

SISU

rapport 1:87

Ett förslag till

referensmodell för Människa/Dator -interaktion

SISU - Svenska Institutet för Systemutveckling

Box 1250
163 13 SPÅNGA

Kistagången 26
KISTA

08-750 75 00

DIALOGMODELLERING

Ett Förslag till REFERENSMODELL för Människa/Dator-Interaktion

Slutrapport från projektet dialogmodellering
Ett ramprojekt inom SISU

Stefan BRITTS, SISU

Marie-Louise WARNSTRÖM, statskontoret

Karl-Olof WIGANDER, Programator (projektledare)

Gunilla ÅKERBLOM, Kommundata

För SISU, juni 1987

INNEHÅLLSFÖRTECKNING:

Bakgrund och syfte	1
Behov / projektbeskrivning	1
Sammanfattning	2
Rapportstruktur	2
Generellt om referensmodeller	3
Modellpresentation	5
Modellbild	6
Modellbeskrivning	7
Sidoordnade delar	10
Exempel	12
Avslutning	20
Referenser	21

BAKGRUND OCH SYFTE

Inom SISU (Svenska Institutet för Systemutveckling) har ett projekt som syftat till att utveckla en referensmodell för människa / dator-interaktion just avslutats. Projektet har drivits som ett samarbetsprojekt inom ramprogrammet. Förutom SISU har tre intressenter - Programator, Kommundata och statskontoret deltagit i arbetet. Föreliggande dokument utgör projektets slutrapport. Projektet har slutförts inom planerad tid och inom ramen för tilldelade resurser.

Den föreslagna referensmodellen är avsedd tjäna som ett ramverk inom vilket olika arbeten kring människa / dator-dialoger kan utföras, t ex studium av befintliga dialoger / programprodukter eller utformning av nya sådana. Den centrala uppgiften för projektet har varit att konstruera och beskriva modellens olika komponenter / skikt.

BEHOV / PROJEKTBESKRIVNING

Mycket arbete som går ut på att förbättra människa / dator-dialoger läggs ned i olika organisationer. Dessa arbeten är såväl av teknisk art som av typen "human factors". Exempel på det förra är effektivisering av dialogprogrammeringen, utformning av bättre (i någon mening) dialoger och standardiseringsansträngningar. Exempel på det senare är studium av olika dialogstilar och interaktionstekniker samt användaranpassning av dialoger. Samtidigt försöker varje organisation följa utvecklingen av området människa / dator-interaktion i syfte att ta till sig nya kunskaper som kan vara av direkt nytta.

Arbetet försvåras avsevärt av den brist på "struktur" som kännetecknar området. Ojämförbara begrepp ställs mot varandra, det saknas en referensram inom vilket arbetet kan ske. Detta försvårar och försenar en utveckling - som eftersträvas i många organisationer - av det viktiga området människa / dator-interaktion. Denna utveckling väntas leda till ökad insikt om möjligheter och svårigheter samt bättre grundade val av utrustning och dialogformer, för att nämna ett par exempel.

Detta projekt har utformat ett förslag till en referensmodell vilken:

- o underlättar strukturering av koncept, tekniker, problemställningar som figurerar inom området,
- o förbättrar kommunikationen om dialogfrågor,
- o möjliggör bättre bedömning av produkters funktionalitet,
- o bidrar till att skapa insikt om vilken betydelse människa / dator-dialogen spelar för den totala systemutformningen och användarnas uppfattning av det färdiga systemet.

Rätt använd bör referensmodellen kunna bidra till att skapa en bas för bättre absorbering av den metodik och teknik för människa / dator-interaktion som strömmar fram i en allt snabbare takt och alltså tillfredsställa de ovan nämnda behoven.

SAMMANFATTNING

Arbetet med att ta fram en referensmodell för gränssnittet människa-dator har resulterat i sex skikt och två ändpunkter. Ändpunkterna avgränsar referensmodellen uppåt och nedåt, men ingår inte i dess centrala del. *Uppgift*, dvs arbetsuppgiften som skall utföras, utgör modellens gräns uppåt och *fysik*, dvs de enheter som används vid interaktionen, utgör dess gräns nedåt. De sex skikten är *verktyg*, *dialogstruktur*, *dialogstil*, *uppgiftshantering*, *formatering* och *interaktionsform*.

Skikten är hierarkiskt presenterade i denna rapport. I verkligheten är dock sambanden ofta mer komplicerade än så. Vissa skikt är mer intimt bundna till varandra än andra. I ett tidigt skede försökte arbetsgruppen grafiskt beskriva skiktens inbördes relationer. Det visade sig emellertid svårt att ge uttryck för detta rittekniskt. Den hierarkiska indelningen är dock fullt tillräcklig för att åskådliggöra alla väsentliga aspekter på denna övergripande nivå.

Ett flertal funktioner skär genom modellens samtliga skikt. Dessa benämns sidoordnade aspekter. Sådana aspekter finns på båda sidor om modellen. På ena sidan har vi lokaliserat *beslutsfattande*, *problemlösande*, *memorerande*, *igenkännande* och *uppiäckande*. På andra finns *sårbarhet*, *säkerhet*, *integritet*, *behörighet* och *kvalitetssäkring*. Den senare är mer datororienterad än den förra.

Samtliga skikt i modellen, utom skiktet uppgiftshantering, skulle kunna realiseras i ett skelett (*) för generellt bruk vilket kan ses som hjälpfunktioner för att uppgiftshanteringen, som är helt applikationsorienterad, ska kunna utföras eller nås så enkelt som möjligt.

RAPPORTSTRUKTUR

Rapporten inleds med ett generellt avsnitt om referensmodeller. Därefter presenteras översiktligt arbetsgruppens förslag till referensmodell för människa / dator-interaktion varpå följer en mer detaljerad beskrivning av dess centrala del. Sidoordnade delar behandlas i ett separat avsnitt. Före avslutningen följer ett längre kapitel med exempel på tillämpningar av referensmodellen.

(*) Med "skelett" avses en standardiserad programvara vars struktur ansluter till skikten i referensmodellen. Skelettet är applikationsoberoende och återfinns i alla applikationers användargränssnitt.

