

Människa-datorinteraktion

Teknik, principer, metaforer och modeller Bengt Sandblad

Syfte med lektionen

- Att beskriva grunderna för hur människor interagerar med datorsystem och andra tekniska system, för att utföra en arbetsuppgift. Det handlar om grundläggande tekniker, principer och metaforer samt något kort om teoretiska modeller för att beskriva interaktionen. Vi ska senare komma in på hur gränssnittsdesign går till i praktiken och ge en rad exempel på bra och dålig design.

Innehåll

1. Tekniker för människa-datorinteraktion.

- All interaktion mellan människor och teknik måste gå genom våra sinnen. Människan har i detta avseende ett antal "kanaler" genom vilka informationen kan passera. Vi skiljer mellan "input-kanaler" som vi använder för att hämta in information från omgivningen till våra kognitiva förmågor, och "output-kanaler" som vi använder för att förmedla information ut till datorn och den tekniska omgivningen. Observera att vi här använder begreppen "input" och "output" sett ur människans perspektiv. Sett ur datorns perspektiv blir det förstås tvärtom, input är det som matas in av användaren och output blir det datorn förmedlar till användaren.
- Det finns många anvisningar för hur man kan och bör utforma en bra arbetsplats för datorstött arbete, och hur tekniska system bör anpassas till människan. En mycket bra länk till sådana anvisningar finns via:
 - Arbetslivsinstitutet: <http://www.niwl.se/datorarbete/>
 - Arbetsmiljölagen: http://www.av.se/reoler/arbetsmiljolagen/kapitel2_shtm
 - Arbetskyddsstyrelsens författningssamling: http://www.niwl.se/datorarbete/pdf/1998_05.pdf

2. Interaktionsprinciper och interaktionsstilar

- Det finns ett antal grundläggande principer för hur man kan bygga upp interaktionen, "dialogen", mellan användaren och datorn. Man kan säga att dessa olika principer hänger ihop med den historiska utvecklingen, dvs den tekniska utvecklingen. I datorernas barndom tillät inte tekniken annat än mycket enkel interaktion, medan dagens mer avancerade teknik ger oss många fler möjligheter att utforma interaktionen på. Det som idag till stor del dominerar är de s.k. grafiska användargränssnitten som vi ser i moderna gränssnitt t ex i Windows- och Macintosh-baserade system. De äldre, enklare, principerna ingår ofta som byggstenar i de moderna gränssnitten. Vi ska här bara redovisa olika typer av interaktioner, och kommentarer om vad som är bra eller dåligt kommer senare.

3. Metaforer för användargränssnitt

- Detta avsnitt handlar främst om hur man använder sig av olika metaforer som byggstenar i ett modernt användargränssnitt. Med metafor menar vi en "liknelse", t ex en bild eller ett begrepp, som vi använder för att göra interaktionen enklare, dvs mer förståelig för användaren. I ett gränssnitt ger vi information och operationer fysisk (grafisk) form. Meningen är att skapa bilder som underlättar förståelse, inlärning, utförande, organisation av information m.m. Metaforer kan finnas på flera olika nivåer, t ex som grundläggande byggelement eller som princip för att organisera information på bildskärmen. Exempel på metaforer är ikoner, "drag-and-drop", skrivbordsmetaforen m.m. Metaforer används som begrepp och byggstenar när vi bygger upp ett gränssnitt. Det är viktigt att de metaforer vi väljer att använda oss av i ett visst system kommer att uppfattas som logiska, självklara och stödjande av användarna. En artikel om rumsmetaforen finns tillgänglig separat.

4. Modeller för människa-datorinteraktion

- Här ska vi bara kort, som en orientering, ta upp några teoretiska modeller som kan användas för att beskriva människa-datorsystem. Dessa modeller kan användas dels för syftet att beskriva och förstå hur interaktionen går till, dels för att i samband med utvärdering beräkna hur interaktionen kommer att fungera i praktiken. Vi gör ingen täckande genomgång av sådana teoretiska modeller här, utan avsikten är bara att orientera om några exempel.

