

Emma Bosson Erika Svensson

Krav på användbarhet

Att relatera användbarhet till funktionalitet
vid kravspecifisering

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystemteknik Box 1165 581 11 Linköping	Rapportnummer, ISRN FOI-R--0252--SE	Klassificering Underlagsrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år December 2001	Projektnummer E7737
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41 Ledning med samband och telekom	
	Författare/redaktör Emma Bosson Erika Svensson	
Projektledare Johan Fransson		Godkänd av
Tekniskt och/eller vetenskapligt ansvarig Johan Fransson		
Rapportens titel Krav på användbarhet Att relatera användbarhet till funktionalitet vid kravspecifisering		
Sammanfattning (högst 200 ord) Denna rapport är en studie av användbarhetskrav vid kravspecifisering och ämnar utreda frågeställningen: Hur relaterar man användbarhetskrav till funktionella krav i en systemutvecklingsprocess? Frågan utreds genom en djupstudie av litteratur. Teorier som behandlas är bl.a. Software Engineering, Usability Engineering, objektorienterad systemutveckling och ISO-standarder. En jämförelse och diskussion kring vad de olika teorierna säger resulterar i en modell över hur användbarhet relaterar sig till andra krav i en kravspecifikation.		
Nyckelord Användbarhet, användbarhetskrav, kravspecifisering		
Övriga bibliografiska uppgifter Även publicerad av Lunds Universitet som kandidatuppsats	Språk Svenska	
ISSN 1650-1942	Antal sidor: 47 s.	
Distribution enligt missiv	Pris: Enligt prislista Sekretess	

Issuing organization FOI – Swedish Defence Research Agency Command and Control Warfare Technology P.O. Box 1165 SE-581 11 Linköping	Report number, ISRN FOI-R--0252--SE	Report type Base data report
	Research area code 4. C4ISR	
	Month year December 2001	Project no. E7737
	Customers code 5. Contracted Research	
	Sub area code 41 C4I	
Author/s (editor/s) Emma Bosson Erika Svensson	Project manager Johan Fransson	
	Approved by 	
	Scientifically and technically responsible Johan Fransson	
Report title (In translation) Usability requirements – How to relate usability to functionality in requirements specification		
Abstract (not more than 200 words) The report is a study of usability requirements in requirements engineering and has the purpose to analyse how usability requirements relate to functional requirements in system development processes. The method used is a literature study. Theories that are brought up are: software engineering, usability engineering, object-oriented system development and ISO-standards. A comparison and discussion of the theories result in a model showing how usability requirements relate to functional requirements		
Keywords Usability, usability requirements, requirements gathering		
Further bibliographic information Also published at the University of Lund as a bachelor paper	Language Swedish	
ISSN 1650-1942	Pages 47 p.	
	Price acc. to pricelist Security classification	

Krav på användbarhet

**Att relatera användbarhet till funktionalitet vid
kravspecifiering**

Kandidatuppsats, 10 poäng, informatik

Framlagd: augusti, 2001

Författare: Emma Bosson
 Erika Svensson

Handledare Michael Potok

Krav på Användbarhet

Att relatera användbarhet till funktionalitet vid kravspecificering

© Emma Bosson & Erika Svensson
Kandidatuppsats framlagd juni 2001
Omfång: ca 40 sidor
Handledare: Michael Potok

Resumé

Denna uppsats är en studie av användbarhetskrav vid kravspecificering och ämnar utreda frågeställningen: Hur relaterar man användbarhetskrav till funktionella krav i en systemutvecklingsprocess? Frågan utreds genom en djupstudie av litteratur. Teorier som behandlas är bl.a. Software Engineering, Usability Engineering, objektorienterad systemutveckling och ISO-standarder. En jämförelse och diskussion kring vad de olika teorierna säger resulterar i en modell över hur användbarhet relaterar sig till andra krav i en kravspecifikation.

Att ta fram modellen är ett uppdrag från Totalförsvarets Forskningsinstitut där modellen ska användas i ett systemutvecklingsprojekt.

Nyckelord

Användbarhet
Användbarhetskrav
Kravspecificering

Key words

Usability
Usability requirements
Requirements gathering

Innehållsförteckning

1 INLEDNING.....	7
1.1 BAKGRUND	7
1.2 PROBLEMMOMRÅDE	8
1.2.1 Problemformulering	8
1.2.2 Berörda ämnesområden	8
1.2.3 Avgränsningar.....	9
1.3 SYFTE	9
1.4 METOD.....	10
1.5 UPPSATSENS DISPOSITION	12
2 TEORI.....	13
2.1 SOFTWARE ENGINEERING - SOMMERVILLE.....	13
2.1.1 Hur definieras användbarhet?	15
2.1.2 Var kommer användbarhetskrav från?.....	15
2.1.3 Hur används användbarhetskrav i Software engineering?	16
2.1.4 Hur relateras användbarhet till andra krav?	16
2.2 THE USABILITY ENGINEERING LIFECYCLE - MAYHEW	17
2.2.1 Hur definieras användbarhet?	18
2.2.2 Var kommer användbarhetskrav från?.....	18
2.2.3 Hur används användbarhetskrav i Usability Engineering Lifecycle?.....	20
2.2.4 Hur relateras användbarhetskrav till andra krav?	20
2.3 ISO 9241-11	20
2.3.1 Hur definieras användbarhet?	21
2.3.2 Var kommer användbarhetskrav från?.....	22
2.3.3 Hur används användbarhetskrav?	23
2.3.4 Hur relateras användbarhetskrav till andra krav?	23
2.4 ADDING USABILITY - MAGNUS LIF.....	23
2.4.1 Hur definieras användbarhet?	24
2.4.2 Var kommer användbarhetskrav från?.....	25
2.4.3 Hur används användbarhetskrav i UIM?.....	26
2.4.4 Hur relateras användbarhetskrav till andra krav?	26
3 DISKUSSION OCH RESULTAT.....	27
3.1 RESULTAT AV LITTERATURJÄMFÖRELSE.....	27
3.1.1 Hur definieras användbarhet?	27
3.1.2 Var kommer användbarhetskrav från?.....	29
3.1.3 Hur används användbarhetskrav?	30
3.1.4 Hur relateras användbarhetskraven till andra krav?.....	31
3.2 SAMMANFATTNING AV LITTERATURJÄMFÖRELSE	31
3.2.1 Funktionalitet och användbarhet	31
3.2.2 Design och användbarhet.....	32
3.3 FÖRSLAG PÅ MODELL	33
3.3.1 Modellen.....	33
3.3.2 Analys av domän	34
3.3.3 Mål med systemet	35
3.3.4 Krav på systemet	36
3.3.5 Krav på gränssnitt.....	39
3.3.6 Krav på implementation	40
3.3.7 Användbarhet i modellen.....	41
3.3.8 Exempel på kravhantering.....	42
4 SLUTSATS.....	44
REFERENSFÖRTECKNING	

1 Inledning

Det finns många olika metoder och modeller för systemutveckling och därav ett antal angreppssätt med betoning på användbarhet. Vikten av sådana angreppssätt blir allt mer tydligt i ett samhälle där människan i allt fler sammanhang, både i arbetslivet och privat, interagerar med tekniska system.

Att det är viktigt att ta hänsyn till användare och användbarhet är numera ett faktum inom systemutveckling. Användare behöver inte längre acceptera låg användbarhet eftersom det idag finns både möjligheter och kunskap att utveckla användbara system. Trots detta utvecklas det fortfarande system som inte är tillräckligt användbara vilket visar att en fortsatt utveckling av metoder och teorier kring användbarhet är nödvändig. En aspekt att undersöka är hur man hanterar användbarhet vid kravspecifiering.

Uppsatsen faller inom ramen för ett uppdrag från Totalförsvarets Forskningsinstitut (FOI) som i korthet innebär att undersöka hur krav relaterar till varandra i en kravspecifiering. Att ta del i ett utvecklingsprojekt av detta slag innebar för vår del att sätta sig in i en helt ny verksamhet och yrkesområde vilket är en nyttig erfarenhet för oss som blivande systemvetare.

1.1 Bakgrund

Försvarmakten har gett Försvarets Materielverk (FMV) i uppgift att utveckla ett nytt stridsledningssystem där Totalförsvarets Forskningsinstitut (FOI) bistår Försvarmakten och FMV i utvecklingsarbetet. Systemet ska användas för ledning av stridsfordon på bataljonsnivå och är således ett vagnburet system. Försvarmakten har tagit fram en funktionsmodell av systemet som har testats av en grupp användare. Resultatet därav har utvärderats och ska tillsammans med en sammanställning av organisatoriska krav på systemet ligga till grund för en kravspecifikation till det slutliga systemet. Utvecklingsprojektet har ett tydligt fokus på användbarhet och användarmedverkan vilket påverkar de systemutvecklingsmetoder som används. Ett problem som har upptäckts är svårigheten med att formulera användbarhetskrav utifrån erfarenheter från funktionsmodellen. Ett generellt problem, som FOI också upplever, med den ISO-standard som behandlar

användbarhet är att den inte är tillräckligt tydlig i hur kravhantering ska formaliseras. FOI anser det viktigt att kunna sätta krav på användbarhet och inte lämna beslut om design till de som implementerar systemet. För att underlätta arbetet med att specificera användbarhetskrav vill man ta fram en modell som visar hur användbarhetskrav relaterar till funktionella krav. Modellen ska vara en meta-modell, d.v.s. en modell som visar hur man kan modellera krav. Vi har fått i uppgift att med utgångspunkt från det ovan beskrivna projektet utveckla ett förslag på denna modell. FOI anser att ett systemvetenskapligt perspektiv med god förståelse för systemutveckling och användbarhet är lämpligt för en undersökning av detta slag.

1.2 Problemområde

1.2.1 Problemformulering

Den frågeställning vi kommer att utreda är: Hur relaterar man användbarhetskrav till funktionella krav i en systemutvecklingsprocess?

Frågeställningen syftar till att utreda hur användbarhet förhåller sig till funktionalitet vilket innebär en undersökning av olika synsätt på funktionalitet och användbarhet. Utifrån dessa olika synsätt syftar frågeställningen också till att utreda hur relationen mellan användbarhet och funktionalitet uppstår och ser ut.

1.2.2 Berörda ämnesområden

För att kunna utreda frågeställningen behöver man inhämta kunskap från en rad olika ämnesområden. De vi främst kommer beröra är:

- Software Engineering
- Requirements Engineering
- Human Computer Interaction
- Usability Engineering
- Design

Software Engineering är ett ämnesområde som behandlar hela systemutvecklingsprocessen varav kravhantering är en del. Kravhanteringen i Software Engineering benämns Requirements Engineering och är den första fasen i en systemutvecklingsprocess. Usability Engineering är en riktning som fäster större vikt vid användare och användbarhet i systemutveckling än vad traditionella systemutvecklingsmetoder gör. Human Computer Interaction behandlar teorier om hur

människor interagerar med datorer. Design är principer för framställning och formgivning och i det här fallet intressant vid utformning av gränssnitt.

Frågeställningen förutsätter att man har en helhetsbild av systemutveckling. Vidare måste man se till vad krav är och hur man specificerar krav. Detta i sin tur innebär att man behöver veta hur man analyserar delar av eller en hel verksamhet och därur kan dra slutsatser om vad som förväntas av ett framtida system. Att veta hur man specificerar krav innebär också att man måste veta hur de ska formuleras för att kunna implementeras och verifieras. Dessutom behövs kunskap om designprinciper då kraven ska realiseras i designen.

1.2.3 Avgränsningar

Vi kommer inte att göra en heltäckande analys av alla de ämnesområden som frågeställningen berör. Vi har valt att studera fyra perspektiv närmare; Software Engineering, Usability Engineering, ISO 9241-11 och UIM enligt Magnus Lif. Varje ämnesområde och metod introduceras kort för att ge en översikt och analyseras och diskuteras därefter. Vi kommer endast att använda de metoder och aspekter av respektive ämnesområde som berör användbarhet och specificering av krav. Eftersom fokus är på användbarhetskrav kommer inte funktionella krav diskuteras i samma utsträckning i litteraturgenomgången.

Frågeställningen berör även hur man kopplar funktionella krav till organisatoriska krav. Denna aspekt kommer vi dock inte att behandla i någon större utsträckning eftersom fokus, som tidigare nämnts, ligger på användbarhet.

1.3 Syfte

Det finns mycket publicerat material inom de berörda ämnesområdena, dock finns det lite material om hur man integrerar användbarhet i en utvecklingsprocess och hur man går från kunskapen om *vad* användbarhet är till *hur* man realiserar det. Detta är emellertid två stora och komplexa frågor som vi inte har någon möjlighet att helt utreda men hoppas på att kunna bidra till arbetet med att besvara dem.

Syftet med detta arbete är att göra en djupstudie i litteratur och att utifrån den ta fram en modell. Modellen syftar inte till att visa hur användbarhet kan integreras i en systemutvecklingsprocess utan till att visa hur användbarhetskrav relaterar till funktionella krav. Genom att få en alternativ bild över hur olika krav relaterar till varandra är det vår förhoppning att därmed kunna lyfta fram användbarhetskraven och göra det lättare att ge användbarhet en större vikt i kravspecifikation och systemutveckling.

Att ta fram modellen är ett uppdrag från FOI. Modellen kommer att användas vid sammanställning av krav i en kravspecifikation och syftar till att underlätta hanteringen av användbarhetskrav i processen. Vid arbetets början ifrågasatte vi inte syftet med och huruvida det fanns ett behov av en sådan modell. Innan vi hade satt oss in i ämnet var det svårt att se dels hur relevant en modell av det här slaget skulle vara och dels om den verkligen saknades. Om vi under arbetets gång hade upptäckt att det finns en modell som täcker vårt problem eller om vi hade upptäckt att det inte finns något behov för modellen hade vi inte fortsatt att utreda frågeställningen. Nu fann vi dock stöd på många ställen i litteraturen för det faktum att det finns glapp på olika ställen inom utvecklingsprocessen (Lif 1998, s. 86; Mayhew 1999, s. 16; Sommerville 1998, s. 64).