GENERELLT OM REFERENSMODELLER

Två vanligt förekommande problem vid utveckling av datasystem är dels att få komponenter från olika leverantörer att samverka dels att åstadkomma en gemensam bild av vad de olika komponenterna skall åstadkomma oavsett om de ska samverka eller ej.

Det förra problemet aktualiseras när utrustning och / eller programvara upphandlas från olika leverantörer eller när existerande komponenter skall kopplas samman, t ex vid datakommunikation. Det senare problemet aktualiseras när man vill föreskriva en lämplig ansats att organisera ett system, t ex ett databashanteringssystem.

Referensmodeller har visat sig fylla en viktig funktion vid hanteringen av båda dessa typer av problem.

Det mest kända exemplet på en referensmodell är OSI-modellen. Denna är avsedd att hantera kommunikation i öppna system (kommunicerande datorer). OSI-modellen strukturerar på ett föredömligt sätt olika aspekter på kommunikationsprocessen i sju nivåer / skikt, från fysiska signaler till applikationsprogram. Utgående från OSI-modellen har olika standarder utvecklats täckande olika nivåer i modellen. Exempel på sådana standarder är X.400, X.25 och X.21.

OSI-modellen har haft mycket stor genomslagskraft och har därigenom hjälpt till att bringa reda inom det tidigare mycket ostrukturerade datakommunikationsområdet. Referensmodeller har med varierande framgång också utvecklats inom andra områden såsom databashantering (ANSI / SPARC-modellen) och grafisk databehandling.

Även referensmodeller för människa / dator-interaktion är under utveckling. Tanken är att dessa, på motsvarande sätt som OSI-modellen, ska bringa ordning inom ett område där antalet verktyg, tekniker och metoder ständigt växer i antal. Vad som saknas idag är en gemensam bild av vad människa / dator-interaktionen innehåller och hur denna bör organiseras med hänsyn taget till både datortekniska och kognitiva / psykologiska aspekter.

Referensmodeller kan användas för många olika ändamål. Till de vanligaste användningsområdena hör:

- o ramverk för *begreppsbildning* och förståelse
- o *analys* av existerande system / modeller
- o *standardisering*
- o *konstruktion* ("design")

Vi kommer nedan att kort behandla vart och ett av de fyra användningsområdena.

För det första kan en referensmodell tjäna som ett ramverk för att diskutera både *terminologi* och funktionalitet rörande användargränssnittet. Denna aspekt är viktig då det ännu inte existerar vare sig en vedertagen

begreppsapparat eller en klar bild av funktionaliteten i användargränssnittet. En referensmodell för dialoghantering skulle i sig kunna utgöra ett ramverk inom vilken begrepp och funktioner kan definieras, relateras till varandra och därigenom leda till bättre förståelse av området bland berörda parter.

För det andra kan en referensmodell stödja *analys* av olika typer av användargränssnitt samt tekniker för t ex specifikation av interaktionsprocessen mellan människa och dator. Referensmodellen måste vara tillräckligt generell för att kunna tillämpas både på existerande och framtida teknologi. Modellen kan därför t ex inte förutsätta användning av vissa typer av in- / utmatningsenheter. I analysarbetet hjälper referensmodellen till med att dels peka ut relevanta aspekter dels att hålla ihop relaterade respektive hålla isär orelaterade sådana. Analysarbetet förenklas därigenom avsevärt.

För det tredje används referensmodeller vid *standardiseringsarbete*. Detta är förmodligen det vanligaste tillämpningsområdet (jfr OSI-modellen ovan). Modellen används här som en referenspunkt inom vilken olika typer av standarder kan utvecklas. Referensmodellen pekar ut komponenter som kan och bör standardiseras.

För det fjärde, slutligen, kan referensmodeller användas som en bas för *konstruktion* av nya modeller, tekniker och system för människa / dator-interaktion. Ett exempel på detta är dialoghanteringssystem (DHS). Ett DHS är en typ av system för hantering av "allt" som har med människa / dator-dialogen att göra. Denna typ av system bygger idag på en mycket vag referensmodell, eller snarare arkitektur, där interaktionen i olika applikationer har generaliserats och brutits ut ur applikationen. Med en klart definierad referensmodell skulle denna typ av system kunna göras betydligt kraftfullare.

Vilket nämnts ovan har referensmodeller allmänt haft stor betydelse inom olika områden för att strukturera dessa och därmed underlätta både förståelse och vidareutvecklingsarbete. En referensmodell inom området människa / dator-interaktion kan därför förväntas få samma effekt vad gäller utvecklingen av interaktiva system.

MODELLPRESENTATION

Referensmodellen består av en central del och sidoordnade delar. Arbetsgruppen har enbart utvecklat den centrala delen. Sidoordnade delar är hämtade från Martin (9). Den centrala delen består av sex skikt som avgränsas uppåt och nedåt av ett *uppgifts-* skikt respektive ett *fysiskt* skikt. Denna del omges av två sidoordnade delar, en som anger själva problemlösningens funktioner och en som anger säkerhets / kvalitets-aspekter.

Modellen bör tolkas som att en dialog:

- o byggs för att en arbetsuppgift (uppgift) ska kunna utföras.
- o kräver tillgång till hårdvara (fysik) för att kunna utföras.
- o bör ha de sex funktionella skikten realiserade antingen i skelett och / eller applikationsprogram för att fungera tillfredsställande som människa / dator-gränssnitt.
- o bör vara realiserad med hänsyn tagen till den sidoordnade mänskliga problemlösningens processen för att ge ett tillfredsställande stöd.
- o bör vara realiserad med hänsyn tagen till de sidoordnade delarna säkerhets- / kvalitetsaspekter.