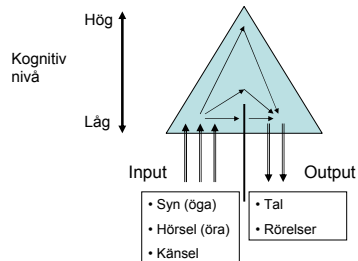
1. Tekniker för människa-datorinteraktion

- All interaktion mellan människor och teknik måste gå genom våra sinnen. Människan har i detta avseende ett antal "kanaler" genom vilka informationen kan passera. Vi skiljer mellan "input-kanaler" som vi använder för att hämta in information från omgivningen till våra kognitiva förmågor, och "output-kanaler" som vi använder för att förmedla information ut till datorn och den tekniska omgivningen. Observera att vi här använder begreppen "input" och "output" sett ur människans perspektiv. Sett ur datorns perspektiv blir det förstås tvärtom, input är det som matas in av användaren och output blir det datorn förmedlar till användaren.

Input- och outputkanaler

- Vi människor använder oss av våra sinnen för att kommunicera med omgivningen.
- Vi använder "input-kanaler" för att ta emot information.
- Mottagen information förs vidare för bearbetning i "det kognitiva systemet". Denna bearbetning kan ske på olika nivåer, t ex:
 - på en låg "automatiserad" nivå (där vi är snabba, reagerar direkt på stimuli och agerar utan direkt medveten tanke, vi har hög parallellkapacitet och arbetar utan någon högre kognitiv belastning),
 - på en hög "medveten" nivå (där vi är förhållandevis långsamma, är "singleprocessande" och gör allt med hög kognitiv belastning).
- Via "output-kanaler" förs information ut till omgivningen.
- De olika kanalerna har också olika kapaciteter och förmåga till parallella processer (se tidigare lektioner).
- Figuren på nästa sida illustrerar detta.

Viktiga kanaler för "input" och "output"



Viktiga kanaler forts.

Sammanfattningsvis ska vi kort diskutera följande informationskanaler, och de tekniker som är aktuella för respektive kanal. Vi diskuterar inte de kognitiva och perceptiva aspekterna, utan har bara detta som grund för att diskutera teknikerna:

- Input (till människan)
 - Syn
 - Hörsel
 - Känsel
- Output (från människan)
 - Tal
 - Rörelser, muskler

Input: Syn

- Med synen hämtar vi in information genom att läsa från t ex bildskärm, paneler etc. Bildskärmen är idag den absolut viktigaste tekniken för information från dator till människa!
- Ögat och synsinnen har vissa egenskaper som man därvid bör beakta, t ex vikten av kontrast vid läsning av text. Gestaltlagarna ger regler för hur vi reagerar på mönster av olika slag. (Mer om detta finns att läsa i litteratur om perception och syn.)
- Olika slags bildskärmar, med olika teknik och olika prestanda, ger olika möjligheter och begränsningar i vad som kan visas.
- Med hjälp av VR-hjälm (Virtual Reality) eller speciella glasögon kan vi producera virtuella världar som visas framför ögonen.
- 3D-teknik av olika slag kan användas för att ge en uppfattning av tredimensionella informationsmängder och rymder.
- Man kan projicera bilder direkt på näthinnan via en extern projektor placerad framför ögat.

Input: Hörsel

- Via ljud av olika slag kan vi göras uppmärksamma på något eller få information från omvärlden.
- Ljud kan t ex vara enkla ljud som "pip" och "pling" etc. som är avsedda att ge en uppmärksamhet på att något sker, som varning m.m.
- Ljud i form av larm av olika slag används ofta i processtyrning m.m. Frågor kring bra utformning av larm och de problem som uppstår i sådana tillämpningar diskuteras kort i andra lektioner.
- Syntetiskt tal är något som kan användas för att förmedla information då man inte kan ha uppmärksamhet på en bildskärm, för synskadade etc.