Tyvärr har vi ingen möjlighet att utvärdera resultatet av användningen av modellen i denna uppsats då användningen av den är planerad till augusti-september, d.v.s. efter att denna uppsats färdigställts. Eftersom syftet med uppsatsen är att utifrån befintlig litteratur utreda hur användbarhet förhåller sig till funktionalitet och modellen är ett resultat av denna utredning anser vi inte att det påverka uppsatsen negativt. Däremot tycker vi att det skulle vara mycket intressant att göra en uppföljning av modellen och utvärdera användningen av den i ett senare skede.

Syftet med ett fördjupningsarbete av detta slag är för oss som blivande systemvetare att se på användbarhet ur utvecklarens perspektiv. Vi ville ta del av ett verkligt projekt för att få just detta perspektiv och omsätta det vi lärt oss under utbildningen.

1.4 Metod

För att utreda frågan ska vi göra en djupstudie av litteratur inom de berörda ämnesområdena. Utifrån det ska vi konstruera en modell som visar hur krav relaterar sig till varandra och var olika krav kommer ifrån. Litteraturen kommer att användas för att diskutera och ifrågasätta aspekter av problemområdet och för att motivera vad som ska inkluderas och lyftas fram i modellen.

Vid genomgång av varje ämnesområde ger vi en kort presentation av området som introduktion för att kunna följa resonemanget varefter området kommer att presenteras utifrån fyra frågor. Frågorna har använts på de huvudområden och litteraturen därtill utgör grunden för vår genomgång. Utöver detta kommer vi att använda annan litteratur för att resonera, ifrågasätta och styrka olika resonemang dels i diskussionen av litteraturen och dels i våra egna slutsatser.

Frågorna har använts för att gripa sig an litteraturen och strukturera läsningen för att lättare fokusera på det som är intressant för frågeställningen. Genom att bryta ner vår ursprungliga frågeställning i mindre hanterbara frågor och jobba med dem under genomgången av litteraturen har vi varit mer strukturerade i vår sökning av material. Frågorna har dessutom gjort det möjligt att gå igenom litteraturen på, så

långt som möjligt, samma sätt. Vi anser detta ytterst viktigt vid en jämförelse av olika ämnesområden. Frågorna har slutligen använts som utgångspunkt vid jämförelse av litteraturen. Genom att jämföra de svar vi har kommit fram till har vi kunnat diskutera skillnader inom de olika teorierna och utifrån de skillnaderna utvidga diskussionen.

Frågorna vi har arbetat med är:

Hur definieras användbarhet?

Frågan syftar främst till att lägga fast innebörden av ordet och därmed skapa en utgångspunkt för vidare presentation och diskussion kring användbarhet och användbarhetskrav. Vi har här även sett närmare på hur man definierar krav eller mål och hur man mäter användbarhet.

Var kommer användbarhetskraven från?

Syftet med frågan är att få svar på vad som ligger till grund för krav. Vet man var kraven kommer ifrån kan man lättare jämföra användbarhetskrav med andra krav genom att se om de har gemensamt ursprung. Frågan behandlar även hur man tar fram användbarhetskrav eftersom det vittnar om synen på användbarhet, d.v.s. vad man betonar i analysen. Är det användarna, omgivningen eller uppgifterna som väger tyngst?

Hur används användbarhetskraven?

Genom att se hur användbarhetskrav används senare i utvecklingsprocessen och se vilken påverkan de har för den vidare utvecklingen står det klart hur viktiga de anses vara. Det är intressant att se hur användbarhet påverkar den fortsatta utvecklingen för att se hur analys och specificering av användbarhetskrav påverkar det slutliga resultat.

Hur relaterar användbarhetskraven till andra krav?

Frågan är egentligen den mest intressanta och den som ligger närmast vår ursprungliga frågeställning. Syftet med att ta med frågan var att se om det fanns något ämnesområde som täckte denna fråga och i så fall hur.

Eftersom all litteratur är skriven på engelska kan det uppstå problem vid översättning av vissa termer och begrepp. Engelska språket har ett större ordförråd och är nyansrikare än det svenska vilket kan få till följd att den svenska översättningen inte gör ursprungsformuleringen rättvisa. Så långt som möjligt har vi översatt alla termer till svenska men där vi tycker att betydelsen går förlorad har vi valt att även ge den engelska termen. Ett fall då vi alltid använder de engelska termerna är vid mått av användbarhet eftersom vi vid jämförelse vill ha den exakta betydelsen. Ett annat fall då engelska uttryck används är vid uttryck som kan anses vara vedertagna begrepp eller namn t.ex. Software Engineering och Usability Engineering.

1.5 Uppsatsens disposition

Kapitel två består av en litteraturgenomgång där de olika teorierna går igenom med ett objektiva synsätt. En jämförelse av de olika teorierna presenteras i början av kapitel tre där våra egna synpunkter får ett visst utrymme. Jämförelsen följs av en diskussion av vad jämförelsen gett och därefter presenteras modellen följt av en diskussion kring den. I diskussionen och presentationen av modellen lägger vi in mer av våra egna synpunkter, värderingar och slutsatser. Uppsatsen avslutas med en sammanfattning där slutsatser summeras.

2 Teori

Teorigenomgången kommer att fokusera på fyra angreppssätt på användbarhet. Angreppssätten är Software Engineering, Usability Engineering, ISO-standard om användbarhet och en doktorsavhandling om användbarhet.

Software Engineering kommer att presenteras utifrån professor Ian Sommervilles bok *Software Engineering*. Boken behandlar en hel systemutvecklingsprocess och riktar sig både till studenter och systemutvecklare. Software Engineering är intressant att ta upp då det behandlar en traditionell syn på systemutveckling.

Usability Engineering kommer att presenteras utifrån professor Deborah Mayhews bok *The Usability Engineering Lifecycle*. Boken presenterar olika tekniker inom Usability Engineering, som är en mer användarcentrerad syn på systemutveckling.

ISO-standard 9241-11 (Ergonomic requirements for office work with visual display terminals Part 11: Guidance on usability) sätter upp riktlinjer för specificering och utvärdering av användbarhet. ISO-standarder används inom många områden och många verksamheter följer dem i sitt arbete vilket gör det intressant att diskutera dem även i detta avseende.

Adding Usability - Methods for Modelling, User Interface Design and Evaluation är en doktorsavhandling inom människa-dator interaktion framlagd av Magnus Lif, Uppsala Universitet. Avhandlingen presenterar olika modeller inom användbarhet där vi ser närmre på UIM (User Interface Modelling).

2.1 Software engineering - Sommerville

Software Engineering kan översättas till mjukvaruutveckling och är ett traditionellt arbetssätt med metoder och verktyg för utveckling av mjukvarusystem. Det finns många olika metoder och angreppssätt inom Software Engineering och det sker en ständig utveckling. Trots detta implementeras det fortfarande system som inte uppfyller uppsatta krav, projekt som misslyckas och system som visar sig ej vara lönsamma. Detta visar på att Software Engineering fortfarande är en relativt ung disciplin som behöver utvecklas ytterligare. Utmaningen Software

Engineering står inför är att försäkra att mjukvaran verkligen svarar mot de krav som ställs och på så sätt möter de behov som finns (Sommerville 1998, s. 4ff)

Utvecklingsprocessen i software engineering består i grunden av fyra faser (Sommerville 1998, s. 7):

- analys och specificering av systemets funktionalitet och begränsningar
- utveckling och implementering av systemet
- testning att systemet uppfyller uppsatta krav
- vidareutveckling av systemet

Den del av software engineering som specificerar systemet och sätter krav, den första fasen, kallas requirements engineering. Denna fas kan delas upp i fyra steg. Första steget är att göra en utvärdering av ett systems effekter med avseende på lönsamhet och möjlighet att lyckas. En uppskattning om systemet kan tillgodose de behov som finns och om utvecklingen är möjlig inom de givna budgetramarna görs. Denna analys ska göras relativt snabbt och resultatet är ett beslut om huruvida man ska gå vidare med en analys eller lägga ner projektet. Det andra steget är en kravanalys d.v.s. att genom en analys av användare, uppgifter, omgivning och eventuellt befintliga system samla information för att komma fram till de krav som ska ställas på systemet. I det tredje steget definieras och formuleras kraven. Kravdokumentet ska skrivas i icke-tekniska termer för att kunden/beställaren ska kunna förstå innebörden av definierade krav. Det fjärde och sista steget resulterar i en kravspecifiering som är en mer detaljerad beskrivning av systemet och kan fungera som ett kontrakt mellan utvecklare och beställare (Sommerville 1998, s. 67f).

Det är viktigt att kraven som tas fram under specificeringen av ett system är giltiga, kompletta, konsekventa och realistiska. Ett krav är giltigt om det är sant och korrekt d.v.s. om det verkligen är ett krav på systemet. Kraven ska vara kompletta, d.v.s. alla krav på funktionalitet och begränsningar ska finnas med. Att kraven ska vara konsekventa betyder att de inte ska stå i konflikt med varandra. Kraven är realistiska om de är möjliga att genomföra tekniskt och kostnadsmässigt (Sommerville 1998, s. 71).

Kraven ska vara mätbara för att kunna testas. De ska även vara spårbara eftersom det ibland är nödvändigt att gå tillbaka till kravets ursprung för att kunna bedöma effekten av en förändring i kraven. Ytterligare en aspekt att beakta är hur fristående ett krav är, d.v.s. om man kan ändra kravet utan att det får stora effekter på andra krav (Sommerville 1998, s. 72).

Sommerville betonar vikten av att kraven ska vara så noggranna och väl satta som möjligt innan man går vidare i utvecklingen. Om det senare under utvecklingens gång upptäcks att krav hamnar i konflikt eller inte längre stämmer är det viktigt att uppdatera och modifiera kravdokumentet. Att kraven efter en tids arbete inte stämmer är naturligt eftersom utvecklaren under arbetets gång lär sig allt mer om domänen och användarens behov (Sommerville 1998, s. 71ff).

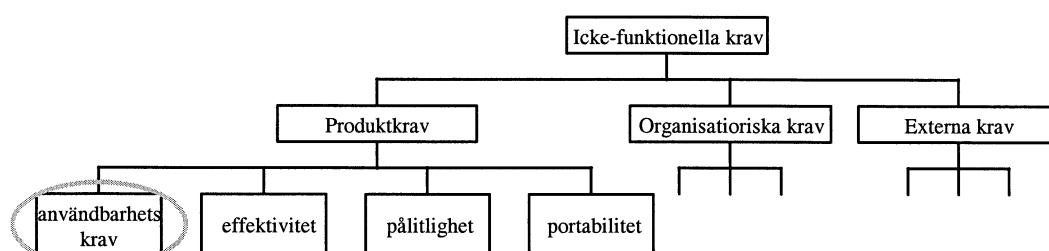
Att utveckla genom att skapa en prototyp och utvärdera och modifiera sina krav utifrån erfarenheterna av prototypen är ett sätt att försäkra sig om att man har ställt rätt krav på ett system (Sommerville 1998, s. 71).

2.1.1 Hur definieras användbarhet?

Sommerville definierar aldrig uttryckligen användbarhet. Däremot förslår han tänkbara mått att mäta användbarhet i vid utvärdering av gränssnittet (Sommerville 1998, s. 341):

- learnability - hur lätt systemet är att lära sig
- speed of operation - hur svarstiderna i systemet överensstämmer med tiden det tar att utföra uppgifterna
- robustness - hur mycket fel en användare kan göra
- recoverability - hur lätt det är att återställa systemet efter fel gjorda av användarna
- adaptability - hur mycket systemet kan anpassas till användarnas olika arbets-sätt

I framställningen av krav presenterar Sommerville två kategorier av krav: funktionella och icke-funktionella. Funktionella krav är en beskrivning av vad systemet ska göra och icke-funktionella krav är begränsningar på de funktionella kraven. Dessa krav uppstår av t.ex. användarnas behov, befintlig mjuk- och hårdvara, organisatoriska faktorer och lagar. Figur 1 visar att icke-funktionella krav delas upp i flera olika kategorier av krav där användbarhetskrav är en kategori (Sommerville 1998, s. 118, 130-131).



Figur 1. Icke-funktionella krav enligt Sommerville.

2.1.2 Var kommer användbarhetskrav från?

Innan krav kan tas fram måste en analys av domänen göras för att skaffa kunskap och inse användarnas krav. Detta gör man enligt Sommerville genom att interagera med systemets intressenter för att upptäcka deras behov. Vidare i analysprocessen klassificeras krav och prioriteringar görs om det skulle finnas konflikter

mellan kraven. Avslutningsvis kontrolleras att de specificerade kraven är i enlighet med vad intressenterna vill ha ut av systemet (Sommerville 1998, s. 80ff). Man får anta att användbarhetskrav kommer fram i denna process, men Sommerville förklarar inte närmre vad för slag krav som fås fram.

Vad som ingår i att skaffa kunskap om domänen och att interagera med intressenterna framgår inte heller. Han skriver: *"there are few if any methods which include steps to analyse and document the human interaction requirements."* (Sommerville 1998, s. 94). Han menar att sociala och organisatoriska faktorer är viktiga för kraven på systemet men eftersom de är svåra att identifiera finns det inte något systematiskt sätt att analysera dem.