Modellens centrala del utgör som sagt kärnan i referensmodellen. Dess olika skikt är funktionellt orienterade och utgår ifrån att en *arbetsuppgift* ska utföras, varefter man väljer *verktyg* för att utföra uppgiften. Hur uppgiften sedan kan lösas framgår av en *dialogstruktur* i vilken man kan navigera eller välja arbetssätt med hjälp av en viss *dialogstil* varefter själva uppgiften hanteras i ett skikt kallat *uppgiftshantering*. Presenterade data *formateras*, t ex amerikansk eller svensk datumangivelse, så att de passar mottagaren som kan välja *interaktionsform* i modellens lägsta skikt. Hela arbetet sker slutligen med hjälp av någon *fysisk* utrustning. (*)

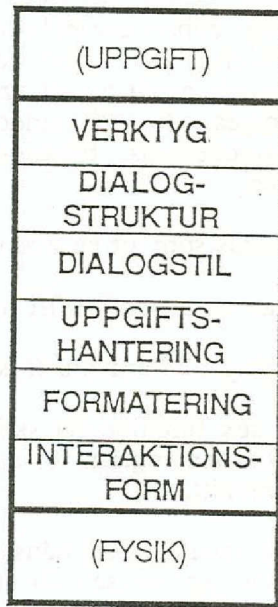
Modellens skikt och dess sidoordnade delar beskrivs närmare under rubrikerna "modellbeskrivning" respektive "sidoordnade delar".

Trots att en interaktion kan ha så skilda syften som registrering, bearbetning, eller återvinning av information, så har vi valt att beskriva skikten generellt för att passa alla dessa olika typer av interaktion.

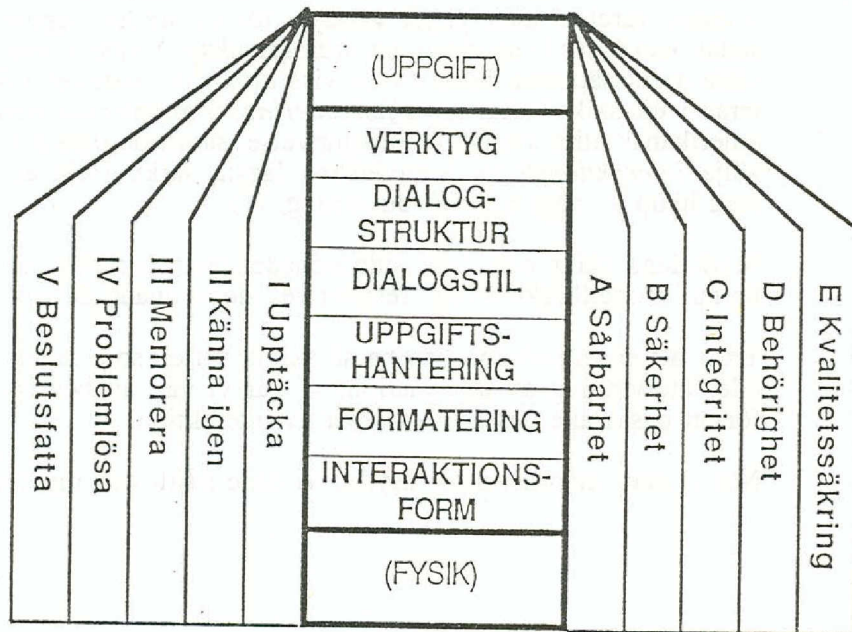
När vi nämner ordet dator åsyftar vi både hård- och mjukvara.

(*) Det bör noteras att preskriptiva uttalanden av typen användaren *bör* kunna välja arbetssätt eller interaktionsform inte hör till själva referensmodellen. Referensmodellen säger ingenting om hur människa / dator-interaktionen skall eller bör implementeras. Påståendena är projektmedlemmarnas åsikter i dessa frågor vilka kopplats till motsvarande skikt i modellen.

MODELLBILD



Figur: Referensmodellens centrala del



Figur: Referensmodellen inklusive sidoordnade delar

MODELLBESKRIVNING

Uppgift

Denna nivå är överordnad och utgör modellens ena ändpunkt. Det som styr användaren till detta skikt är den uppgift som användaren har valt att arbeta med. Först väljs alltså arbetsuppgift och därefter verktyg. Skiktet definierar meningen (semantiken) med det som skall utföras med hjälp av något verktyg. Ett typexempel på en uppgift är att ta emot och registrera order per telefon eller brev.

Verktyg

När användaren beslutat vilken arbetsuppgift denne skall arbeta med väljs det verktyg som krävs för att utföra uppgiften. Olika verktyg kan samverka för att lösa uppgiften. Det är emellertid ej säkert att användaren upplever dem som mer än ett. Verktöget är ett instrument som användaren tar till hjälp för att lösa en uppgift.

Ett verktyg kan vara både ett arbetsredskap för aktiv bearbetning av data och en passiv informationsgivare. Exempel på verktyg är texthanteringsprogram, databashanterare, fönsterhanterare eller diagnos hjälpmedel. Naturligtvis inbegrips även programmeringsverktyg och programvara för en systemadministratör i denna nivå.

Dialogstruktur

På övergripande nivå i en applikation kan man i regel välja mellan att utföra någon eller några av ett antal delfunktioner. Dessa delfunktioner är relaterade till varandra i en struktur som anger hur användaren får förflytta sig mellan delfunktionerna. Denna struktur kallar vi dialogstruktur. Dialogstrukturen kan ses som en del av styrstrukturen i en applikation. Anledningen till att den brutits ut på detta vis är att man i referensmodellen vill kunna behandla de delar av styrstrukturen som är direkt relaterade till interaktionsprocessen separat från andra delar av styrstrukturen.

Det bör noteras att enbart applikationsoberoende styrinformation beskrivs på denna nivå. Det är vanligt att åskådliggöra en dialogstruktur med hjälp av en dialoggraf eller dialogmodell.

Dialogstil

För att navigera i dialogstrukturen måste systemet kunna presentera möjliga vägar i strukturen för användaren. Användaren måste i sin tur kunna ge styrinformation till systemet om vilken väg som valts. Sådana val görs inom ramen för olika stilar, t ex meny och kommando.

Meny, kommando och andra stilar skall här ses som en dialog på ett abstrakt plan. Det är här irrelevant om styrinformation ges i klartext, via funktions-tangenter eller via mus-klickningar (*). (Observera att vi fortfarande rör oss på styrstrukturnivå.) Viktigt är att styrinformationen skall vara tolkningsbar av både människa och dator.