Input: Känsel

- Vi skiljer mellan taktil och haptisk information. Taktil handlar om känsel, t ex beröring, medan haptisk handlar om en uppfattning av kraft.
- Taktila system för informationsöverföring kan t ex vara system för överföring av text och bilder till blinda. Speciella skrivare för sådant finns, där "piggar" etc. ger ett mönster som kan kännas och avtolkas.
- Haptisk information har vi t ex i sk force-feedback joy-sticks. Här återför man en kraft i styrspaken, som man kan känna av och därigenom få en uppfattning av vilket "motstånd" det finns i det styrda systemet.

Output: Tal

- Via tal kan vi förmedla information till en dator eller ett tekniskt system, t ex för att ge ett kommando, ge en styrorder eller för att registrera data/text.
- Vi kan se två huvudsakliga typer av output genom tal:
 1. Kommandogivning, som fungerar genom att datorn lär sig att känna igen en given mängd ljudbilder, vilka fungerar som kommandon. Exempel på detta är röststyrning av telefoner, biljettbokning, styrning av datorfunktioner etc.
 2. Språkanalys, dvs en analys av det språkliga innehållet i en talad sekvens. Detta är något som idag befinner sig på forskningsstadiet, även om många intressanta system redan finns i praktiken. Vi ska här inte alls gå in på språkteknologin och alla de svårigheter som finns i att göra så att datorn kan tolka och "förstå" naturligt tal, utan hänvisar till andra kurser inom det området. (Se tex: <http://stp.ling.uu.se/educa/> eller http://stp.ling.uu.se/~rloukano/classes/2002s/intro_nlp/index.html)

Output: Rörelse

- Output genom rörelser - dvs att vi ger signaler till muskler som i sin tur styr någon teknisk pryl som ger signaler till datorn. Detta är den huvudsakliga tekniken, vilken i sig har många olika utformningar. (Egentligen är även tal en sådan teknik, där vi med muskler i talorganen ger ljudimpulser till en mikrofon som i sin tur leder till digitala signaler till datorn. Vi väljer dock att se tal som en egen kanal)
- Vi kan utnyttja många olika tekniker för att överföra muskelrörelser till data in i datorn. Följande är några huvudtekniker (se mer i kursboken):
 - Tangentbord: Med detta menar vi all "knappsatser" där vi genom knapptryckningar ger signaler till datorn. Flera olika typer och designers finns:
 - Vanliga tangentbord. Finns i olika design och ergonomisk utformning. Den vanligaste designen kallar vi QWERTY efter hur bokstavslängderna är placerade på första raden. Många andra designers har utvecklats men haft svårt att nå kommersiell framgång.
 - Speciella tangentbord för vissa arbetsuppgifter eller användargrupper. T ex för blindskrift, där flera knapptryckningar samtidigt blir ett tecken, varvid antalet knappar minimeras.
 - Andra knappsatser, tex telefonens knappar för dialog med bankkonton etc.

Output: Rörelse forts.

- Pekdon av olika slag. Olika tekniker finns. Ingen teknik kan sägas vara bättre än den andra generellt, men de kan fungera olika bra i olika arbetssituationer. I varje enskilt fall bör man därför utvärdera vilken teknik man ska använda.
 - Mus. Det allra vanligaste idag, ingår som standard i de flesta datorsystem. Kan oftast ställas in med olika parametrar för att anpassas till individuella krav.
 - Joy-stick. Kan ge noggrann absolutpositionering snabbt. Används t ex ofta i spel och i processtyrningssystem.
 - Rullboll. Är fix i sin position.
 - Touch-teknik, tex tryckkänslig skärm eller "touch-pad" på en bärbar dator. Kan användas för inmatning direkt på skärmen.
- Gester. Kan användas för att genom en gest förmedla information till datorn. Exempel är hur man skriver tecken på en beröringskänslig yta på en PDA, handdator, med hjälp av en pinne. Rörelsen, gesten, tolkas av systemet till ett tecken, ord eller kommando. En annan gestteknik är en "data-glove", eller VR-handske, där rörelser med fingrar, händer och armar tolkas till data av systemet.
- Ögonrörelser och huvudrörelser. Det finns fungerande tekniker för att läsa av huvudets och ögats rörelser. Man kan t ex registrera på vilken plats i rummet ögat är fokuserat. Detta kan t ex användas för att med blicken styra processer, att skriva på ett "virtuellt" tangentbord som visas i synfältet osv. Används t ex i processtyrning och för förlämnade personer som har kvar rörelseförmågan i ögonen.