2.1.3 *Hur används användbarhetskrav i Software engineering?*

Sommerville skiljer inte ut användbarhetskrav från andra icke-funktionella krav och därmed är det svårt att uttyda hur hänsyn tas till användbarhetskrav i Software Engineering. Han skriver: *"Unfortunately, it is rare for quantitative usability requirements to be included in the system requirements specification."* (Sommerville 1998, s. 341). Han menar alltså att det är ovanligt att mätbara användbarhetskrav finns med i kravspecifikationen och påpekar också att det gör det svårt att utvärdera användargränssnittet objektivt.

Senare i utvecklingen, närmare bestämt i utvärdering av gränssnittet, tas krav på användbarhet dock upp. Sommerville menar att det är viktigt att utvärdera användargränssnittet och ger exempel på olika tekniker att använda (Sommerville 1998, s. 341f). Utvärderingen sker då mot hans mått på användbarhet (learnability, speed of operation, robustness, recoverability och adaptability) och inte mot egna definierade användbarhetskrav.

2.1.4 *Hur relateras användbarhet till andra krav?*

De icke-funktionella kraven relaterar ofta till andra funktionella krav antingen genom att de står i konflikt eller interagerar. Sommerville menar att i princip ska de funktionella och de icke-funktionella kraven skiljas på i specificeringen. När ett krav formuleras ska det antingen vara ett funktionellt eller ett icke-funktionellt krav. Det framhålls dock att nackdelen med att separera kraven är att relationen mellan kraven kan bli svår att se. Eftersom de icke-funktionella kraven är av mer abstrakt karaktär är det viktigt att behålla relationerna för att komma ihåg de icke-funktionella kraven som en begränsning på de funktionella (Sommerville 1998, s. 133).

2.2 The Usability Engineering Lifecycle - Mayhew

Usability Engineering är en riktning där man vid design av system i förväg kvantitativt specificerar användbarhet för att sedan kunna testa att man uppnått den nivå man bestämt. Det är en iterativ process där design och utvärdering pågår tills man nått de uppsatta målen (Preece 1994, s. 650).

Mayhew presenterar *The Usability Engineering Lifecycle* som en sammanställning av de olika principer, metoder och tekniker som finns inom Usability Engineering. Sammanställningen gör hon för att hon tycker att det behövs en ansats att se på användbarhet under hela utvecklingsprocessen. *The Usability Engineering Lifecycle* är anpassad för att kunna användas genomgående som ett stöd i användningen av andra utvecklingsmodeller (Mayhew 1999, s. xf), t.ex. parallellt eller tillsammans med Software Engineering. Exakt hur denna överlappning eller integrering ska ske är enligt Mayhew inte helt klart (Mayhew 1999, s. 17). Däremot beskriver hon speciellt hur man kan använda *The Usability Engineering Lifecycle* tillsammans med Object-Oriented Software Engineering(OOSE), d.v.s. objekt-orienterad systemutveckling. Hon påpekar att OOSE har ett användarcentrerat utvecklingsätt men saknar många av de tekniker och mål som finns i Usability Engineering. Man kan använda de tekniker som Mayhew beskriver som en utvidgning av OOSE och därmed förbättra användbarheten (Mayhew 1999, s. 27ff).

Livscykeln inleds med en analys innefattande observation och studier av användare, uppgift, plattform och generella designprinciper som är möjliga att tillämpa i det aktuella fallet. Denna analys är grunden för att sätta de användbarhetsmål som sedan ska driva utvecklingen av användargränssnittet. Användbarhetsmålen måste prioriteras och dokumenteras innan man går in i designfasen. Mål på användbarhet specificeras redan från början. För att kunna jämföra resultat från prototyp eller designförslag ska de uppsatta målen vara väldefinierade och mätbara.

Designfasen är uppdelad i tre nivåer. Den första organiserar funktionalitet och utarbetar förslag på gränssnitt i konceptuella modeller d.v.s. på hög nivå. Den andra sätter standarder för detaljerad gränssnittsdesign. Resultatet från de två designnivåerna dokumenteras och ligger sedan till grund för den tredje designnivån då gränssnittet färdigställs. Gränssnittet testas och utvärderas av användare och vidareutvecklas tills användbarhetsmålen har uppfyllts. Design, testning och utveckling sker med andra ord iterativt. Gränssnittet driver sedan utvecklingen av hela systemet. Systemet ska tjäna användarna och eftersom de bara kommer i kontakt med användargränssnittet är det just användargränssnittet som är systemet för användarna (Mayhew 1999, s. 5ff, 17).

Mayhew skriver inte om användbarhetskrav utan använder termen användbarhetsmål. Krav kan ses som mål som måste uppnås (Newman & Lamming 1998, s. 143) och därför kommer vi att använda Mayhews val av termen användbarhetsmål. Dock är det en skillnad mellan rena användbarhetskrav som är krav på användbarhet och användbarhetsmål som är mål med användandet, d.v.s. även funktionalitet och funktionella krav.

2.2.1 Hur definieras användbarhet?

Ett gränssnitt har en viss nivå av användbarhet. Användbarhet är enligt Mayhew en mätbar karakteristika i ett systems gränssnitt. En förklaring på användbarhet är hur lätt ett system är att lära sig och hur lätt det är att använda (Mayhew 1999, s. 1).

Användbarhet specificeras genom användbarhetsmål som kan vara kvalitativa eller kvantitativa. Kvalitativa mål är generella och icke mätbara mål som tjänar som riktlinjer för designen varför de är speciellt användbara i början av designfasen. Exempel på ett kvalitativt mål är: "det ska gå snabbt för en användare att lära sig systemet". Kvantitativa mål är något man kan mäta och delas upp i hur lätt det är att lära sig och hur lätt det är att använda ett system. Hur lätt ett system är att använda har att göra med hur effektivt, flexibelt och kraftfullt det är för en användare som lärt sig grunderna. Kvantitativa mål är mätbara och kan därmed kontrolleras och kan härledas ur de kvalitativa målen (Mayhew 1999, s. 126ff). Ovan nämnda exempel på kvalitativa mål - ett system ska gå snabbt att lära sig - kan bestå bl.a. av det kvantitativa målet: "en användare ska kunna utföra uppgiften med högst två fel efter en timmas träning".

De kvantitativa målen kan mätas enligt (Mayhew 1999, s. 128f):

- performance - hur en användare presterar, hur lång tid en uppgift tar att göra, hur många fel som görs och hur mycket hjälp som behövs
- preference - vad en användare föredrar vid val mellan två olika designförslag eller mellan två olika tillvägagångssätt
- satisfaction - en användares allmänna tillfredsställelse, d.v.s. en subjektiv uppfattning av systemet

Det är inte alltid som dessa tre mål korrelerar. T.ex. behöver inte det designförslag där användaren presterar bäst vara det förslag som han föredrar om han får välja. Att uppfylla ett prestationsmål innebär därför inte att även den allmänna tillfredsställelsen automatiskt uppfyllts. De tre olika målen ska ses som självständiga mål som alla ska tas hänsyn till (Mayhew 1999, s.129).

2.2.2 Var kommer användbarhetskrav från?

Som tidigare nämnts är första steget i The Usability Engineering Lifecycle en analys då man samlar in data om bl.a. tilltänkta användare, användarnas arbetsuppgifter och generella designprinciper (Mayhew 1999, s. 122). Insamling av data sker genom intervjuer och frågeformulär. Den insamlade datan bör vara förstahandsinformation vilket kommer naturligt då Mayhew förespråkar användarmedverkan (Mayhew 1999, s. 39).

Användare.

Syftet med att samla in information om användarna är att kategorisera dem och få en generell bild av varje kategori för att se vad de har för krav och vad som passar dem. En designlösning kan optimera en kategori användares prestation medan

samma lösning kan försämma en annan kategori användares prestation. Ett exempel på detta är valet att prioritera antingen att ett system är lätt att lära sig eller att ett system är lätt att använda. Ett system som är lätt att lära sig har ett mer intuitivt gränssnitt och användaren kan prestera bra redan första gången han använder det. Visar sig en användarkategori ha lite datorvana kan detta vara en bra prioritering. Om en användarkategori å andra sidan visar sig vara skickliga datoranvändare blir ett alltför enkelt gränssnitt frustrerande och begränsande att arbeta med (Mayhew 1999, s. 35).

Aspekter och faktorer hos användare att analysera (Mayhew 1999, s. 36):

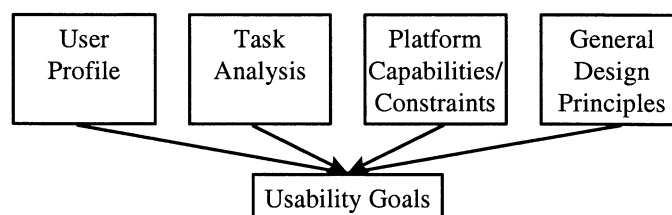
- psykologiska egenskaper - inställning till arbetet, arbetsuppgiften och datoranvändning samt inställning till förändring i arbetssituationen
- kunskap och erfarenhet - inom t.ex. arbetsområde och datoranvändning
- arbetsuppgifter - karaktär av uppgift d.v.s. vad som utförs och hur det utförs
- fysiska egenskaper - t.ex. kroppsbyggnad och eventuella handikapp

Arbetsuppgifter.

Nästa steg i processen är att analysera arbetsuppgifterna. Beroende på hur mycket av systemets funktionalitet som redan har specificerats kan denna fas se olika ut. Mayhew utgår från att generell funktionalitet redan har specificerats och beskriver inte hur en större process, med t.ex. business reengineering, går till (Mayhew 1999, s. 67). Datan som ligger till grund för analysen av arbetsuppgifter bör samlas in genom intervjuer och observationer av användarna i deras arbetsmiljö (Mayhew 1999, s. 69). I en traditionell systemutvecklingsprocess samlar man också in data om arbetsutförandet, men enligt Mayhew krävs ett mer användarcentrerat perspektiv. Traditionell systemutveckling kan på grund av avsaknad av ett användarcentrerat perspektiv enligt Mayhew misslyckas i att organisera och presentera data och funktionalitet på ett sätt som optimerar arbetsutförandet för användaren. Analys av arbetsuppgifter i The Usability Engineering Lifecycle är ett komplement till traditionell systemutveckling. Syftet med analysen är att ge en beskrivning av hur arbetet utförs idag och fokuserar på hur användarna upplever sin nuvarande arbetssituation d.v.s. både arbetsuppgifter och arbetsmiljö. Beskrivningen ska ge designern en ökad förståelse för användarnas arbete (Mayhew 1999, s. 67ff).

Generella designprinciper.

Det finns redan mycket kunskap om generella designprinciper och det är viktigt att beakta denna kunskap när man formulerar användbarhetsmål (Mayhew 1999, s. 160).



Figur 2. Analys enligt Mayhew.

Analysen av användare, användarnas arbetsuppgifter och de generella designprinciperna är alltså grunden till alla användbarhetsmål. Som figur 2 visar ska även plattformsbegränsningar och möjligheter ingå i analysen. Användbarhetsmål är målen med användningen av systemet och inkluderar därmed även funktionalitet. Mayhew betonar vikten av analysfasen då man behöver kunskapen om domänen för att kunna möta användarnas krav på systemet.

2.2.3 *Hur används användbarhetskrav i Usability Engineering Lifecycle?*

Användbarhetsmål tjänar enligt Mayhew två syften:

- som riktlinjer för beslut om gränssnittsdesignen. Inga beslut som tas angående designen ska vara slumpartade utan varje beslut ska vara förankrat i enlighet med användbarhetsmålen. Gränssnittsdesignen är av yttersta vikt och ska driva hela utvecklingsprocessen (Mayhew 1999, s. 17, s. 126).
- som kriterier vid utvärdering. Metoden förespråkar en iterativ utveckling där resultatet jämförs med de mål man har satt upp. Utveckling och utvärdering fortsätter tills målen har uppfyllts. Användbarhetsmålen kan också användas för att utvärdera olika designidéer redan under utvecklingsarbetet (Mayhew 1999, s. 123ff).

2.2.4 *Hur relateras användbarhetskrav till andra krav?*

Enligt Mayhew bör de kvantitativa målen formuleras utifrån de mer generella kvalitativa målen. Man skulle således kunna säga att de kvantitativa användbarhetsmålen kommer från de kvalitativa (Mayhew 1999, s. 131).

Mayhews framställning saknar resonemang kring hur de funktionella kraven relaterar till användbarhetskraven. Mayhew tar inte upp funktionella krav utan skriver bara om krav som användbarhetsmål. När man integrerar Usability Engineering med Software Engineering blir relationen mellan funktionella krav och användbarhetskrav mycket viktig. Mayhew menar att relationen inte är tydlig och beroende på domän- och projektspecifika faktorer (Mayhew 1999, s. 17).

2.3 **ISO 9241-11**

ISO (the International Organization for Standardization) är en internationell organisation bestående av nationella standardinstitut från ca 130 länder. Sverige representeras av SIS, Swedish Standards Institute. ISO arbetar med att utveckla internationella standarder för att genom gemensamma definitioner och referenser

underlätta globalt samarbete. En standard är enligt ISO *"documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose"* (ISO 1999a).

Arbetet med att ta fram förslag till en standard görs i tekniska kommittéer där varje institution som har intresse i ett ämne har rätt att delta i kommittén. I kommittéerna sitter representanter från industrin, forskningsinstitutioner, myndigheter och konsumentorganisationer. Innan ett förslag kan godkännas och publiceras som internationell standard ska 75 % av de nationella standardinstitutionerna rösta för förslaget. ISO bedriver samarbete med andra internationella organisationer; t.ex. International Electrotechnical Commission (IEC) och World Trade Organization (WTO) (ISO 1999a).