(*) Med *mus* avses vad som ibland kallas *rätta* eller *styrklots*.

Uppgiftshantering

Den konkreta hanteringen av vald uppgift täcks av detta skikt. Arbetet med en uppgift kan spjälkas upp i två delar, en generell del, t ex val av en del-funktion ur en meny, och en applikationsorienterad del, t ex registrering av en ny kunds namn och adress. Den senare omfattar applikationsorienterad hantering och är placerad i detta skikt.

Registrering av data görs som regel i en viss ordning. Man kan också säga att den följer ett visst mönster. Vanligast är att mönstret är helt förutbestämt av systemkonstruktören. I vissa fall kan ett par, tre, alternativa ordningsföljder vara tillåtna. Det borde emellertid vara möjligt för användaren att registrera information i den ordning som passar denne bäst. Registreringsprocessen skulle därigenom få formen av ett dynamiskt och individuellt mönster. Här förvandlas datorn från en hårt styrande automat till en hjälpsam kontrollant som enbart ser till att all nödvändig information blir registrerad. Skulle något glömmas bort får användaren veta vad som saknas. Modellen är öppen för båda tillvägagångssätten.

Uppgiftshanteringen omfattar dels applikationsberoende styrinformation dels datahantering direkt relaterad till en specifik applikation. För att beskriva applikationsberoende styrstrukturer används samma tekniker som för dialogstrukturen ovan, dvs dialoggrafer eller dialogmodeller. Olika stilar för människa / dator-dialogen som här är aktuella är meny, kommando, formulär och fråga-svar.

Formatering

För att kunna presentera styrinformation och utdata inom ramen för t ex menyer och formulär måste dessa formateras. Formatering omfattar ett successivt uppbyggande av en "bild" som går att presentera på en utenhet.

Användaren ser inte resultatet av formateringen i detta skede då denna utförs internt i datorn. Först när formateringen är klar och presenteras på ett utmatningsmedia - bildskärm, skrivare, högtalare el dyl - får användaren del av den formaterade informationen och omvänt. Själva in- och utmatningen ligger dock inte inom detta skikt.

Omvänt måste indata tolkas till ett format som är begripligt för datorn.

Målet med ett särskilt formateringsskikt är att lyfta fram formateringen av data som idag ofta sker implicit i olika program. Därigenom får både programmerare och användare möjlighet att styra denna process. Programmerare kan t ex bygga mer avancerade layouter och användare kan dynamiskt ändra formateringen (jfr ändring av kolumnbredd i matriskalkylprogram). Användaren bör också vid behov i förväg kunna välja hur formateringen skall göras (användarprofil).

Interaktionsform

Användaren kan interagera med datorn på många olika sätt, t ex genom att tala, peka eller skriva. För att kunna hantera olika interaktionsformer måste in- och utmatningen delas upp i tre delar: en del som är oberoende av informationsform (t ex ljud, text, bild), en annan som är enhetsoberoende inom en

viss informationsform (t ex logiska enheter i grafisk databehandling) och en tredje som är enhetsberoende (drivrutiner eller "device drivers").

I dagens interaktiva system finns det sällan möjlighet att använda olika interaktionsformer parallellt. Utvecklingen går dock mot att ett flertal interaktionsformer skall gå att använda samtidigt mot samma verktyg.

Observera att resultatet av formateringsprocessen ovan inte nödvändigtvis går att presentera på den typ av enhet som användaren begagnar för tillfället. Grafiska data kan t ex inte presenteras på en vanlig teckenskärm. Formateringen bör därför automatiskt styras mot den typ av enhet som skall användas vid presentationen. Detta är emellertid inte alltid möjligt. Hur skall t ex en digitaliserad bild konverteras till text ? Konventioner måste därför utvecklas för hantering av dessa fall.

Fysik

Fysik är modellens andra ändpunkt och inrymmer de hjälpmedel som behövs för att kunna arbeta med ett verktyg. De fysiska komponenterna går att se eller ta på. Exempel på fysiska komponenter är bildskärm, digitaliseringsbord (CAD / CAM), mikrofon, mus, tangentbord etc.

SIDOORDNADE DELAR

Modellen är kopplad till två sidoordnade delar, en som innehåller problemlösningsprocessens funktioner och en som innehåller säkerhets- / kvalitetsaspekter. Dessa två sidoordnade delar ingår inte i den centrala delen av referensmodellen men är medtagna på grund av att man i praktiken så gott som alltid måste ta hänsyn till dem vid människa / dator-interaktion.

Problemlösningsprocessen

Processen består av fem delar som alla måste tas hänsyn till för att en uppgift ska kunna hanteras tillfredsställande. De fem delarna, dessas funktioner och påverkan på dialogmodellen beskrivs nedan:

- *Upptäcka*, anger att aktuella delar av de realiserade skikten måste kunna upptäckas, d v s synas, höras eller kännas av den som valt verktyget för att lösa en uppgift. Teckenstorlek, färg samt redigering påverkar denna funktion.
- *Känna igen* innebär att de upptäckta delarna måste kännas igen för att kunna hanteras. Formateringen påverkar denna funktion t ex genom att datum presenteras som år-mån-dag eller som månad-dag-år.
- *Memorera* innebär att människan måste komma ihåg den igenkända situationen och kan alltså inte implementeras i den fysiska utrustningen / programvaran, möjligen kan stöd ges genom eventuella hänvisningar till tidigare ärenden av samma slag. Exempel på hur ett sådant stöd kan realiseras är expertsystem, historielista över hittills utförda kommandon eller genom att visa var i dialogstrukturen användaren befinner sig.
- *Problemlösa* innebär att känna till, veta, hur den ihågkomna situationen ska hanteras och är intimt förknippad med användarens kompetens. Den kan förstärkas genom olika hjälpfunktioner i dialogen eller genom hanteringsregler i handböcker och dylikt.
- *Beslutsfatta*, slutligen, är vändpunkten i denna sidoordnade struktur och innebär att lösningen på aktuellt problem antingen förs tillbaka till dialogen eller förs vidare till någon annan utanförliggande funktion. Beslutet kan innebära att göra något / att inte göra något, eller att göra så / att göra så.