2. Interaktionsprinciper och interaktionsstilar

- Det finns ett antal grundläggande principer för hur man kan bygga upp interaktionen, "dialogen", mellan användaren och datorn. Man kan säga att dessa olika principer hänger ihop med den historiska utvecklingen, dvs den tekniska utvecklingen. I datorernas barndom tillät inte tekniken annat än mycket enkel interaktion, medan dagens mer avancerade teknik ger oss många fler möjligheter att utforma interaktionen på. Det som idag till stor del dominerar är de s.k. grafiska användargränssnitten som vi ser i moderna gränssnitt t ex i Windows- och Macintosh-baserade system. De äldre, enklare, principerna ingår ofta som byggstenar i de moderna gränssnitten. Vi ska här bara redovisa olika typer av interaktioner, och kommentarer om vad som är bra eller dåligt att använda i olika slags tillämpningar kommer senare.

Interaktionsstilar - dialogtyper

Man kan definiera två olika grundsynsätt på hur människor interagerar med datorer:

- Människan och datorn är två dialogparter i ett samspel
- Människan styr datorn som ett verktyg för att utföra en viss arbetsuppgift.

Vi väljer här att främst använda det senare synsättet, eftersom vi huvudsakligen är intresserade av datorn som hjälpmedel i ett arbetssammanhang. Här är det mindre fruktbart att se människan och verktyget som jämställda, utan målet är en enkel och effektiv styrning av arbetsverktygen.

Man kan mycket väl tänka sig situationer där man kan se människan och datorn som "jämspelare" dialogparter, t ex i spel, men här väljer vi alltså främst det senare synsättet ovan.

Dialogtyper

- Vi ska kort beskriva följande dialogtyper, för att sedan mer utförligt beskriva den sista:
 - Fråga-svardialoger
 - Menystyrning
 - Kommandospråk
 - Formulärinmatning
 - Naturligt språk
 - Grafiska, direktmanipulerade gränssnitt
- All interaktion har som mål att minska "gapet" mellan användarens mål för det man vill utföra, och möjligheterna att få det utfört i systemet, samt "gapet" mellan systemets beteende och användarens mål.

Fråga-svardialoger

- I datorernas barndom skrevs dialogerna i form av ett program som ställde frågor. Användaren fick svara, t ex ja eller nej eller annat svar, och beroende på vad användaren matade in gick programmet vidare i exekveringen.
- T ex:

Datorn	Användaren
Vill du mata in ny person?	Ja
Ange personnummer	470527-5656
Ange namn	Kalle Anka
Ange gatuadress	Fiolgränd 18
Etc.....	

Menystyrning

- Menystyrning användes i tidiga system, och idag är menyer av olika slag mycket vanliga beståndsdelar i nästan alla gränssnitt.
- Menyer innebär att datorn anger ett antal möjliga val, och användaren väljer ett av alternativen.
- I dagens gränssnitt används ofta menyer både för navigering och för val av alternativ och datainmatning.
- Menyer kan organiseras i hierarkier, som sk. pull-down eller pop-up menyer etc.
- På den bildskärm du just nu arbetar ser du säker redan ett antal exempel på menyer!
- Menyer är effektiva eftersom de bygger på igenkänning (recognition), vilket innebär att vi inte behöver lära oss alla valmöjligheter utantill, utan kan se vad som finns tillgängligt och välja det som stämmer med våra mål.