ISO 9241-11 (Ergonomic requirements for office work with visual display terminals Part 11: Guidance on usability) syftar till att ge vägledning i hur man specificerar och utvärderar användbarhet. Standarden definierar vad användbarhet är och ger riktlinjer för hur man kan gå till väga för att ta fram den bakgrundsinformation som behövs för att kunna specificera användbarhet (ISO 1998, s. 1). Någon utarbetad metod för hur man går till väga vid specificering presenteras dock inte, utan endast exempel på hur man utifrån angivna riktlinjer kan mäta användbarhet eller skriva en kravspecifikation. Det finns däremot en annan standard, ISO 13407, som i samband med användarcentrerad design presenterar ett arbetssätt för att specificera krav från användare och organisation och därefter utforma design (ISO 1999b, s. 6). ISO 9241-11 är främst tänkt att användas för arbete vid skärmar i en kontorsmiljö, men kan även vara tillämplig i andra miljöer där en användare interagerar med en produkt (ISO 1998, s. 1).

2.3.1 Hur definieras användbarhet?

ISO 9241-11 definierar användbarhet som *"Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."* (ISO 1998, s. 2) Tre mått för att mäta användbarhet är således (ISO 1998, s. 2):

- effectiveness - hur noggrant och hur fullständigt ett mål uppnås
- efficiency - hur effektivt mål uppnås, dvs hur mycket resurser som krävs för att uppnå ett mål
- satisfaction - hur nöjda användarna är med systemet och deras inställning till systemet

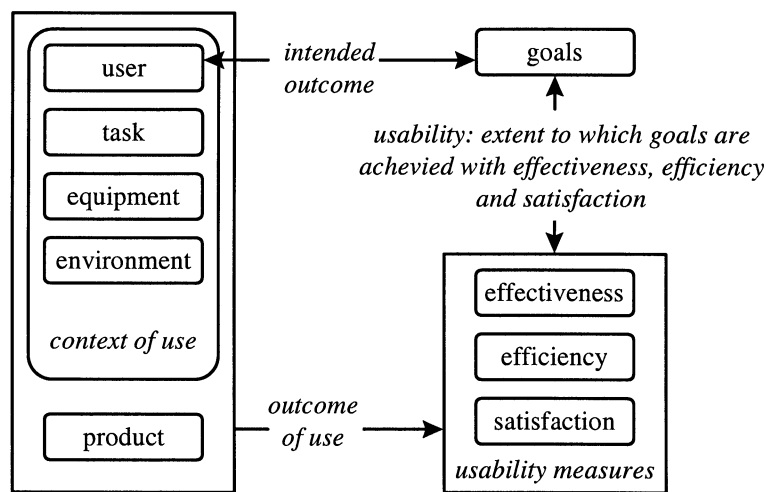
Olika aspekter av användbarhet är viktiga beroende på vilket sammanhang systemet används i och beroende på vad för slags mål man mäter. Det finns därför inga generella regler för vilka variabler som ska användas eller hur de ska bedömas och kombineras (ISO 1998, s. 5).

2.3.2 Var kommer användbarhetskrav från?

I standarden behandlas inte användbarhetskrav utan mål. Med mål menas de förväntningar som användare har med användningen av ett system och kan vara av mer övergripande karaktär eller indelade i mer specificerade delmål. Eftersom målen beskriver användningen av systemet innefattas både funktionalitet och användbarhet i beskrivningen av dem.

För att kunna specificera mål måste man ha information om kontexten, d.v.s. en beskrivning av användare, uppgifter, utrustning och miljö, för användningen. Beskrivningen av användarna bör innefatta attribut såsom kunskap, skicklighet, erfarenhet och fysiska attribut. Uppgifter ska beskrivas utifrån de aspekter som kan påverka användbarhet, t.ex. hur ofta en uppgift utförs och hur lång tid den tar att utföra. Beskrivning av miljön innefattar både den fysiska miljön, t.ex. ljud, ljus och temperatur på arbetsplatsen, samt den sociala miljön, t.ex. organisation, attityder och kultur (ISO 1998, s. 4).

ISO 9241-11 presenterar ett ramverk för att beskriva komponenterna i användbarhet och funktionalitet och hur de förhåller sig till varandra (se figur 3).



Figur 3. Ramverk för användbarhet.
Från ISO 9241-11, 1998, Geneva: ISO. Copyright ISO och SIS Förlag 1998.
Reproducerad med tillstånd.

Ramverket består av tre komponenter.

- Målet med användningen av systemet.
- Produkten placerad i kontexten, d.v.s. systemet i samspel med användare, uppgifter, utrustning och omgivning.

- Måtten effectiveness, efficiency och satisfaction formulerade så att de är verifierbara.

2.3.3 *Hur används användbarhetskrav?*

Ramverket kan användas för att specificera användbarhet samt för att utvärdera eller mäta användbarheten i ett befintligt system. Genom att utgå från ramverket ger systemet i sin kontext, d.v.s. användning av systemet, ett resultat som kan mätas med användbarhetsmåten. Resultatet jämförs därefter med målen och i den utsträckning som målen uppfylls är graden av användbarhet. Måtten på användbarhet används således för att kontrollera att målen med användningen av systemet har uppfyllts.

2.3.4 *Hur relateras användbarhetskrav till andra krav?*

En specifikation av användbarhetskrav innehåller de uppsatta målen, beskrivning av kontexten systemet ska användas i samt de kriterier för användbarhet som har specificerats för effektivt arbete och tillfredsställelse. Vid beskrivning av uppgifterna i kontexten ska alla aktiviteter som krävs för att genomföra en uppgift inte enbart beskrivas som funktioner som kan utföras av ett system utan också relateras till de uppsatta målen (ISO 1998, s. 4). Vid bedömning av användbarhet undersöker man i vilken utsträckning de uppsatta målen har blivit uppfyllda vid användning av systemet, varför man kan säga att målen är ett steg mot användbarhet. Om man beskriver aktiviteter och uppgifter i relation till målen har man gjort en svag koppling mellan vad systemet ska göra i form av funktionalitet och vilka möjligheter och begränsningar kontexten för systemet ger. Tydligare koppling mellan olika krav än så får man dock inte eftersom inga riktlinjer för hur man sätter upp kriterier för användbarhet ges av ISO 9241-11.

2.4 **Adding Usability - Magnus Lif**

Magnus Lif från Uppsala Universitet behandlar i sin doktorsavhandling hur man integrerar användbarhet i en systemutvecklingsprocess. Lif har tillsammans med andra forskare (Gulliksen, Lind, Nygren, Sandblad) tagit fram modellen AIU och själv tagit fram modellen UIM. AIU (Analysis of Information Utilization) är en modell för analys av användarnas behov och användning av information. UIM (User Interface Modelling) är en modell som sätter användargränssnittet och designern av användargränssnittet i fokus. Målet med modellen är att specificera mål som användarna vill uppnå i sitt arbete med informationssystemet. Dessa mål ska ligga till grund för alla beslut om designen.

Idag används ofta objektorienterade utvecklingsmetoder där man drar nytta av objektorienterad modellering. UML (Unified Modeling Language) och är ett grafiskt språk för att specificera, visualisera, dokumentera och konstruera i en systemutvecklingsprocess (Kruchten 2000, s. 28). I objektorienterad systemutveckling modellerar man för att få en bättre förståelse av det man arbetar med. Ett sätt att modellera är i så kallade use case. Use case är en sekvens av händelser i ett system sett från användarens perspektiv.

Objektorienterade metoder är bra för att utveckla många delar av systemet men garanterar inte ett användbart gränssnitt. När designern har dåligt stöd från analysen är risken att allt för många gränssnittsbeslut är upp till designerns tycke och smak. UIM är utvecklad för att fokusera på en avgränsad bit i systemutvecklingen; att samla användarnas krav och använda dem för att utforma användargränssnittet (Lif 1998, s. 84). UIM är tänkt att används tillsammans med use case som beskrivs mer i detalj nedan.

2.4.1 *Hur definieras användbarhet?*

Lif hänvisar till Jakob Nielsens och ISO 9241-11:s definitioner av användbarhet.

Nielsen slår fast att användarvänlig inte är en bra term att använda i det här sammanhanget eftersom användare inte vill att datorer ska vara vänliga. Istället vill han använda termen nyttig (usefulness) för att benämna huruvida ett system kan användas för att uppnå uppsatta mål. Termen kan vidare delas in i ändamålsenlig/funktionell (utility) och användbarhet (usability). Ändamålsenlig/funktionell är en bedömning av huruvida systemet innehåller den funktionalitet som behövs och användbarhet är en bedömning av hur väl en användare kan utnyttja denna funktionalitet (Nielsen 1993, s. 24ff).

ISO 9241-11:s definition av användbarhet beskrivs ovan (2.3.1)

Vidare diskuterar Lif användbarhet i gränssnitt. Han menar att vid design av gränssnitt ska man sträva efter att minimera den kognitiva belastningen. Arbetet med att hantera gränssnittet och systemet är inte en användares huvudsyssla och ska därför inte belasta den kognitiva förmågan. Genom att gränssnittet presenterar information på ett sätt som kräver minimal kognitiv kapacitet att hantera lämnar man så mycket uppmärksamhet som möjligt åt själva arbetet att utföra. Lif menar också att om användaren kan hantera gränssnittet mer automatiskt när han utför sina arbetsuppgifter blir systemet transparent för användaren. Detta är önskvärt eftersom att utföra arbetsuppgifter och att hantera gränssnittet kan ses som två uppgifter som konkurrerar om användarens uppmärksamhet. Ytterligare en aspekt av användbarhet är att informationen ska kunna uppfattas likväl som att läsas eftersom skickliga arbetsutövare avkodar information genom att se mönster (Lif 1998, s. 49ff).

2.4.2 Var kommer användbarhetskrav från?

Lif menar att användarkrav kommer från analys av verksamheten vilket är till hjälp vid utveckling av systemet. Denna analys ger inte tillräckligt mycket av den information som behövs för att utforma design av gränssnittet varför Lif föreslår en ny metod för detta ändamål. Syftet med Lifs metod, UIM, är att minska glappet mellan analys och design och öka möjligheterna att skapa användbara gränssnitt (Lif 1998, s. 22).

UIM är alltså en metod för insamling av användarkrav som ska användas för att utforma design av gränssnitt. UIM bygger på användning av use case modell. För att bygga en use case modell använder man termerna use case och actors. En actor är någon eller något som interagerar med ett system. En actor kan alltså vara en kategori användare men också t.ex. ett annat system. En actor är en kategori av användare d.v.s. en generaliserad grupp och benämns som klass i objektorientering. En specifik användare är en instans av en klass. Ett use case är interaktion mellan systemet och en actor genom att en actor utför något av systemets funktionalitet. Ett use case är en klass och en instans av ett use case är en sekvens av relaterade transaktioner. I ett transaktionssystem är kassörskan, administratör, lagerarbetare exempel på actors och ta betalt för vara, registrera levererade varor och ge användare en rättighet exempel på use case. De verkliga användarna ska ta del i modelleringen då det är de som har kunskapen om use case. Genom att beskriva verksamheten på detta sätt får man en bra uppfattning om hur systemets olika objekt kommunicerar i systemet. Denna modell räcker dock inte för att kunna skapa ett användbart gränssnitt (Lif 1998, s. 85).

UIM innebär att man utöver use case model gör ytterligare tre modeller (Lif 1998, s. 90ff):

Goal model – användarnas mål med användandet. I UIM anges dessa mål på hög nivå, d.v.s. inte allt för detaljerat. Målen är en beskrivning från användarnas perspektiv. Varje mål får en prioritet.

Actors model – varje klass av användare beskrivs i detalj. Ibland kan det vara nödvändigt att dela upp en klass ytterligare i t.ex. nybörjare och expert.

Work situation model – i en arbetssituation kommunicerar en actor med en eller flera use case. En arbetssituation är en samling use case som naturligt hör ihop och som utförs av en instans av en actor. Omfattningen av en arbetssituation sträcker sig till avslutandet av alla inkluderade instanser av use case.

Frågor att beakta för att beskriva de olika arbetssituationerna kan vara:

- Vilka är de huvudsakliga uppgifterna för de olika actors?
- Vilka use cases är logiskt kopplade till varandra?
- Vilka use cases är kopplade tidsmässigt?
- Vilka use cases behöver tillgång till samma information?

2.4.3 *Hur används användbarhetskrav i UIM?*

Utifrån de modeller som gjorts kan man dra slutsatser om vilka krav som ska sättas på användargränssnittet. För att en användare ska kunna utföra en uppgift är det viktigt att all information och alla funktioner finns tillgängliga på skärmen samtidigt. Därför är det viktigt att ta reda på vilken information och vilka funktioner en actor behöver i en viss arbetssituation.

Designbeslut kan tas med hänsyn till hur många objekt (samling av information) som användaren vill se samtidigt. Behövs många objekt måste man se till den prioritering som gjorts och göra de viktiga objekten tydligare och framhäva dem visuellt t.ex. med hjälp av färg. På samma sätt tar man beslut om hur funktionalitet i form av olika operationer ska framhävas. De som prioriterats högst och används mest framhävs.

Fördelen med att se på arbetssituationer istället för enskilda uppgifter är att man ser på arbetsuppgifterna i sitt sammanhang vilket gör att kontexten och syftet med uppgifterna framhävs (Lif 1998, s. 47). Genom att se på arbetssituationer minskar risken för ett fragmenterat gränssnitt där varje uppgift har en skärmsida (Lif 1998, s. 84). Istället grupperas uppgifterna efter arbetssituationer även i gränssnittet.

2.4.4 *Hur relateras användbarhetskrav till andra krav?*

Då Lif hänvisar till Nielsens definition av användbarhet som består av ändamålsenlig/funktionell (utitlity) och användbarhet (usability) (Nielsen 1993, s. 24ff) ser man en tydlig relation mellan funktionalitet och användbarhet.