Problemlösningsprocessen har beskrivits som en människa / dator-dialog. Det är därför viktigt att påpeka att den också kan användas vid en renodlad människa / människa-dialog eller i en renodlad dator / dator-dialog och att de olika funktionerna då får en något annorlunda betydelse.

Säkerhets- / Kvalitetsaspekter

Dessa aspekter kan anses speciellt viktig i dialogsituationer på grund av att den fysiska utrustningen ofta är lättillgänglig och att dialogen ofta är snabb och flexibel vad gäller registrering av data.

De fem ingående delarna i denna struktur beskrivs nedan:

- o Med *sårbarhet* menas det fysiska skyddet av anläggningen, t ex skydd mot skadegörelse, sabotage och stöld.
- o *Säkerhet* omfattar allt som har att göra med att hårdvaran fungerar, t ex strömförsörjning och teknisk kvalitet. Mått på säkerheten är bl a lång MBF (Meantime Between Failures) och korta reparationstider.
- o *Integritet* avser känsligheten i presenterade data vare sig dessa avser personuppgifter eller affärsuppgifter. Denna del är applikationsorienterad och måste tas hänsyn till vid realiseringen av samtliga skikt i referensmodellen, den är också intimt förknippad med nästa del.
- o *Behörighet* avser människans rätt att dels använda valt verktyg, dels ta del av hela eller delar av den information som presenteras. Denna del är både fysisk och applikationsorienterad och måste alltså tas hänsyn till både på den fysiska sidan och inom samtliga skikt i referensmodellen. Behörigheten är vanligen kategoriindelad både vad gäller typ av arbetsuppgifter och personal.
- o *Kvalitetssäkring* avser såväl den s k funktionella kvaliteten som den s k tekniska kvaliteten. Detta innebär att hänsyn till kvaliteten måste tas redan vid analysen av den verksamhet som skall använda dialogen så att den vid användandet får de avsedda effekterna i verksamheten. Vidare innebär denna del att dialogen som helhet måste testas så att den fungerar oklanderligt vid användandet och slutligen innebär den inte minst att det i dialogen måste finnas inbyggt kontroll som säkerställer kvaliteten på registrerade och bearbetade data.

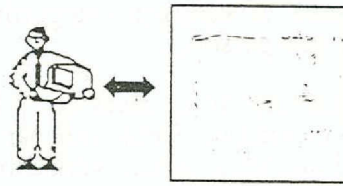
Eftersom dialoger kan vara av olika typ, t ex registrerings-, återvinnings- eller bearbetningsdialoger följer att säkerhets- / kvalitetsaspekten får olika dignitet beroende på vilken typ av dialog som hanteras. Vidare kan dialogen äga rum i olika miljöer, t ex såväl i offentlig som i privat miljö. Detta innebär inte att någon del av den sidoordnade säkerhets- / kvalitetsaspekterna kan utelämnas utan att tyngdpunkten dem emellan kan förskjutas beroende på typ av dialog och vilken miljö den är avsedd äga rum i.

EXEMPEL

En kort rekapitulering

För att illustrera referensmodellens olika nivåer, vilka beskrivits ovan, och dess tillämpbarhet på olika problem ges här några exempel. Exempelen belyser främst modellens användbarhet för att analysera existerande system.

Som tidigare nämnts, är referensmodellen tänkt att fungera som ett övergripande ramverk både för analys och konstruktion av programvara eller av mer detaljerade modeller. Referensmodellen fokuserar därför interaktions-processen mellan människa och dator, utan hänsyn till vilken typ av verktyg användaren för tillfället begagnar. Modellen omfattar därför enbart funktionella aspekter på interaktionsprocessen oberoende av funktionernas realisering i det enskilda fallet.



Figur: Interaktionsprocessen människa-dator

För att visa att referensmodellen inte är beroende av någon specifik typ av interaktivt system har exemplen hämtats från tre olika områden: ett applikationsprogram, ett fönsterhanteringssystem och en kommandoavtolkare för ett operativsystem (OS-shell). Det fysiska skiktet och uppgiftsskiktet berörs endast ytligt då dessa ligger utanför modellens egentliga kärna.

Exemplet på applikationsprogram illustreras med hjälp av en orderhantlingsrutin. Denna beskrivs ingående. Först visas allmänt hur de olika arbetsstegen i applikationen är relaterade till respektive nivå i referensmodellen. Därefter visas hur tre alternativa implementeringar av samma applikation kan relateras till referensmodellen. De resterande två exemplen - fönsterhantering och en kommandoavtolkare för ett operativsystem - beskrivs endast kort.

Exempel 1: Applikationsprogramvara

En vanlig typ av uppgift är orderhantering. Denna uppgift kan realiseras i form av en eller flera applikationer för hantering av kundregister, order, restorder, etc. Vi inleder med att beskriva hur orderhanteringen rent allmänt kan relateras till olika skikt i referensmodellen.

- o *Verktyg*: applikationen (applikationerna) ses här som verktyg vilka tillhandahåller tjänster / funktioner som användaren kan begagna sig av för att lösa sin uppgift. Exempel på sådana tjänster är "registrera order", "registrera kund" och "uppdatera kunduppgifter". Användaren väljer här applikationen "orderhantering" och därmed också indirekt någon eller några av dess tjänster.
- o *Dialogstruktur*: dialogstrukturen i denna tillämpning är tämligen enkel. De övergripande funktionerna kan t ex vara definierade som en hierarki, dvs från högsta nivån kan man välja: "registrera order", "registrera kund" och "uppdatera kunduppgifter". Inom respektive funktion kan sedan mer detaljerade funktioner finnas definierade, t ex "registrera orderrad" och "notera restorder", vilka i sin tur ingår i en dialogstruktur på mer detaljerad nivå. En vanlig restriktion i hierarkiskt organiserade strukturer är att man bara kan röra sig uppåt och nedåt i strukturen, men ej på tvären.
- o *Dialogstil* (*): Val av funktioner som skall anropas i dialogstrukturen görs alltid inom ramen för en viss dialogstil. En vanligt förekommande dialogstil i hierarkiskt organiserade dialogstrukturer som denna är menyer. De olika valmöjligheterna måste presenteras för användaren och utgör styrinformation för denne. Användaren ger i sin tur styrinstruktioner till datasystemet för att indikera vilket av de presenterade alternativen som valts. Om man fritt kan vandra mellan olika funktioner brukar kommandon användas i stället för menyer.