Kommandospråk

- Kommandospråk är formella språk, vilka i en grammatik definierar vad som kan uttryckas i språket.
- Kommandon kan användas på operativsystemnivå, för att styra datorns grundfunktioner, eller inne i ett program/tillämpning för att ange vad man vill ha utfört.
- Användaren måste kunna det formella språket väl för att veta hur man ska uttrycka sig. Detta fordrar mycket inläring och träning. När man väl kan språket är det ofta mycket kraftfullt.
- Exempel på kommandospråk är MS/DOS eller UNIX.

Exempel: - del 13-91
 - mov curs r23 u44
 - http://stp.ling.uu.se/~rioukano/classes/2002s/intro_nlp/index.html

Formulärinmatning

- Ofta består tillämpningar av att användaren ska mata in en viss mängd data som systemet behöver i ett visst läge. Detta kan då göras genom fält i ett "formulär". Ofta är detta idag en del av många tillämpningar även i moderna system.
 - Navigering mellan fält kan t ex ske genom pilar, tab-tangent etc.
- T ex:

A screenshot of a web form with a light blue background. It contains five input fields: 'Namn', 'Pnr', 'Gata', 'Postnr', and 'Postadress'. The 'Postadress' field is a single-line text area. There is a yellow button labeled 'OK?' in the bottom right corner.

Naturligt språk

- Vi ska här inte alls gå in på detta, utan hänvisar till vad vi sagt tidigare.
- Syftet är att skapa en dialog där vårt naturliga språk, skrivet eller talat, kan användas av såväl användare som dator.
- Detta är idag möjligt endast med mycket stora begränsningar, och måste ses som ett forskningsområde snarare som färdigt för tillämpningar. För vissa specifika tillämpningar och användargrupper finns dock idag mycket intressanta försök.

Grafiska direktmanipulerade..

- Med sin grund i forskning främst vid Xerox-laboratorier (STAR-projektet) under 1970-talet utvecklades grunderna för de moderna grafiska användargränssnitten. Utvecklingen fortsattes av Apple, IBM och MicroSoft m fl.
- Ett antal grundidéer utgör basen i dessa gränssnitt:
 - Allt har en grafisk form, man representerar allt i form av bilder av olika slag (ikoner, fönster, knappar, timglas, visare, scrollningslistor.....)
 - Man ska kunna se vad som sker på ett korrekt sätt, t ex textens layout i ett dokument etc. (WYSIWYG-what you see is what you get)
 - Användaren manipulerar objekten på skärmen genom att peka, klicka etc. antingen med mus eller med hjälp av kortkommandon via tangentbord (direktmanipulation). Man utnyttjar alltså *recognition* fullt ut.
 - All dialog är (bör vara) händelsestyrd, dvs användaren styr enväldigt över vad man vill göra och när. Ingen dialog ska vara styrd av datorn, om man inte uttryckligen väljer att göra den styrd, t ex av säkerhetsskäl.
 - Mer om uppbyggnaden av grafiska direktmanipulerade gränssnitt kommer senare i kursen.

3. Metaforer för användargränssnitt

- Detta avsnitt handlar främst om hur man använder sig av olika metaforer som byggstenar i ett modernt användargränssnitt. Med metafor menar vi en "liknelse", t ex en bild eller ett begrepp, som vi använder för att göra interaktionen enklare, dvs mer förståelig för användaren. Man kan associera bilden med något man känner igen, och förstår då förhoppningsvis direkt vad man kan och bör göra. I ett sådant gränssnitt ger vi information och operationer *fysisk (grafisk) form*. Meningen är att skapa bilder som underlättar förståelse, inläring, utförande, organisation av information m.m. Metaforer kan finnas på flera olika nivåer, t ex som grundläggande byggelement eller som princip för att organisera information på bildskärmen. Exempel på metaforer är ikoner, "drag-and-drop", skrivbordsmetaforen m.m. Metaforer används som begrepp och byggstenar när vi bygger upp ett gränssnitt. Det är viktigt att de metaforer vi väljer att använda oss av i ett visst system kommer att uppfattas som logiska, självklara och stödande av användarna.