Lif skriver dock inte om krav på användbarhet. Alla krav ryms i hans begrepp arbetssituation och genom att analysera arbetssituationen får man fram alla krav som behövs. Hur de förhåller sig till varandra är dock oklart.

3 Diskussion och resultat

3.1 Resultat av litteraturjämförelse

För jämförelse av litteraturen kommer frågorna från litteraturgenomgången att ligga till grund. Svaren på frågorna kommer att jämföras och vara utgångspunkt för en vidare diskussion.

3.1.1 *Hur definieras användbarhet?*

Sommerville definierar aldrig användbarhet - ett faktum som visar hur lite vikt som fästs vid det. Enligt Sommerville sätter man inte från början upp kriterier på användbarhet som sedan ska testas om de har uppnåtts vilket innebär att man inte behöver definiera vad användbarhet är. Man har istället bara en mer generell uppfattning om användbarhet i ett system att stämma av mot. Följden blir att man i den slutliga produkten inte kan peka på vad som är användbarhet eller vad som har gjorts med hänsyn tagen till användare och organisation eftersom det inte finns någon klar definition att gå tillbaka till.

Mayhew menar att användbarhet är något mätbart i ett gränssnitt som kan förklaras som hur lätt ett system är att lära sig eller hur lätt det är att använda. Till skillnad från Sommerville menar hon att användbarhetsmål ska driva designen av gränssnittet redan från början eftersom gränssnittet är systemet för användaren. Hon lägger med andra ord stor vikt vid användbarhet genom hela utvecklingsprocessen.

ISO 9241-11 behandlar i likhet med Mayhew användbarhet som mål. Användbarhet är i vilken utsträckning användare uppnår de mål de har med användningen av ett system. Det vill säga att det är något som man sätter upp och därefter arbetar för att uppnå. Skillnaden mellan användbarhet och användbarhetsmål är att med mål avses hela användningen. Användbarhetsmål blir krav på funktionalitet där både krav på användbarhet och funktionella krav ingår. Samma princip gäller hos Lif som hänvisar till både ISO:s och Nielsens definition. Nielsen definierar inte användbarhet som mål, men gör en uppdelning i ändamålsenlighet och användbarhet vilket kan liknas vid ISO:s och Mayhews mål i och med att de också täcker

in både funktionalitet och användbarhet i definitionen av mål.

Sommerville å andra sidan delar in krav i funktionella och icke-funktionella där användbarhet är icke-funktionella krav. Icke-funktionella krav ses som begränsning på de funktionella kraven i motsats till t.ex. Mayhew som ser användbarhet som mål. Begränsningar är något man ska hålla sig inom medan mål är något man strävar efter att uppnå.

Mått på användbarhet.

Mått vidgar definitionen av användbarhet. I princip är det samma aspekter av användbarhet som mäts i de olika ämnesområdena, men de lägger olika betoning på de subjektiva och de objektiva måtten. Ett objektiva mått mäter hur användare jobbar med systemet, t.ex. effektivitet, medan ett subjektivt mått mäter användarens inställning till användning av systemet.

Enligt ISO 9241-11 ska man definiera kriterier för vad användbarhet är i termer av effectiveness, efficiency och satisfaction för varje mål användaren har med systemet (se figur 4). Uppnår man dessa kriterier har man uppnått användbarhet. ISO har således två mer objektiva mått och ett subjektivt.

Mayhew delar, till att börja med, upp användbarhetsmål i kvalitativa och kvantitativa. Hon menar att kvalitativa övergripande mål inte går att kontrollera om de har realiserats utan det krävs kvantitativa, d.v.s. mätbara, mål också. Vidare kan kvantitativa mål kan mätas i performance, preference och satisfaction (se figur 4). Performance är ett objektiva mått medan preference och satisfaction är två subjektiva mått.

Sommerville mäter användbarhet i learnability, speed of operation, robustness, recoverability och adaptability (se figur 4). Genom att använda dessa mått specificerar han, mer än vad ISO och Mayhew gör, vad användbarhet är. ISO och Mayhews mått sätts utifrån de mål som har specificerats vilket innebär att det specifika fallet bestämmer exakt vad som ska mätas. Tanken att användbarhet ska specificeras i och är beroende av kontexten finns även hos Bevan (Bevan 1997, s. 2). Sommervilles mått å andra sidan innebär att användbarhet är samma sak i alla situationer och han har därmed bestämt vad som ska mätas. Han skiljer sig dessutom i det att han inte tar upp någon av de subjektiva aspekterna; däremot beaktar han hur tolerant systemet är för användaren genom att mäta robustness och recoverability.

Sommerville	Mayhew	ISO 9241-11
learnability speed of operation robustness recoverability adaptability	performance preference satisfaction	effectiveness efficiency satisfaction

Figur 4. Mått på användbarhet enligt Sommerville, Mayhew och ISO 9241-11.

3.1.2 Var kommer användbarhetskrav från?

Inom alla ämnesområden presenteras någon form av analys som ligger till grund för användbarhetskraven. Analyserna ser väldigt olika ut och fördelarna med de olika metoder som används poängteras av de olika förespråkarna. Gemensamt för de olika analyserna är att de presenterar vilken information som behövs för att formulera användbarhetskrav, varifrån informationen kommer, hur användbarhetskrav ska vara formulerade och definierade, men inte hur informationen används för att specificera krav. Hur man utifrån analysen formulerar användbarhetskrav är svårt att uttyda. En mängd insamlad information är inte samma sak som krav. Triviala exempel ges, t.ex. om systemet endast ska användas lite ska man prioritera att utveckla ett system som är lätt att lära, men inga exempel på hur man gör med mer komplexa situationer ges. En bra analys ger information om olika aspekter men det saknas vägledning i hur man ska kombinera, prioritera och omsätta all information. Både Mayhew och Sommerville hävdar att man ska göra prioriteringar mellan krav som står i konflikt med varandra, men krav som står i konflikt med varandra är redan formulerade krav. Prioriteringen ger därför endast vilka krav som ska implementeras inte hur krav ska formuleras.

Sommervilles analys går ut på att analysera domänen för att upptäcka intressenternas behov, klassificera kraven och göra eventuella prioritering samt avslutningsvis kontrollera att kraven är vad intressenterna vill ha. Intressenter är inte bara användare utan kan vara vem som helst som på något sätt har med systemet att göra eller har kunskap om domänen. Det är upp till utvecklaren att bestämma vilka intressenterna är, vilket kan tyckas ge utvecklaren en stor frihet till skillnad från de andra teorierna. I och med att det är intressenternas behov som ska kartläggas betonar Sommerville inte användarna som förstahandskälla vilket t.ex. Mayhews analysmetod, som kräver ett användarcentrerat perspektiv, gör. Han anser att det finns få, om än någon, metod som behandlar "human interaction requirements". Mayhews analys å andra sidan går ut på att analysera användare, arbetsuppgifter och omgivning som sedan ligger till grund för att formulera användbarhetsmål. Genom att analysera dessa aspekter beaktar analysen de krav som kommer från användarna.

Precis som Mayhew har Lif ett användarcentrerat perspektiv, vilket i hans fall betyder att analysen resulterar i användarkrav snarare än användbarhetskrav. Analysen i Lifs fall kommer från arbete med use case och hans egen metod UIM. Hans analysmetod betonar att man inte ska analysera var sak för sig utan se det i större mer meningsfulla arbetssituationer för att kunna sätta rätt krav.

Enligt ISO ska man specificera vad användbarhet är men standarden behandlar inte användbarhetskrav i någon större utsträckning. Användbarhet kommer från en beskrivning av mål och kontext och utifrån det formulerar man vad användbarhet. I ett exempel på hur en kravspecifikation kan skrivas enligt ISO 9241-11 (ISO 1998, Annex C) ger man exempel på uppgifter som ska utföras och vilka kriterier på effektivitet och tillfredsställelse som har specificerats för dessa uppgifter. Hur man har kommit fram till dessa kriterier nämns dock inte i denna standard utan behandlas i ISO 13407.

3.1.3 Hur används användbarhetskrav?

När användbarhetskraven i en specifikation väl är formulerad gäller det att använda dem i det fortsatta arbetet. Svårigheten med att applicera användbarhet är att ta sig från krav till design, d.v.s. implementera kraven. Detta är ingen enkel uppgift och det ligger i designens natur att en del har med stil och känsla att göra vilket inte kan specificeras i krav. Då programmerare inte alltid har denna kunskap om design vore det en fördel om man utifrån användbarhetskraven kunde sätta krav på utformning av användargränssnittet.

Mayhew sätter upp användbarhetsmål från början och formulerar dem tydligt och mätbart. Dessa mål är grunden för beslut om gränssnittet och gränssnittet är sedan drivande för hela utvecklingen av systemet. Detta förfaringssätt ger användbarhetsmålen en mycket central roll och skiljer sig stort från t.ex. Sommervilles sätt att använda användbarhetskrav. Mayhew ger dock inget direkt exempel på hur man utifrån användbarhetsmålen tillfärdar beslut om gränssnittsdesignen.

Lif går längre än Mayhew i att tala om hur man ska använda den information som analysen ger då han talar om att ordna den i arbetssituationer, ta reda på vilken information som behövs och sedan göra prioriteringar. Lifs utgångspunkt för design av gränssnittet är således de informationsobjekt som behövs i en arbetssituation och utifrån dem och de prioriteringar som gjorts av dem kan man ta beslut om gränssnittet. Till exempel kan beslut om vilken information (informationsobjektets attribut) som ska visas i en arbetssituation tas genom att se på den prioritering som gjorts. Den högst prioriterade informationen och därmed den viktigaste visas i gränssnittet medan mindre viktig information läggs bakom och kan nås från t.ex. en meny. Lif visar på detta sätt hur man kommer från analysen och kraven till beslut angående design.

Kriterierna i ISO 9241-11 för användbarhet kan användas för att kontrollera att man har uppfyllt de mål man har med systemet. Eftersom måtten blir mått på användbarhet först i relation med målen och måste specificeras på egen hand i den aktuella situationen är det svårt att verifiera att kriterierna verkligen är mått på användbarhet. Man kan kontrollera att det t.ex. inte tar mer än 20 timmar att lära sig ett system och att man därmed har uppfyllt sina krav på användbarhet. Däremot kan man inte verifiera att det är just 20 timmar som är det verkliga måttet på användbarhet. Det kanske är 5 timmar eller 40 timmar som är det riktiga måttet. Det måste finnas en spårbarhet eller relation till andra krav för att kunna se varför kraven är satta som de är och för att kunna se att användbarhetskraven verkligen förbättrar användbarheten om de realiserar.

Sommerville säger till skillnad från Mayhew att det är ovanligt att mätbara användbarhetskrav finns med i kravspecifikationen. Användbarhet kommer in först i utvärdering av gränssnittet då de definierade måtten används. Då kraven på användbarhet inte är uppsatta i förväg är det svårt att se hur utvärderingen är kopplad till analysen varifrån användbarheten borde komma. Sommerville mäter användbarhet i gränssnittet efter de fem bestämda måtten och på så sätt har han, som påpekats tidigare, redan bestämt vad användbarhet är i alla system. Att mäta speed of operation i ett gränssnitt är intressant i vissa fall men inte nödvändigtvis i alla. Om t.ex. speed of operation inte tillför systemet något i form av användbarhet

tjänar det inget till att använda det som mått på användbarhet. Effectiveness som används inom de andra ämnesområdena är mer generellt och kan innefatta andra aspekter av effektivitet än just speed of operation.

3.1.4 *Hur relaterar användbarhetskraven till andra krav?*

Kopplingen mellan användbarhetskrav och andra krav beror på hur man delar upp och kategoriserar de olika krav som finns. Sommerville delar traditionsenligt upp krav i icke-funktionella och funktionella. Med en sådan uppdelning, där man skiljer på funktionalitet och användbarhet, är det viktigt att kunna se kopplingen mellan dem.

För de metoder som använder sig av användbarhetsmål, Mayhew och ISO 9241-11, ryms både funktionalitet och användbarhet i målen. Relationen mellan olika sorters krav är då tydligare, men det är svårt att utläsa vad som är vad i målen. I och med att användbarhetsmål inte kan implementeras behöver man bryta ner dem i tydliga krav och då gäller det att man inte tappar kopplingen mellan dem.

Lif skriver inte om kategorier av krav. Man får utgå ifrån att i hans begrepp arbetssituation ryms funktionalitet och användbarhet och då gäller samma sak som för användbarhetsmål.

3.2 **Sammanfattning av litteraturjämförelse**

Efterhand som vi har jämfört svaren på de frågor vi har utrett i litteraturgenomgången har nya frågor uppstått. Hur ska man kategorisera krav? Är det intressant att prata om icke-funktionella och funktionella krav? Vad är användbarhet och kan man se det som krav fristående från funktionella krav? Vad blir det av användbarhetskraven i vidare design? Vi kommer därför att diskutera kopplingen mellan funktionalitet och användbarhet samt kopplingen mellan design och användbarhet.

