] Espinoza, Fredrik (1998) sicsDAIS: Managing User Interaction with Multiple Agents, Licentiate thesis at the department of Computer and Systems Sciences, October, 1998.
- [12] Espinoza, F. (to appear). sicsDAIS: A Multi-Agent Interaction System for the Internet. Paper presented at the *WebNet 99-World Conference on the WWW and Internet*, Hawaii.
- [13] Kristina Höök, Designing and Evaluating Intelligent User Interfaces, in proceedings of *IUI'98, IUI'99 and British HCI'98*, 1998 and 1999.
- [14] Höök, K. (1998) Evaluating the Utility and Usability of an Adaptive Hypermedia System, in *Journal of Knowledge-Based Systems*, Volume 10, issue 5, 1998.

- [15] Kristina Höök. Steps to Take Before IUI Becomes Real. in *Interacting with Computers*, forthcoming.
- [16] Höök, Kristina, Sjölander, Marie, Ereback, Anna-Lena, and Persson, Per (1999) Dealing with the Lurking Lutheran view on Interfaces: Evaluation of the Agneta and Frida system, Workshop on Behavior Planning for Life-Like Characters and Avatars, Elisabeth Andre, Angelica de Antonio Jiménez, and Kristina Höök (eds.) In *Conjunction with the i3 Spring Days '99, March 9th -10th 1999, Sitges, Spain.*
- [17] Olsson T. 1998. Decentralized Social Filtering based on Trust. In: *AAAI-98 Recommender Systems Workshop Papers*. Technical Report WS-98-08. pp 84-88. AAAI Press.
- [18] Lars Rasmusson, Andreas Rasmusson, and Sverker Janson. Using agents to secure the Internet marketplace: Reactive security and social control. In *Proceedings of Practical Applications of Agents and Multi-Agent Systems 1997 (PAAM'97)*, London, UK, 1997.
- [19] Andreas Rasmusson. Interactive security assistance for end-user supervision of untrusted programs. Master's thesis, Royal Institute of Technology, Department of Computer and Systems Sciences, 1996.
- [20] Lars Rasmusson. Socially controlled global agent systems. Master's thesis, Royal Institute of Technology, Department of Computer and Systems Sciences, 1996.
- [21] Andreas Rasmusson and Sverker Janson. Personal security assistance for secure Internet commerce. In *New Security Paradigms '96*. ACM Press, 1996.
- [22] Lars Rasmusson and Sverker Janson. Simulated social control for secure Internet commerce. In *New Security Paradigms '96*. ACM Press, 1996.
- [23] Lars Rasmusson, Andreas Rasmusson, and Sverker Janson. Reactive security and social control. In *19th National Information Systems Security Conference. NISSC*, 1996.
- [24] Lars Rasmusson and Sverker Janson. Agents, self-interest and electronic markets. *Knowledge Engineering Review*, Vol. 13:2, 1999, 1-8.
- [25] Rasmusson A, Olsson T, and Hansen P. 1998. A Virtual Community Library: SICS Digital Library Infrastructure Project. Research and Advanced Technology for Digital Libraries. Second European Conference, ECDL'98, Heraklion, Crete, 19-23 September, Proceedings. *Lecture Notes in Computer Science*, Vol 1513. pp 677-678. Springer.
- [26] Kent Saxin Hammarström, Höök, Kristina and Ereback, Anna-Lena, Convene - MUD interfaces for disabled users, in the ERCIM workshop "User Interfaces for All", edited by Stephanidis, C., D. Benyon, N. Carbonell, P.L. Emiliani, G. Faconti, A. Kobsa, A. Waern, M. Wilson (1997) 3rd ERCIM Workshop on User Interfaces for All. Obernai, November 2-4 1997. INRIA/ERCIM.
- [27] Kent Saxin Hammarström, Anna-Lena Ereback, Kristina Höök, and Marie Sjölander. Convene - MUD Interfaces for Disabled Users, in ed. Britt Östlund "Users In Action!" (1999), KFB-Rapport 1999:8.
- [28] Annika Waern. "DatorVarelser". Populärvetenskaplig artikel i *Framtider* nr 2, 1998, institutet för framtidsstudier. Även i boken "The Global Tendency Machine - trycket från framtiden".
- [29] Annika Wærn. Service Contract Negotiation - Agent-Based Support for Open Service Environments. In *Proc. 1998 workshop on Distributed Artificial Intelligence*, at the 4th Australian conference on Artificial Intelligence (AI'98), Brisbane, Australia, July 1998.

Revised version in 'Multi-Agent Systems: Theories, Languages and Applications'. ed. C. Zhang and D. Lukose, Lecture notes in Artificial Intelligence 1544, 1998.

- [30] Annika Waern, Mark Tierney, Åsa Rudström and Jarmo Laaksohetti. ConCall: An information service for researchers based on EdInfo. SICS Internal Report T98:04, 1998.
- [31] Annika Waern and Kristina Höök. Interface Agents: A new metaphor for human-computer interaction and its application to Universal Access. in ed. C. Stephanidis, 'User Interfaces for All', Lawrence Erlbaum, forthcoming.