Olika slags metaforer

- Metaforer kan förekomma på flera olika nivåer.
- Tex på en låg nivå i form av ikoner som avbildar något som användaren kan använda sig av: en mapp, ett dokument, en papperskorg, ett program, en kalkylator, ett ritverktyg, ett fönster etc.



- På operations-nivå, dvs man vill få något utfört. T ex "drag-and-drop" för att kasta dokument i papperskorgen, starta program genom "dubbel-klickning" etc.
- På huvudnivå, som princip för hur man organiserar hela informationsrymden. Se nästa sida!

Skrivbord och rum

- Det finns några huvudsakliga metaforer för hur vi väljer att se på hur hela informationssystemet organiseras. Vi ska beskriva två sådana: Skrivbordsmetaforen (desktop) samt Rumsmetaforen (rooms).
- Skrivbordsmetaforen är välkänd. Vi ser bildskärmsytan som ett skrivbord där vi kan lägga allt som finns tillgängligt: dokument, mappar, program, hjälpmedel etc. Allt ligger mer eller mindre synligt eller dolt av andra objekt, beroende på hur väl vi "stådat" på skrivbordet. Din bildskärm just nu är förmodligen ett "skrivbord"! Skrivbordsmetaforen bygger på att allt finns tillgängligt någonstans, användaren rör över var saker finns, vad som ligger "överst", hur man organiserar, stoppar in i mappar osv. Skrivbordet är ett generellt system där man kan göra allt som går att göra, men ingenting är direkt förberett för en viss funktion, vilket är denna metafors nackdel!
- Rumsmetaforen är principiellt annorlunda!

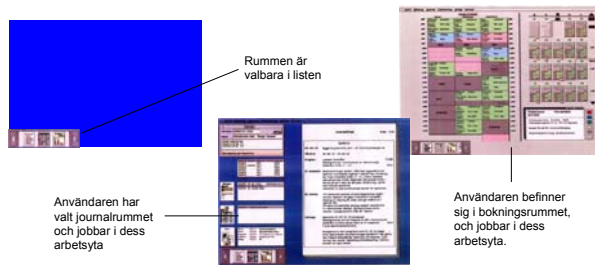
Rum

- I rumsmetaforen ser man informationsvärlden som indelad i ett antal *rum*. Vilka rum man definierar beror på vad man vill kunna göra i systemet. Principen är att man definierar ett speciellt rum för varje huvudsyssla som en viss användare ska kunna utföra i det arbete som systemet ska stödja. Man ska kunna slutföra en arbetsuppgift i det rum man "gått in i".
- I varje rum finns en förberedd *arbetsyta* som är speciellt färdigdesignad för att kunna göra exakt de arbetsmoment som rummet är gjort för att stödja. Rumsmetaforen är lämplig i tillämpningar där användare vill ha direkt tillgång till effektiva färdigdesignade arbetsytor.
- Vill man göra andra uppgifter går man in i de rum som stödjer de önskade uppgifterna.



Rum – ett exempel

I exemplet nedan, avsett för en sjukvårdsenhet, finns bl a två rum, ett för journalläsning och ett för tidsbokning av patientbesök. Journalrummet innehåller en förberedd och specialdesignad arbetsyta för läsning av journaler. Bokningsrummet innehåller hjälpmedel för effektiv tidsbokning. Rummen väljs i "rums-listan" längst ner på skärmen.