Observera att dialogstil på denna nivå är abstrakt och inte säger något om t ex menyval sker genom utpekning medelst mus, inmatning av ett menyalternativ i klartext eller tryckning på en funktionstangent.
- o *Uppgiftshantering*: Inom vald funktion tar hanteringen av själva uppgiften vid, t ex registrering av alla uppgifter om en ny kund. Också in- och utmatning på denna nivå kan göras inom ramen för menyer och kommandon. Vanligast är dock fråga-svar eller formulär. Uppgiftshanteringen, liksom dialogstilen, är att betrakta som abstrakt, dvs den är oberoende av använda I/U-enheter.
- o *Formatering*: En viktig del av in- och utmatning är presentationsaspekten. Denna berör både formatering av utdata och tolkning av indata. Utdata kan successivt byggas upp till en struktur av en datatyp som går att presentera på en I/U-enhet. Omvänt måste indata transformeras från

(*) Observera att begreppet *dialogstil* används i en något annorlunda betydelse än den gängse. Vanligen avses någon av stilarna meny, kommando, formulär eller fråga-svar oavsett användningsområde. Här avses dock enbart användningsområdet applikationsoberoende val i dialogstrukturen.

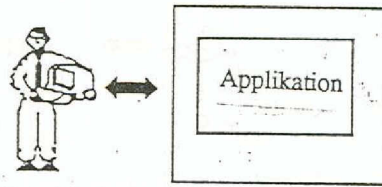
sin externa representation till en av datorn hanterbar internrepresentation. Avbildningen dem emellan behöver inte vara 1:1. Exempelvis kan olika externrepresentationer av datum förekomma, såsom 1987-06-30, 30:e juni 1987 eller 30/6-87. Vid t ex presentation av en tidigare registrerad order måste enskilda dataelement såsom datum, orderrad (artikelnr, antal etc.) successivt formateras och komponeras till en sida som kan presenteras.

- o *Interaktionsform:* Slutligen måste data fysiskt kunna matas in och ut, till och från datorn. In- och utmatning (I/U) kan ses på olika nivåer. På hög nivå skiljer vi bara mellan olika informationsformer, såsom ljud, text och bild. På lägre nivåer måste man ta hänsyn till egenskaper hos olika typer av enheter, t ex grafisk eller teckenterminal, respektive hantering av fysisk I/U för en viss terminaltyp. Målet är att man i en applikation som denna skall kunna använda olika typer av I/U-enheter utan att behöva koda om inmatningsrutinerna. Då skulle man obehindrat kunna byta till en grafisk terminal med fönster och mus, eller en enhet för röstinmatning, i stället för en traditionell teckenterminal.

En applikation och dess interaktionsprocess med användaren kan implementeras på många olika sätt beroende på tillgänglig hård- och mjukvara. Vi skall nedan studera några implementeringsalternativ och relatera dessa till referensmodellen.

Alternativ A (exempel 1)

Det traditionella sättet att implementera en applikation på är att skriva ett stort program som hanterar allt, från utförandet av olika tjänster ("registrera order" etc) till hantering av fysisk in- och utmatning (READ indata INTO orderrad).



Figur: Applikation med "dum" terminal

Enligt referensmodellen täcker applikationsprogrammet in funktioner på samtliga nivåer. Sådana program har en tendens att bli mycket stora och dyra att utveckla. Dessutom finns här en tendens att i applikationen blanda ihop funktioner på olika nivåer. Detta medför i sin tur att olika aspekter löds ihop vilket gör programmen svåra att ändra i med avseende på enskilda aspekter. Exempelvis kan man ej växla mellan olika dialogstilar därför att applikationsprogrammeraren kodat in menyer en gång för alla.

Skikt	Funktioner	Hanteras av
Uppgift	ordermottagning	---
Verktyg	orderhantering, fönsterhantering, OS-shell	Applikation
Dialogstruktur	styrstruktur	
Dialogstil	meny, kommando, formulär, fråga-svar	
Uppgiftshantering	meny, kommando, formulär, fråga-svar	
Formatering	formatering, tolkning	
Interaktionsform	ljud, bild, text	
Fysik	tangentbord, mus, grafiskskärm	---

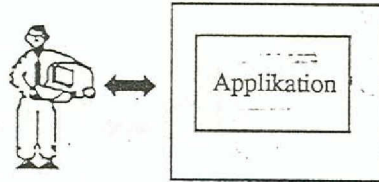
Figur: Referensmodell - applikation med "dum" terminal

Figuren ovan ger exempel på funktioner som måste understödjas och visar hur dessa förhåller sig både till skikten i referensmodellen och till de program eller enheter som hanterar (implementerar) funktionerna. Funktioner markerade med fetstil avser sådana som är aktuella i just detta exempel. Övriga uppräknade funktioner (normal stil) pekar på möjliga alternativ.

Det finns emellertid också andra sätt att implementera samma applikation på. Som vi skall se kan både funktionerna och hanteringen (implementeringen) av dessa varieras.

Alternativ B (exempel 1)

En annan typ av lösning bygger på att en del av funktionaliteten för bl a hantering av in- och utmatning läggs över på en sk intelligent terminal.



Figur: Applikation med "intelligent" terminal

En intelligent terminal hanterar vanligen en hel bildskärmssida åt gången. I denna hantering ingår presentation av av data på bildskärmen. Dessutom kan terminalen ofta ta emot instruktioner om fält på bildskärmssidan i vilka användaren kan mata in data. Terminalen styr användarens förflyttningar mellan de olika fälten och utför dessutom enkla indatakontroller.