3.2.1 *Funktionalitet och användbarhet*

När vi läste Mayhew och ISO 9241-11 som betonar användbarhetsmål upplevde vi att funktioner kom lite i skymundan. Trots allt är det funktionaliteten som är grunden i ett system, vilket traditionell systemutveckling betonar på bekostnad av användbarhet. Vi anser att man i så stor utsträckning som möjligt bör lägga lika mycket vikt vid funktionella aspekter och användbarhetsaspekter. För att förklara vår ståndpunkt ger vi ett exempel: att skicka ett meddelande. Om systemet inte har någon funktion för att skriva in adressen till mottagaren går det givetvis inte att skicka meddelande. Systemet saknar funktionalitet och är därmed värdelöst. Om

funktionen däremot finns med men användarna måste komma ihåg alla adresser i huvudet eftersom det saknas en adressbok kommer systemet troligen upplevas som omständligt och inte användas i samma utsträckning. Man meddelar sig via telefon eller något annat medium istället. Att systemet saknar adressbok innebär att det i teorin går att skicka meddelandet, men i praktiken är det ingen som kommer ihåg alla adresser i huvudet och utan dem går det inte att skicka meddelande. För att funktionaliteten ska kunna användas måste den därför vara användbar, d.v.s. det måste gå att hantera och arbeta med systemet. Vi vill därför betona att funktionalitet till en viss grad är användbarhet. Det är i och för sig också sant att användbarhet till viss del är funktionalitet men inte lika användbart för en diskussion av problemet eftersom användbarhet aldrig är startskottet till utveckling av ett system eller syftet med ett system. Användbarhet kan i och för sig vara startskottet till en vidareutveckling av ett system som har visat sig vara svårt att använda, men problemet är då snarare den bristande graden av användbarhet i funktionaliteten än vice versa.

Utifrån de mått och definitioner som har tagits upp i litteraturgenomgången framgår det att användbarhet finns på olika nivåer och berör olika områden inom ett system. Användbarhet ligger därmed inte på ett enda ställe och uppstår inte vid ett visst skede i en process. Det kommer från många olika ställen och uppstår alltid i kombination av system och kontext.

Frågan som uppstår är då om det finns några fristående användbarhetskrav eller om alla är kopplade till någon funktionalitet. Enligt Sommerville bör man separera funktionella och icke-funktionella krav för att kunna hålla isär vilka krav som är funktionella och vilka som är icke-funktionella. Han påpekar dock att nackdelen med att separera dem är att man tappar kopplingen mellan dem. Vi anser däremot att man inte bör separera användbarhetskrav från andra krav eftersom det är viktigare att behålla kopplingen mellan krav än att kunna skilja på olika kategorier av krav. Kruchten hävdar att det inte är viktigt att dela upp krav i kategorier utan en uppdelning tjänar främst till att kontrollera att man har täckt in alla aspekter vid specificering av krav (Kruchten 2000, s. 157f).

3.2.2 *Design och användbarhet*

Efter att jämfört litteraturen och försökt reda ut svaret på frågan hur användbarhetskrav används tycker vi att det var svårt att se hur de tillämpas efter det att man har satt kraven tills man utvärderar gränssnittet. Det är främst Mayhew som visar att användbarhetskraven är drivande i utformningen av gränssnittet. Lif menar att design är en kreativ process som inte kan visas i en metod, däremot kan man förse designern med krav på gränssnittet. Att sätta krav på gränssnittet anser vi en naturlig följd av att ha specificerat hur interaktionen med systemet ska gå till. Eftersom användbarhet till stor del ligger i gränssnittet vill man försäkra sig om att gränssnittet utformas i enlighet med de användbarhetskrav man har kommit fram till och inte lämnas till de som implementerar att avgöra.

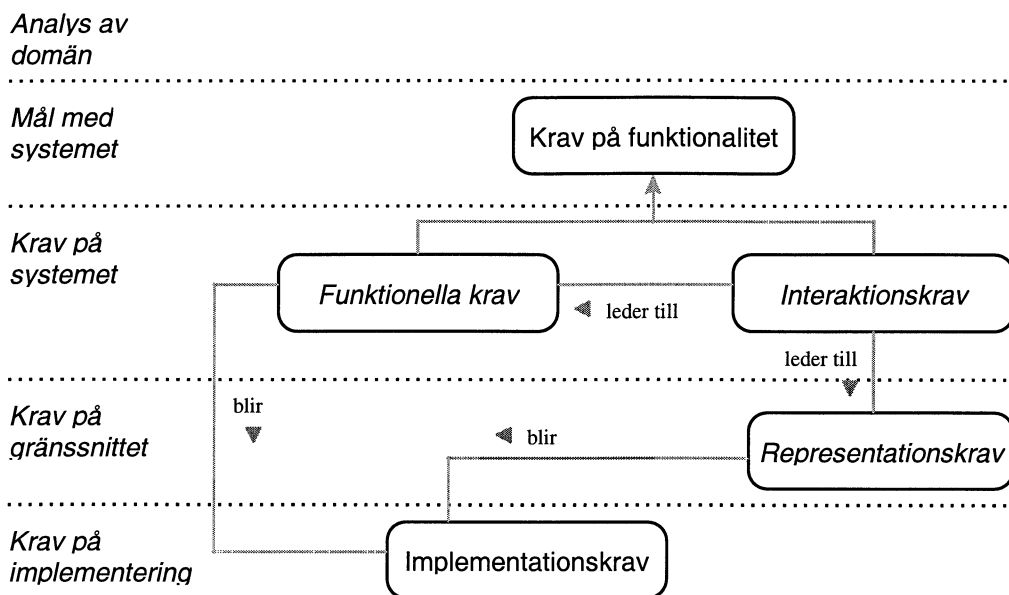
3.3 Förslag på modell

Utifrån resultatet av jämförelsen av litteraturen har vi utvecklat våra tankegångar och kommer att presentera dem i en modell. En modell är ingen avbildning av verkligheten utan ett sätt att representerar en del av verkligheten. Det synsätt en modell visar är inte enda sättet att se på verkligheten och sättet modellen visar synsättet på är inte enda sättet att representera den. Det finns med andra ord olika synsätt på verkligheten och olika sätt att modellera dessa synsätt. Modeller kan användas som utgångspunkt för en diskussion, för att förklara förhållande och samband och därmed ge en vidgad förståelse. För att inte förlora sitt syfte måste en modell vara överblickbar, vilket innebär att en viss förenkling och generalisering måste göras (Holme & Solvang 1991, s. 45).

Modellen är ritad i UML och följer de regler som finns för detta modelleringsätt (Eriksson & Penker, 2000).

3.3.1 *Modellen*

Modellen är indelad i fem nivåer: analys av domän, mål med systemet, krav på systemet, krav på gränssnittet och krav på implementering. Varje nivå beskriver vilka krav som finns på respektive nivå med undantag för den första nivån, analys av domän, där man ännu inte har formulerat några krav. Den andra nivån innehåller krav på systemet men kallas mål med systemet eftersom kraven är på en övergripande nivå och utgör de mål man vill uppnå med att använda systemet. De fem nivåerna utgör stommen i modellen (se figur 5) och kan ses som en del av förloppet i en systemutvecklingsprocess. Dessutom kan modellen till viss del ses som ett förlopp i och med att vissa krav leder till formulering av fler krav på underliggande nivåer. Modellen betonar användbarhet dels genom att på nivån krav på systemet göra en uppdelning i funktionella krav och interaktionskrav och dels genom att visa att man kan sätta krav på gränssnittet.



Figur 5. Översiktlig bild av modellen.

3.3.2 Analys av domän

För att ta reda på vad systemet ska göra och vilken funktion det kommer att ha i den övergripande verksamheten måste en analys göras. Alla systemutvecklingsmetoder beskriver någon form av analys, men framhåller olika tillvägagångssätt och betonar olika områden. Modellen visar fyra områden som vi anser viktiga att undersöka oavsett verksamhet som analyseras och system som ska tas fram (se figur 6). Samma områden återfinns i ISO 9241-11, Mayhew och Lifs analyser. Analysen i modellen syftar till att visa varifrån krav och mål kommer. Att formulera krav på funktionalitet utifrån analys av domänen är ett stort steg som ej behandlas i modellen. Modellen visar därför endast att en relation finns mellan analys och krav på funktionalitet men inte mer specificerat hur relationen ser ut.

De fyra områdena att analysera är:

Uppgift.

Utifrån analys av uppgift får man ut en stor del av funktionaliteten som kommer av uppgiften i sig.

Användare.

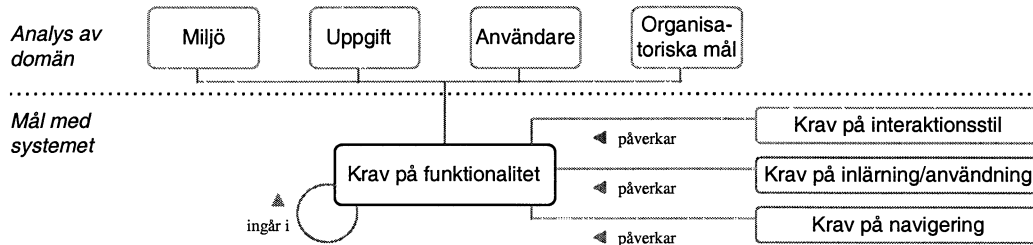
Analys av användare ger en förståelse för de möjligheter och begränsningar som användarna har för användning av systemet.

Miljö.

Analys av miljö kartlägger de specifika möjligheter och begränsningar som arbetsmiljö innebär.

Organisatoriska mål.

Med organisatoriska mål avses alla mål som kommer av att driva verksamheten och de mer övergripande mål som finns i organisationen. Analys av organisatoriska mål ska identifiera de mål som är av betydelse för ett system.



Figur 6. De två översta nivåerna av modellen.

3.3.3 Mål med systemet

Vi anser inte att man ska gå direkt från analys till specificering av precisa användbarhetskrav och funktionella krav utan måste först mellanlanda i en beskrivning av de mer övergripande kraven. I och med att krav på den här nivån är av mer övergripande karaktär kan de ses som mål att sträva efter att uppfylla. Med mål på systemet avses således alla krav på funktionalitet, d.v.s. vad systemet ska göra, vilka problem det ska lösa och vilka uppgifter det ska stödja samt den funktionalitet som kommer av hur systemet ska användas och av vem. ISO 9241-11 och Mayhew använder båda begreppet mål för att beskriva funktionalitet som innefattar både användbarhet och funktioner. Krav på funktionalitet kan jämföras med ISO 9241-11:s definition av de uppsatta målen, d.v.s. att systemet gör vad det ska göra. Man kan också dra paralleller till Löwgrens term relevans d.v.s. hur relevant systemet är för användarna i deras arbete (Löwgren 1993, s. 52). Krav på funktionalitet kan, även om fortfarande på en övergripande nivå, delas in i olika nivåer, dvs ett krav på funktionalitet kan ingå i ett större och mer övergripande krav. Detta visas i modellen genom associationen *ingår i*.

På den här nivån ska man även bestämma vilken interaktionsstil som ska användas, om systemet främst ska vara lätt att lära eller lätt att använda och hur navigeringen i systemet ska gå till (se figur 6). Anledningen till att man tar ställning till dem på denna nivå är att de redan från början påverkar den fortsatta specificeringen av krav.

Ett exempel på detta är att om man har beslutat att ett enkelt och tydligt gränssnitt är viktigare än ett effektivt och kraftfullt arbetssätt påverkar det avsevärt de prioriteringar och beslut som fattas senare. Framförallt Mayhew betonar vikten av att ta ställning till om ett system ska vara lätt att lära eller lätt att använda, men

man kan även finna exempel på detta hos många andra (Sommerville 1998, s. 341; Jordan 1998, s. 12; Newman Lamming 1998, s. 8).

Newman och Lamming menar att valet av interaktionsstil är ett viktigt beslut eftersom det påverkar ett systems användbarhet samt att beslutet tas i början av designfasen (Newman & Lamming 1998, s. 294). Vi menar däremot att beslutet bör tas på ett tidigare stadium om man ska kunna sätta krav på interaktionen innan man kommer till designfasen. Genom att fatta beslutet på ett tidigare stadium sätter man krav på användbarhet och lämnar inte beslutet till implementeringen.

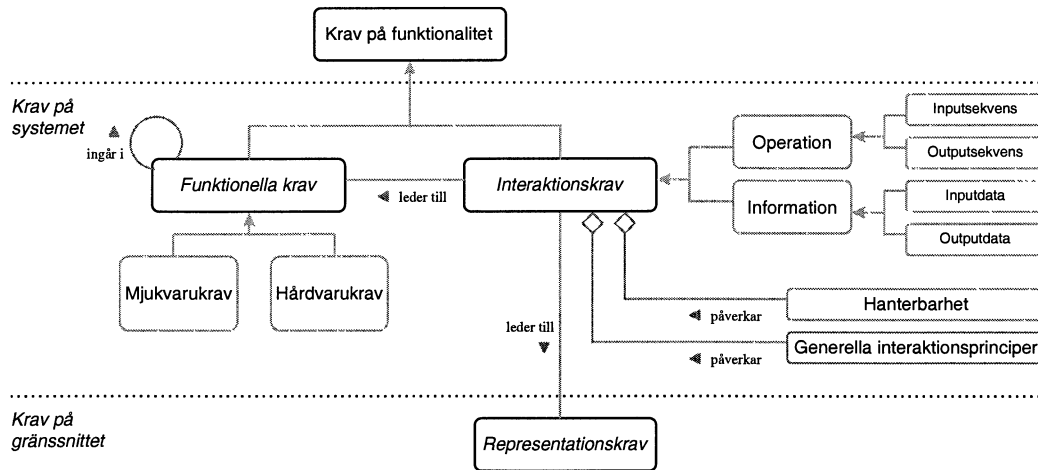
Med navigering avses hur man rör sig mellan olika funktioner i systemet. Krav på navigering innefattar beslut om hur funktioner ska placeras i förhållande till varandra och därmed hur helhetsbilden ska se ut. Det är viktigt att överväga dessa beslut redan i detta skede då det finns en risk att man senare i kravspecificeringen, genom att se på enskilda funktioner, tappat helhetsbilden. Bland annat Lif tar upp detta då han beskriver användningen av work situation model (Lif 1998, s. 91f).