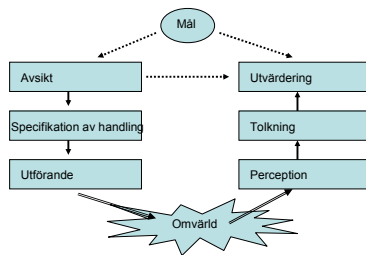


4. Modeller för människa-datorinteraktion

- Här ska vi bara kort, som en orientering, ta upp några teoretiska modeller som kan användas för att beskriva människa-datorsystem. Dessa modeller kan användas dels för syftet att beskriva och förstå hur interaktionen gått till, dels för att i samband med utvärdering beräkna hur interaktionen kommer att fungera i praktiken. Vi gör ingen täckande genomgång av sådana teoretiska modeller här, utan avsikten är bara att orientera om några exempel.
- Modellerna används t ex för att beskriva och förstå hur människor interagerar med datorer och andra tekniska system, eller för att beräkna hur interaktionen kommer att gå till, för att optimera dialogen osv.
- Vi tar några exempel på sådana modeller.

Normans "7 steg"

- Modellen beskriver de kognitiva stegen inblandade i processen att utföra en arbetsuppgift. Uppgiften kan vara "atomär", t ex att trycka på en knapp, eller mer "makro-betonad", t ex en hel arbetsuppgift, vilken då består av en sekvens av deluppgifter.

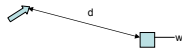


GOMS-modeller

- GOMS står för Goals, Operations, Methods and Selection rules. Syftet med modellen är att beskriva och analysera hur interaktionen går till. Den kan appliceras på olika nivå, från knapptryckning till uppgiftsnivå. Det man beskriver är sättet att utföra en uppgift, inklusive vad man vill göra, stegen i genomförandet samt vilka alternativa sätt att genomföra uppgiften som finns.
- Goms-analyser kan genomföras under design, för att välja bra designlösningar, eller efter implementationen för att utvärdera hur det fungerar för användarna.
- Stegen i analysen är:
 - Mål: Vad man vill genomföra
 - Operationer: stegen i att utföra uppgiften
 - Metoder: sekvensen av steg för att göra klar uppgiften
 - Alternativa lösningar: vilka olika metoder finns? Vilken är bäst?

KLM-metoder

- KLM – Keystroke level methods är ett samlingsnamn på metoder som används för att beräkna t ex hur lång tid olika basala interaktionsoperationer tar att genomföra. Syftet är att få ett underlag för att välja rätt metoder för en viss uppgift, dvs att optimera dialogen. Jämför GOMS ovan. KLM ger numeriska värden för utvärderingen av alternativa designlösningar.
- Grundläggande parametrar rör bl a olika kognitiva cykeltider, dvs tider för perception, kognition och motoriska aktiviteter. Det finns tabeller som anger kvantitativa värden på sådana storheter för olika slags användare.
- Ett exempel är Fitt's lag som beskriver den tid det tar att flytta muspekaren från en punkt på skärmen till en träffyta av viss storlek:
 - $T_{pos} = 100 \log_2(1+d/w)$
 - För en rörelse på $d=10$ cm och en träffyta med bredden $w=0.2$ cm ger detta en positioneringstid på 0.57 sek.



Att fundera över:

- De tekniker för styrning av datorstöd som brukar kallas "pekdon" kan ha många olika slags teknisk utformning. Mus, rullboll, joy-stick, touch-pad etc. Olika tekniker är olika effektiva beroende på hur arbetsplatsen, arbetsituationen och användarna ser ut. Ge några exempel på arbetsituationer där man tydligt kan peka ut en viss teknik som mer lämplig än andra. Motiver dina synpunkter.
- Det finns många olika slags utformning av tangentbord (tangentbordslayout). Den vanligaste, som vi oftast har på våra datorer, brukar kallas QWERTY. Vilken är historien bakom dess tillkomst (sök referens och diskutera hur den kom till)? Är det sant att denna utformning är gjord för att minimera skrivhastigheten?
- Ta som exempel något generellt system som du känner till, t ex MS-Word eller annat. Vilka metaforer innehåller detta system? Anser du att metaforerna är väl valda? Förståeliga? Stödjer de en effektiv användning?
- För vilka slags tillämpningar anser du att rums-metaforen kan vara extra lämplig? Ser du några tillämpningar där rums-metaforen är mindre lämplig?