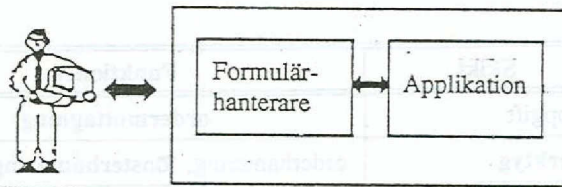
Skikt	Funktioner	Hanteras av
Uppgift	ordermottagning	---
Verktyg	orderhantering, fönsterhantering, OS-shell	Applikation
Dialogstruktur	styrstruktur	
Dialogstil	meny, kommando, formulär, fråga-svar	Intelligent terminal
Uppgiftshantering	meny, kommando, formulär, fråga-svar	
Formatering	formatering, tolkning	
Interaktionsform	ljud, bild, text	
Fysik	tangentbord, mus, grafiskskärm	---

Figur: Referensmodell - applikation med "intelligent" terminal

Som synes fungerar även referensmodellen för att beskriva uppdelningen av funktionaliteten i gränssnittet på olika komponenter, även för komponenter som utgörs av fysiska enheter.

Alternativ C (exempel 1)

En tredje typ av lösning bygger på att en formulärhanterare används för att ta hand om allt som har med användargränssnittet att göra.



Figur: Applikation med formulärhanterare

En formulärhanterare är en programprodukt med vars hjälp man både kan skapa och utföra dialoger. Dessa beskrivs i form av formulär. I formulärdefinitioner beskrivs layouten och indatafält plus indatakontroller på likande sätt som för bildskärmsidor för intelligenta terminaler (se ovan). Dessutom kan man ofta kedja ihop olika formulär genom att till ett formulär koppla villkor och namn på andra formulär, t ex:

IF datumfält = 1987 CALL form-A ELSE CALL form-B.

Det egentliga applikationsprogrammet reduceras därigenom till ett minimum.

Skikt	Funktioner	Hanteras av
Uppgift	ordermottagning	---
Verktyg	orderhantering, fönsterhantering, OS-shell	Applikation
Dialogstruktur	styrstruktur	Formulärhanterare
Dialogstil	meny, kommando, formulär, fråga-svar	
Uppgiftshantering	meny, kommando, formulär, fråga-svar	
Formatering	formatering, tolkning	
Interaktionsform	ljud, bild, text	---
Fysik	tangentbord, mus, grafiskärm	

Figur: Referensmodell - applikation, formulärhanterare

Man kan naturligtvis tänka sig ytterligare implementeringsalternativ utöver dessa tre. Vårt syfte är dock bara att visa på att modellen är opåverkad, och precis lika tillämplig, oavsett vilket implementeringsalternativ som väljs. Därigenom utgör referensmodellen en stabil grund vid analysarbeten av detta slag.

Exempel 2: Fönsterhanteringssystem

Bland olika typer av systemprogramvaror har vi valt ett fönsterhanteringssystem. En fönsterhanterare används som ett verktyg för att organisera information från olika informationskällor på bildskärmen.

Skikt	Funktioner	Hanteras av
Uppgift	ordermottagning	---
Verktyg	orderhantering, fönsterhantering, OS-shell	Fönsterhanterare
Dialogstruktur	styrstruktur	
Dialogstil	meny, kommando, formulär, fråga-svar	
Uppgiftshantering	meny, kommando, formulär, fråga-svar	
Formatering	formatering, tolkning	
Interaktionsform	ljud, bild, text	
Fysik	tangentbord, mus, grafiskskärm	---

Figur: Referensmodell - fönsterhanterare

- o *Verktyg:* Tjänster / funktioner som en fönsterhanterare tillhandahåller är bl a "skapa", "ta bort", "stäng", "flytta", och "ändra storlek" på fönster.
- o *Dialogstruktur:* Dialogstrukturen för ett verktyg av detta slag är vanligen mycket enkel. Vilken som helst av funktionerna kan anropas när som helst av användaren. Den enda förutsättning som gäller är att det måste finnas något objekt att applicera funktionen på, dvs för att anropa "stäng" fönster måste det naturligtvis finnas ett fönster att stänga.
- o *Dialogstil:* Fönsterhanteringsfunktionerna finns alltid direkt eller indirekt tillgängliga. Vanligen måste användaren byta till ett lämpligt kontext för att få direkt tillgång till dessa. Vanligen finns åtminstone tre olika kontexter i ett fönsterhanteringssystem: (1) ett kontext där interaktionen med applikationen sker, (2) ett annat där manipulering av enskilda fönster sker och (3) ett tredje där nya fönster skapas. Val av kontext görs oftast genom att användaren med hjälp av markören pekar: (1) i ett fönster, (2) på ramen till ett visst fönster eller (3) på arbetsytan på vilken fönstren visas.
- o *Uppgiftshantering:* Fönsterhanterare använder sig nästan undantagslöst av menyer för att hantera den interna styrningen (stänga, krympa, flytta fönster).
- o *Formatering:* Denna innefattar bl a generering av menyer med t ex en omgivande ram. Dessutom tolkas här pekning på ett av menyalternativen till motsvarande funktion.
- o *Interaktionsform:* Här sker den egentliga in- och utmatningen inklusive hantering av musen. Ett musklick på ett menyalternativ blir här till en signal på formen <musklick, position på skärmen>.

Exempel 3: Operativsystem

En kommandoavtolkare i ett operativsystem, t ex UNIX-Shell, kan också beskrivas inom ramen för referensmodellen. Skillnaden mot fönsterhanteraren är att tjänsterna / funktionerna som tillhandahålls är betydligt större till antalet och har en betydligt vidare spännvidd. Dessutom är dialogstilen vanligen helt kommandoorienterad.