3.3.4 *Krav på systemet*

På nästa nivå krävs en uppdelning av kraven på funktionalitet eftersom de är för övergripande för att vara möjliga att implementera. En uppdelning i funktionella och icke-funktionella krav resulterar i att de icke-funktionella kraven ses som mindre viktiga och mer som begränsningar vilket vi vill komma ifrån. Som påpekats i resonemanget ovan (se kap 3.2.1) har vi kommit fram till att användbarhet är del av all funktionalitet och därför vill vi inte heller göra en uppdelning i funktionella krav och användbarhetskrav. Därmed inte sagt att man inte ska göra någon uppdelning alls av krav. Användbarhetsaspekten finns i vår modell på flera ställen, men hela tiden kopplat till andra krav.

Kraven på systemet består i vår modell av funktionella krav och interaktionskrav (se figur 7). Man kan här dra paralleller till Nielsens definition av användbarhet som ändamålsenlig/funktionell (utility) och användbarhet (usability) (Nielsen 1993, s. 24ff). De funktionella kraven kan liknas vid ändamålsenlighet och interaktionskraven kan liknas vid användbarhet. Vi vill dock betona att interaktionskraven inte är det enda stället där användbarhetsaspekten kommer in.

Funktionella krav är en klass som specialiseras i mjuk- och hårdvarukrav. Funktionella krav kan finnas i olika nivåer vilket visas i modellen genom associationen *ingår i*. Ett mer övergripande funktionellt krav kan bestå av fler funktionella krav som därmed ingår i det mer övergripande. Mjukvarukrav är alla programmeringsrelaterade krav som sätts för att realisera funktionaliteten. Hårdvarukrav är krav på hårdvara som krävs för att realisera funktionalitet.



Figur 7. Den tredje nivån i modellen: Krav på systemet.

Interaktion är kommunikationen som sker mellan systemet och användaren. All denna kommunikation sker via gränssnittet (Newman & Lamming 1998, s. 6) och ska således möjliggöras i designen av gränssnittet. Innan man ser på *hur* gränssnittet ska möjliggöra kommunikationen måste man bestämma *vad* kommunikationen ska bestå av. All interaktion består av operationer och information. Allt som görs i ett system behandlar data antingen i form av den data som matas in, input, eller den data som matas ut, output. I analys och krav på funktionalitet finns all data som behandlas och på nivån krav på systemet ska man specificera vilken data som är input respektive output för enskilda interaktioner. Informationen behandlas i operationer. En operation består av sekvenser av input och output, d.v.s. användaren och systemet utväxlar data. För att utföra en uppgift krävs inmatning och utmatning av information i en viss ordning, d.v.s. en sekvens (Newman & Lamming 1998, s. 276ff). Om en uppgift kan utföras på olika sätt finns det ett antal specificerade input- och outputsekvenser. Olika inputsekvenser kan ge samma outputsekvens, d.v.s. det finns olika sätt att ge samma kommando. Ett exempel på detta är att antingen klicka på en karta för att ange position eller skriva in koordinaterna för positionen, ger samma outputdata i form av en markering av position på kartan. Omvänt gäller att samma inputdata kan ge olika outputsekvenser. Ett exempel på detta är att klicka i kartan kan vara olika kommando vid olika tillfällen och ge olika outputsekvenser. Ett klick i kartan ger position i form av x och y koordinater men betyder olika saker beroende på sekvensen, vad som har gjorts innan.

Utöver att interaktionskrav består av information och operationer påverkas de av generella interaktionsprinciper och hanterbarhetsfaktorer.

Generella interaktionsprinciper syftar till sådana interaktionskrav som grundas på den allmänna samlade kunskap som finns om interaktion. Jordan skriver om fysiska och kognitiva aspekter på användbarhet (Jordan 1998, s. 40) vilket vi anser är en lämplig uppdelning. Kognitiva aspekter är relaterade till hur vi uppfattar saker i omgivningen och grundar sig på kognitionsvetenskap. Exempel på detta

kan vara hur mycket man kan kräva av användaren i form av minneskapacitet. Fysiologiska krav är relaterade till fysiska förutsättningar och är ofta kopplade till ergonomi. Att bestämma vilka tangenter som ska användas för en funktion med hänsyn tagen till vilka fingrar man lättast använder är ett exempel. Man bör även överväga om det ska finnas olika sätt att utföra samma uppgift, t.ex. att använda både tangentbord och mus för att ge samma kommando, för att dels förhindra att belasta samma muskler och dels för att ge variation i utförande för att undvika monoton. I det övervägandet ligger både kognitiva och fysiologiska aspekter.

Hanterbarhet av interaktion är hur operationer och information hanteras. Till skillnad från representationskrav där operationer och information visualiseras handlar det på denna nivå hur hanterbar interaktionen vilket har många olika aspekter. Dessa aspekter kan ses som attribut till objektet hanterbarhet i modellen. Som vi tidigare påpekat är användbarhet beroende av kontexten och eftersom hanterbarhet är en viktig del av användbarhet är även den beroende av kontexten. Exakt vilka aspekter/attribut av hanterbarhet som ska beaktas är således beroende av situationen. Vi har definierat följande attribut som grundläggande aspekter av hanterbarhet:

Effektivitet.

Krav på effektivitet speglar hur effektivt man kan hantera funktionaliteten och kommunikationen. Med effektivitet avses effektivitet i ordets svenska betydelse vilket innefattar engelskans efficiency och effectiveness. Exempel på effektivitet är hur lång tid en uppgift tar att utföra. Helander menar att hur snabbt operationer går att utföra är en av de viktigaste faktorerna för att användarna ska vara nöjda (Gould, Boies & Ukelson 1997, s. 1) och även Sommerville anser att snabbheten är ett mått på användbarhet. En annan aspekt av effektivitet är hur mycket arbete som krävs för att utföra en uppgift. Liknande definition finns på ett flertal ställen i litteraturen i form av termen prestation (Mayhew 1999, s.128; Kruchten 2000, s. 157; Newman & Lamming 1998, s. 151).

Anpassningsförmåga.

Ett systems anpassningsförmåga är i hur stor utsträckning en användare själv kan anpassa interaktionen efter eget behov. Exempel på anpassningsförmåga är att storleken på fönster kan varieras, att användaren kan välja vilka funktioner som visas på skärmen och att det finns möjlighet att skapa egna genvägar. Enligt Lif är det viktigt att ge utrymme för undantag och avvikelser i arbetsrutiner och hos användare och att genom flexibilitet i interaktionen undvika missnöje som annars kan uppstå (Lif 1998, s. 55). Även Sommerville tar upp anpassningsförmåga som innebär hur hårt ett system är knytet till ett bestämt arbetssätt (Sommerville 1998, s. 341).

Felhantering.

Med felhantering avses dels hur man förhindrar att användaren gör fel dels hur man kan återställa systemet efter att fel trots allt har uppstått. Att förhindra att användaren gör fel kan vara att ge tydliga instruktioner, förhindra felaktig inmatning eller att ställa kontrollfrågor med möjlighet att ångra vilket ökar pålitligheten i systemet. Liknande kriterier återfinns hos Bevan (reliability) (Bevan 1994, s. 2). Att återställa systemet efter att fel har gjorts kan vara möjlighet att gå tillbaka och

ändra redan sparade uppgifter. Newman & Lamming skriver om felincidenter och återhämtning från fel (Newman & Lamming 1998, s. 8) och Jordan för samma resonemang (Jordan 1998, s. 30). Felhantering återfinns också hos Sommerville fast definierat som systemets förmåga att återhämta sig från fel användarna gjort och menar att detta är ett användbarhetsattribut. I vår modell avses inte denna definition utan snarare hur användaren kan återställa systemet efter att fel har gjorts.

När det gäller generella designprinciper har vi valt att dela in dem i två nivåer: interaktionsprinciper och designprinciper. Interaktionsprinciper innefattar de principer för interaktionen som inte direkt har med gränssnittets utseende att göra. Gränssnittets utseende faller under designprinciper. Denna uppdelning har vi gjort för att anpassa principerna till uppdelningen utformning av interaktion och utformning av gränssnitt som är gjord i modellen.

3.3.5 *Krav på gränssnitt*

Preece diskuterar operations- och representationsaspekt vad det gäller design av system. Operationsaspekten är hur funktioner hanteras i systemet och representationsaspekten är exakt hur funktionerna presenteras. Krav på interaktion kan därmed ses som den operationella aspekten av design och krav på gränssnittet, vilket är nästa nivå i modellen, kan ses som den representationella (Preece et al. 1994, s. 435).

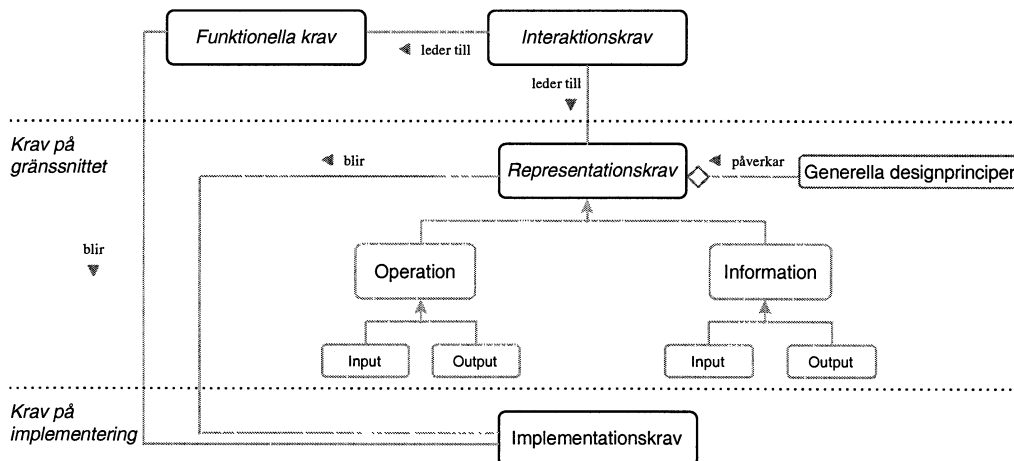
I Lifs objektorienterade synsätt ses all information som informationsobjekt som har attribut och operationer. När det kommer till gränssnittsdesign menar han att man kan ta beslut baserat på de prioriteringar som är gjorda av attribut och operationer (Lif 1998, s. 95ff). Med utgångspunkt i detta resonemang och uppdelningen av interaktionskrav har vi valt att dela upp krav på gränssnittet i information och operationer. Även på denna nivå finns det generella designprinciper som påverkar krav på gränssnittet (se figur 8).

Generella designprinciper innebär att man använder sig av den kunskap som finns om utformning och framställning. Precis som för generella interaktionsprinciper kan man skilja på kognitiva och fysiologiska principer. Kognitiva principer är hur man uppfattar det som visas på skärmen, t.ex. gruppering av knappar, kontraster och färgsättning. Fysiologiska principer är fysiska förutsättningar att beakta, t.ex. ska knappar på skärmen inte vara för små eftersom det kräver större precision att hantera och vilket ökar ansträngningen i armen. Uppdelningen av designprinciper blir nu tydlig genom exemplet att på interaktionsnivå sätter man krav på musen man klickar med medan man på design nivå måste sätta krav på det man ska klicka på, t.ex. en knapp.

Information är precis som det låter den information som visas på skärmen, d.v.s. den data som systemet behandlar. Information som presenteras är output, t.ex. ren text, karta eller en bild. Information som matas in är input, t.ex. hur textfält för inmatning utformas.

Operationer är representationen av de input- och outputsekvenser man har specificerat för interaktionskraven. Representation av operationer är hur man i gränssnittet visar vilken funktionalitet som finns och hur man går tillväga för att använda den. Exempel på detta är hur knappar och textfält ser ut, hur de är grupperade och hur de reagerar när man använder dem.

Det kan vara svårt att skilja på input och output för operationer och information eftersom operationerna alltid behandlar information. Input för operation representerar hur man använder input för information. Ett exempel är kravet att kunna välja mellan man och kvinna. Information är att visa de olika alternativen, i det här fallet man och kvinna. Detta kan visas på flera sätt: man - kvinna, M - F, ♀ - ♂ eller inget alls (man tar för givet att användaren känner till alternativen). Operationen är hur man gör valet, t.ex. knappar, radioknappar, checkbox, textruta för egen ifyllning. Output av en operation representerar vad systemet gör och kommer att göra som resultat av operationen. Output av information är resultatet av samma operation. Ett exempel är att ge kommandot visa könsfördelning. Output av operationen kan vara feedback, i form av ett timglas, som visar att operationen håller på att utföras. Output av information är själva resultatet som presenteras i form av t.ex. ett cirkeldiagram.



Figur 8. Nivå fyra och fem i modellen: Krav på gränssnittet och krav på implementering.

3.3.6 Krav på implementation

Att vissa krav kommer att implementeras medan andra inte kan implementeras direkt utan leder till andra krav som kan implementeras visas genom en uppsamling i slutet som är krav på implementation. Representationskrav kan leda till ytterligare funktionella krav för att realiseras. Det kan ge mjukvarukrav eftersom gränssnittet ska implementeras och därmed kodas. Det kan även ge hårdvarukrav t.ex. en avancerad grafisk presentation som kräver speciell hårdvara i form av grafikort. Funktionella krav som kommer av representationskrav kallas i modellen implementationskrav precis som de funktionella kraven som ska implementeras. Interaktionskrav blir inte implementeringskrav direkt eftersom de implementeras

genom funktionella krav eller representationskrav. De är för övergripande för att kunna implementeras, men däremot kan man använda dem för att kontrollera att man har implementerat de interaktionskrav man har satt.