Skikt	Funktioner	Hanteras av
Uppgift	ordermottagning	---
Verktyg	orderhantering, fönsterhantering, OS-shell	Kommandoavtolkare
Dialogstruktur	styrstruktur	
Dialogstil	meny, kommando, formulär, fråga-svar	
Uppgiftshantering	meny, kommando, formulär, fråga-svar	
Formatering	formatering, tolkning	
Interaktionsform	ljud, bild, text	
Fysik	tangentbord, mus, grafiskskärm	---

Figur: Referensmodell - kommandoavtolkare

Kommentarer

Av exemplen ovan torde framgå att referensmodellen är lika tillämpbar oavsett inom vilken domän den än tillämpas. För att lösa en uppgift kan verktyg av olika slag kombineras, såsom applikationsprogram, formulärhanterare och intelligenta terminaler. Det bör noteras att modellen *inte* försöker specificera gränssnitten mellan dessa verktyg. Antalet möjliga verktyg och kombinationer av dessa är alltför stort. Målet med referensmodellen är i stället att specificera de aspekter på interaktions-processen mellan människa-dator som är relevanta *oavsett* vilken typ av verktyg som används i det specifika fallet.

AVSLUTNING

Den modell som vi presenterat är sprungen ur ett arbete som pågått sedan oktober 1986. Vi som deltagit i att ta fram modellen kommer från olika arbetsplatser och har olika utbildnings- och yrkesbakgrund. Två av oss är data- / systemvetare och två är beteendevetare. Alla fyra har arbetat utifrån olika utgångspunkter mot den gemensamma modellen.

Utarbetandet av förslaget till referensmodell har inte varit helt trivialt. Redan i ett tidigt skede konstaterades att de begrepp som användes stod för vitt skilda saker för olika personer i gruppen. Våra olika bakgrunder förklarar till en del varför olika begrepp från början tolkats så olika.

De första månaderna gick därför åt till att försöka hitta gemensamma begrepp och klara definitioner av dessa. Trots detta inledande arbete upptäcktes under arbetets gång att begreppen vare sig var helt självklara eller heltäckande. Det är först mot slutskedet av arbetet som begreppsapparaten och innehållet i modellens skikt stabiliserats.

Under arbetets gång har även diskuterats var gränsen går mellan referensmodellens centrala del och dess sidoordnade delar. Det har inte heller varit lätt att konstatera vilka delar som är helt sidoordnade respektive finns i modellens centrala del.

Skikten i modellen är inte hierarkiska som det kan tyckas vid en första anblick. För att klargöra skiktens inbördes förhållanden har vi testat modellen i scenarier, d v s försökt att följa olika arbetsuppgifters kopplingar till olika skikt. Vi hade önskat att det vore möjligt att i en modell visa skiktens alla inbördes relationer. Detta har emellertid visat sig vara mycket svårt. I stället har vi låtit exemplen peka ut några sådana kopplingar.

Vi som deltagit i arbetet är medvetna om att denna modell bara är en av flera referensmodeller som just nu arbetas fram. Vad vi känner till är dock inte andra modeller mer utvecklade än den här presenterade. Detta arbetsfält ligger ännu i sin linda. Mycket arbete återstår ännu för att precisera innehållet i dessa modeller och för att klargöra gränssnitten mellan dessas komponenter. Arbetet har bara börjat.

Vår förhoppning är att den av oss föreslagna referensmodellen ska kunna ligga till grund för ett vidareutvecklingsarbete av såväl den centrala som de sidoordnade delarna. I mer utvecklat skick kan modellen såväl ge stadga åt arbetet med att analysera, konstruera och tillverka dialoger som att ge stadga åt de diskussioner som förs kring problem rörande människa / dator-interaktion. I ett längre perspektiv kan referensmodellen eventuellt bilda underlag för standardiseringsarbete liknande OSI-modellen.

Intresserade kan vända sig till någon av författarna / projektmedlemmarna för både klargöranden rörande innehållet i rapporten och i frågor som rör modellens realisering.

REFERENSER

1. Beech, D., "A reference model for command and response languages," in *Proceedings of the IFIP 9th World Computer Congress*, ed. R.E.A. Mason, pp. 793-797, North-Holland, Paris, France, Sept. 19-23, 1983.
2. Bennett, J.L., "The concept of architecture applied to user interfaces in interactive computer systems," *INTERACT'84, First IFIP Conference on Human-Computer Interaction*, pp. 2.156-161, London, 4-7 September 1984.
3. Carson, G.S. and McGinnis, E., "The Reference Model for Computer Graphics," *Computer Graphics and Applications*, vol. 6, no. 8, pp. 17-23, Aug. 1986.
4. Fong, E.N. and Jefferson, D.K., "Reference Models for Standardization," *Computer Standards & Interfaces*, no. 5, pp. 93-98, 1986.
5. Fährnrich, K.-P. and Ziegler, J., "Workstations using Direct Manipulation as Interaction Mode - Aspects of Design, Application and Evaluation," *INTERACT'84, First IFIP Conference on Human-Computer Interaction*, pp. 203-208, London, 4-7 Sept. 1984.
6. Hill, R.D., "Supporting Concurrency, Communication, and Synchronization in Human-Computer Interaction - The Sassafras UIMS," *ACM Transactions on Graphics*, vol. 5, no. 3, pp. 179-210, July 1986.
7. Lindsay, P.M. and Norman, D.A., *Human Information Processing*, Academic Press, London and New York, 1972.
8. Lynch, G. and Meads, J., "In Search of a User Interface Reference Model: Report on the SIGCHI Workshop on User Interface Reference Models," *ACM SIGCHI Bulletin*, vol. 18, no. 2, pp. 25-33, Oct. 1986.
9. Martin, J., *Design of Man-Computer Dialogues*, Prentice-Hall, 1973.
10. Meads, J., "The Standards Factor," *ACM SIGCHI Bulletin*, vol. 18, no. 2, pp. 17-20, Oct. 1986.
11. Moran, T.P., "The Command Language Grammar: A Representation for the User Interface of Interactive Computer Systems," *International Journal of Man-Machine Studies*, pp. 3-50, 1981.
12. Shaw, M., "An Input-Output Model of Interactive Systems," *CHI 86 Conference Proceedings*, pp. 261-273, Boston, April 13-17, 1986.
13. Sisson, N., "Dialogue Management Reference Model," *ACM SIGCHI Bulletin*, vol. 18, no. 2, pp. 34-35, Oct. 1986.