3.3.7 *Användbarhet i modellen*

Som tidigare påpekats anser vi inte att användbarhet återfinns endast på ett ställe i modellen (se figur 9). Till att börja med kommer användbarhet in på alla nivåer i modellen. Användbarhet återfinns i mål med system som t.ex. i krav på interaktionsstil, krav på inläring/användning och krav på navigering. I krav på systemet återfinns användbarhet i krav operation, krav på information, hanterbarhet och generella interaktionsprinciper. Vidare återfinns användbarhet i krav på gränssnitt i operation, information och generella designprinciper.

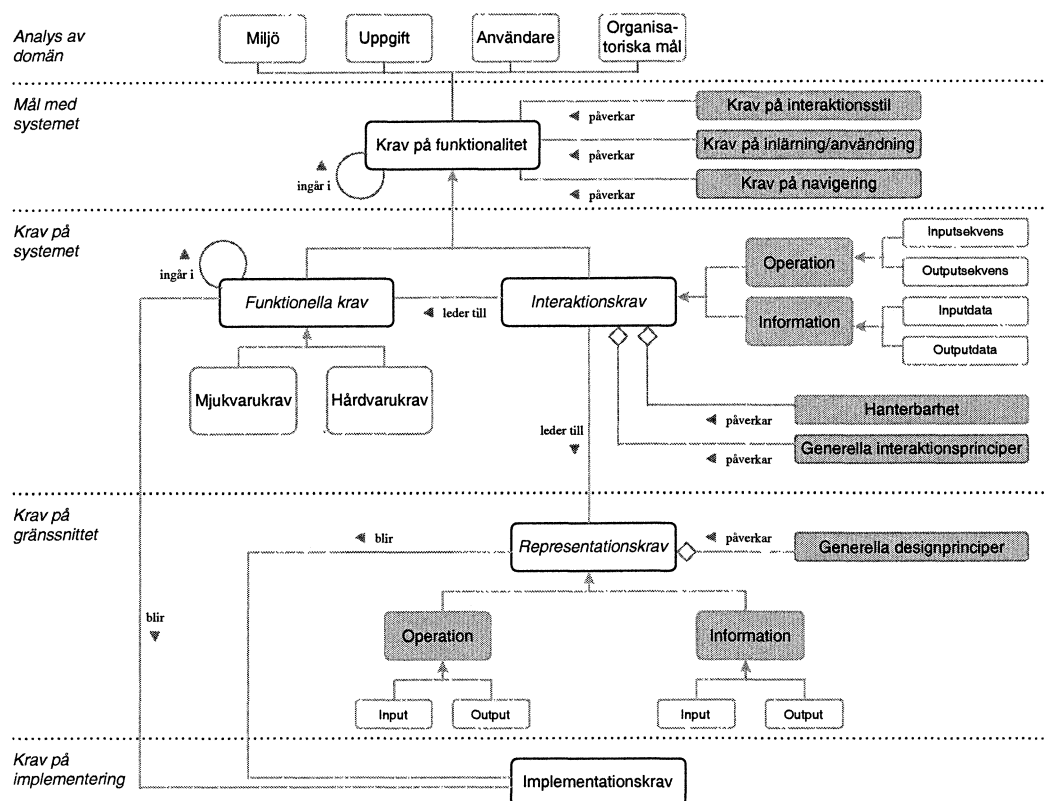
Redan på nivån mål med systemet tar man ställning till användbarhetsfrågor. De beslut man fattar här är övergripande och sätter prägel på hela kravspecificeringen. För att senare i kravspecificeringen kunna specificera användbarhet i detalj måste först en övergripande riktlinje finnas till hands.

I och med att operation på nivån krav på systemet delas in i input- och output-sekvenser överväger man vilken information användaren respektive systemet har och i vilket skede viss information finns tillgänglig. Genom att göra så försäkrar man sig om att interaktionen inte kräver information som inte är tillgänglig eller lätt att tillgå. Det innebär också att output förser användaren med den eventuella information som behövs för fortsatt interaktion och att output kommer i den ordning användaren vill ha resultatet presenterat. Eftersom information är nära kopplat till operation kommer användbarhet även in här.

Att användbarhet finns på olika ställen blir tydligt genom att t.ex. hanterbarhet beaktar anpassningsförmåga och att operation har olika tänkbara sekvenser för input och output vilket i sig innebär en möjlig anpassning.

Den verkliga interaktion sker genom gränssnittet eftersom det är den enda del av systemet som användaren möter varför denna del av modellen är väldigt viktig för användbarheten. Därmed inte sagt att användbarhet återfinns till störst del här utan det är här den genom representation realiseras och kommer till uttryck.

Implementationskrav är resultatet av alla krav, d.v.s. alla krav blir i slutändan funktionella krav. Genom att samla alla krav till implementationskrav behöver man inte ta hänsyn till vilken kategori ett krav tillhör. Alla krav kan ses funktionella krav, som vi kallar implementationskrav, då allt implementeras som mjuk- eller hårdvarukrav. Det viktiga är att användbarhetsaspekten genom interaktionskrav och representationskrav har påverkat och lett till funktionella krav. Genom att implementera dessa krav blir resultatet ett användbart system utan att hänsyn har tagits till användbarhetskrav utan koppling till funktionella krav.



Figur 9. Användbarhet i modellen.

3.3.8 Exempel på kravhantering

För att förtydliga modellen ges nedan ett enkelt exempel på olika kategorier av krav. Exemplet visar hur ett övergripande krav på funktionalitet delas upp i olika krav. För varje kategori av krav ges endast ett exempel och därmed visas inte alla tänkbara krav som uppstår av ett krav på funktionalitet.

Krav på funktionalitet:

Kartor ska kunna användas i systemet
 Zoomfunktion ska finnas för karta
 Direktmanipulation för karthantering
 Prioritera lätt att använda då funktionen är tidkritisk och tid för utbildning finns.
 Kartbilden ska inte lämnas vid zoomning

Krav på interaktionsstil:

Krav på inlärninng/användning:

Krav på navigering:

Funktionella krav

Mjukvarukrav:

En funktion som känner av var i kartan det har klickats.

Hårdvarukrav:

Visst grafikkort för karthantering

Interaktionskrav

Operation:Inputsekvens:	Klick på knapp för zoomning, klick i kartan.
Operation:Outputsekvens:	Feedback att bearbetning pågår.
Information:Input:	Grad av inzoomning, position på kartan.
Information:Output:	Ny kartbild visas.
Hanterbarhet:	Max två klick för att visa ny kartbild.
Generell interaktionsprincip:	Klicka med den vänstra knappen på musen (vanlig funktion där man bör använda det fingret som man lättast klickar med).

Representationskrav

Generell designprincip:	förstoringsglas på knappen för zoomning (lättförståelig metafor).
Operation:Input:	Zooma knappen ska placeras nära kartan (på ett lättillgängligt ställe).
Operation:Output:	Pekaren blir ett förstoringsglas vid zoomning (för att veta vad man gör).
Information:Input:	På knappen står aktuell grad för inzoomning.
Information:Output:	Lämlig detaljnivå för ny kartbild.

Implementationskrav

(som kommer av representationskrav): Pekaren ska byta utseende vid zoomning.

Funktionellt krav

(som kommer av interaktionskrav): Funktion för olika grader för inzoomning.

4 Slutsats

Frågeställningen som uppsatsen syftar till att utreda lyder: Hur relaterar man användbarhetskrav till funktionella krav i en systemutvecklingsprocess? För att utreda frågeställningen har vi valt ut litteratur vi anser relevant att studera. Litteraturen gav upphov till diskussioner och många skisser av tänkbara synsätt och modeller. En del litteratur har varit svårt att förstå och ta till sig på grund av att ämnet på många punkter är öppet för tolkning. Design anses t.ex. av många vara lite av en konstform som inte fullt ut kan beskrivas eller inrymmas i metoder och modeller vilket gör det svårt att reda ut vad som egentligen avses i definitionerna. Efterhand märker man dock att resonemang återkommer på flera ställen i litteraturen och att man kan styrka sina tolkningar utifrån flera källor även om man inte hittar exakt samma förklaring i en och samma referens.

Modellen ska ses som resultatet av litteraturgenomgången och syftar till att visa vårt perspektiv på användbarhet i kravspecificering. Modellen är ett första utkast som inte ska ses som en fulländad modell då tiden, trots hårt arbete, har varit knapp för en uppgift av detta slag. En tänkbar fortsättning för modellen är en utvärdering av användningen i FOI:s projekt och utifrån det vidareutveckla och förbättra modellen. En annan tänkbar fortsättning är att studera mer litteratur, då det finns mycket publicerat inom problemområdet, och på så sätt vidga perspektivet på användbarhet och funktionalitet. Problemet med ett så hårt tidsbegränsat arbete, som en kandidatuppsats, är just de avgränsningar man tvingas att göra.

Det vi främst vill framhålla i vårt resultat är det faktum att användbarhet inte finns på ett ställe i modellen utan överallt där man finner funktionalitet. Funktionalitet är med andra ord till viss del användbarhet. Vi tycker att detta synsätt är ett intressant alternativ till det traditionella synsättet med uppdelning av funktionella och icke-funktionella krav. Genom att peka på att användbarhet ligger överallt och inte enbart i gränssnittsdesign visar man att användbarhet är något man måste ta ställning till tidigt i utvecklingsprocessen. Genom att inkludera användbarhet i funktionella krav har man lyft upp användbarheten. Detta kommer till uttryck i modellen genom implementationskrav där funktionella krav och representationskrav utgör systemets funktionalitet.

Vi har stött på olika uppfattningar om hur man kommer från kravspecifikation till gränssnitt och vill visa att ett tänkbart sätt är att utifrån interaktionskrav sätta krav på gränssnittsdesign. Fördelen med att göra detta är att man inte lämnar det upp till den som implementerar systemet att avgöra hur användbarhet ska utformas.

Den som har gjort analysen och specificerat kraven ska också ta beslut angående gränssnittet in den mån gränssnittsbesluten ska påverkas av analys och användbarhetsspecificering. Det finns även gränssnittsbeslut som är av ren estetisk karaktär vilket en designer är mer lämpad att fatta. Till viss del löser krav på gränssnittet svårigheten med att utifrån en formulering av vad användbarhet är komma till hur det ska implementeras. Krav på gränssnittet grundar sig i analysen och på vägen har man tagit hänsyn till olika användbarhetsaspekter och på så sätt successivt närmat sig designen. Främst genom att skilja på interaktion och representation, men även genom att dela in interaktionen i dialog, hanterbarhet och generella principer och därefter dela in dem, åstadkommer man detta successiva närmande.

Genom att göra en uppdelning i funktionella krav, interaktionskrav och representationskrav och i den uppdelningen visa att användbarhet ligger på flera ställen anser vi att vi har visat på ett nytt sätt att behandla användbarhet vid kravspecifisering.

Referensförteckning

- Backman, J. (1998). *Rapporter och uppsatser*. Lund: Studentlitteratur.
- Bevan, N. (1994). *Measuring usability as quality of use*. Opublicerat manuskript, National Physical Laboratory, Storbritannien.
- Bevan, N. (1997). *Quality and usability: A new framework*.
- Eriksson, H-E., & Penker, M. (2000). *Business Modeling With UML - Business Patterns at Work*. New York: Wiley.
- Helander, M. G., Landauer, T. K., & Prabhu, P. V. (1997). *Handbook of Human-Computer Interaction* (2nd ed.). Amsterdam: North-Holland.
- Holme, I. M., & Solvang, B. K. (1991). *Forskningsmetodik – om kvalitativa och kvantitativa metoder*. Lund: Studentlitteratur.
- ISO. (1998). *ISO 9241-11 Ergonomic Requirements For Office Work With Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*. Geneva: International Organization for Standardization.
- ISO. (1999a). *Introduction to ISO* [WWW dokument]. URL <http://www.iso.ch/infoe/intro.htm>. Accessdatum: 2001-05-07.
- ISO (1999b). *ISO 13407 Human-centred Design Process For Interactive Systems*. Geneva: International Organization for Standardization.
- Jordan, P. W. (1998). *An Introduction to Usability*. London: Taylor & Francis.
- Kruchten, P (2000). *The Rational Unified Process An Introduction* (2nd ed.). Boston: Addison-Wesley.
- Lif, M. (1998). *Adding usability - Methods for Modelling, User Interface Design and Evaluation* (avhandling för doktorsexamen, Uppsala universitet).
- Löwgren, J. (1993). *Human-computer interaction What every system developer should know*. Lund: Studentlitteratur.
- Mayhew, D. J. (1999). *The Usability Engineering Lifecycle - A Practitioner's Handbook For User Interface Design*. San Fransisco: Morgan Kaufmann.
- Mullet, K., & Sano, D. (1995). *Designing Visual Interfaces - Communication Oriented Techniques*. NJ: Prentice Hall.

Newman, W., & M. Lamming, M. G. (1998). *Interactive System Design*. Harlow: Addison-Wesley.

Nielsen, J. (1993). *Usability Engineering*. San Diego: Academic Press.

Olerup, A., & Steen, O. (2001). *Utformning av rapporter i informatik, vt. 2001*. Opublicerat manuskript, Lunds universitet, Institutionen för Informatik.

Oxford Advanced Learner's Dictionary. (1995). Oxford: Oxford University Press.

Patel, R., & Davidsson, B. (1998). *Forskningsmetodikens grunder - Att planera, genomföra och rapportera en undersökning* (2:a uppl.). Lund: Studentlitteratur.

Preece, J. et al. (1994). *Human-Computer Interaction*. Harlow: Addison-Wesley.

Prismas Engelska ordbok. (1993). Norge: Prisma.

Sommerville, I. (1998). *Software Engineering* (5th ed.). Harlow: Addison-Wesley.