

Forord

Dette speciale er udarbejdet af Lasse Høgh og Rasmus Flyger Berg Andersen, studerende på 10. semester Informatik ved Aalborg Universitets institut for Datalogi, og er baseret på arbejde udført af gruppen i perioden februar 2005 til juni 2005.

Vi vil gerne takke de 6 falckreddere fra falckstationen i Nibe, som har hjulpet ved at deltage som testpersoner ved brugbarhedseksperimenterne. Derudover vil vi gerne takke John Gade og Frederik Vindum Jensen fra Judex A/S som udover at have stillet amPHI systemet til rådighed for os og været løbende sparringspartnere, har formidlet kontakten til Falck. Desuden vil vi gerne takke vores medstuderende Berith Andersen, Martin Lindholdt Jørgensen og Ulrik Kold for deltagelse i kreativt arbejde og pilottest, og vores vejleder Jan Stage for løbende sparring, og kreative forslag til hvad vi ”Også” kunne gøre med vores speciale.

Lasse Høgh

Rasmus Flyger Berg Andersen

Indholdsfortegnelse

1	Indledning	5
1.1	Motivation.....	6
1.2	Problemformulering	8
1.3	Afgrænsning.....	9
2	Multiplatform systemer.....	11
2.1	Karakteristika	11
2.2	Usability faktorer	12
2.3	Udvikling	13
2.3.1	Diverse tilgange	13
2.4	Design by Graceful Degradation.....	15
2.5	Kontinuitet	18
2.5.1	Graceful Degradation kontinuitet.....	18
2.5.2	Videnskcontinuitet	19
2.6	Vores overvejelser.....	20
3	Contextual Design.....	23
3.1	Beskrivelse af Contextual Design	23
3.1.1	Dataindsamling	24
3.1.2	Modellering	24
3.1.3	Konsolidering.....	25
3.1.4	Design	25
3.2	Brug af Contextual Design.....	26
3.2.1	Contextual Inquiry Interviews	27
3.2.2	Bygning af Affinity-diagram	29
3.2.3	Visioning.....	32
3.2.4	Mock-ups	34
3.3	Vores erfaringer med Contextual Design.....	36
4	Udvikling af programmet.....	39
5	Programmer.....	45
5.1	Tablet PC.....	46
5.2	Pocket PC	50
6	Beskrivelse af eksperiment	57
6.1	Planlægning af test	57
6.2	Testpersoner	59
6.3	Pilot og prototype-test	59
6.4	Opgaver.....	60
6.5	Omgivelser og udstyr	61
6.6	Databehandling	61
6.6.1	Gennemsyn af videooptagelser	61
6.6.2	Evaluatoreffekt.....	62

6.6.3	Arbejdsbelastningsdata	64
6.6.4	Brugbarhedsmålinger	65
7	Testresultater	67
7.1	Brugbarhedsproblemer	67
7.1.1	Antal brugbarhedsproblemer	67
7.1.2	Typer af brugbarhedsproblemer	71
7.1.3	Delkonklusion	74
7.2	Arbejdsbelastning	75
7.2.1	Analyse	75
7.2.2	Delkonklusion	79
7.3	Måling af usability	81
7.3.1	Analyse	81
7.3.2	Delkonklusion	92
8	Konklusion, begrænsninger og videre arbejde	95
8.1	Konklusion	95
8.2	Begrænsninger	99
8.3	Videre arbejde	100
9	Appendix	101
9.1	Appendix A – Pocket PC Limited	101
9.2	Appendix B – Judex’ udvikling af programmel	103
9.3	Appendix C – Context-awareness	111
10	Summary	113
11	Litteraturliste	115

1 Indledning

I kraft af at computere efterhånden begynder at blive allestedsnærværende i vores samfund, er der opstået et behov for at kunne tilgå data på flere forskellige måder. Det er ikke længere nok at have en computer ved arbejdspladsen, man skal også have en PDA, en mobiltelefon og en laptop, som alle helst skal kunne gå på Internettet, tjekke e-mail, spille spil og beregne atomprøvesprængninger.

Dette betyder, at flere programmer laves til andre platforme end den almindelige PC, samtidig med at antallet af mulige operativsystemer er eksploderet – fra de gamle arbejdsheste MS-DOS/Windows, Mac-OS, Solaris og Unix, er der nu en hob af Linux/Unix varianter, samtidig med at der bruges forskellige varianter af MS-Windows til såvel servere og arbejdsstationer, som hjemmecomputere. Selv PDA'er og mobiltelefoner kan køre Windows-CE, Symbian OS eller PALM OS. Alle disse platforme, kombineret med ønsket om at kunne tilgå sine data overalt, gerne via de programmer man er vant til, har afstedkommet et ønske om flere og bedre multiplatforme systemer, ligesom der er opstået et felt indenfor Computer Science, der beskæftiger sig med måden hvorpå systemer kan overføres fra en platform til en anden.

Multiplatform systemer er edb-systemer, der kører på flere forskellige typer platforme. Dette kan grundlæggende tage sig ud på to måder. Et system kan være uafhængigt af operativsystemet, hvilket programmeringssproget Java er et eksempel på. Et Java-program kan køre på alle platforme, hvortil der eksisterer en Virtual Machine, der kan køre programmet. Java er dermed yderst portabelt på tværs af operativsystemer, og altså multiplatformt. I denne betydning er alle programmer dog i princippet multiplatforme, så længe programmet kan kompileres til at køre på mere end én platform.

En anden måde at tænke multiplatform på er dog, at ikke alle apparater har de samme ressourcer til rådighed. Dette betyder, at et system, der er lavet til en bestemt platform, ikke nødvendigvis vil kunne bruges på en anden, eksempelvis grundet at den ene platform har flere ressourcer til rådighed end den anden, eller at brugerens muligheder for at interagere med platformene er forskellige. I dette tilfælde kan et system dog stadig være et multiplatform system, hvis der blot laves en version af programmet, som tilbyder en ækvivalent funktionalitet på dette alternative apparat. Eksempelvis kan et selvbetjeningssystem, som tilbyder den samme funktionalitet på Internettet og via telefonen siges at være et multiplatform system.

Multiplatform systemer er i vores forståelse en disciplin indenfor informatikken, hvori omdrejningspunktet er, at tilbyde brugeren en funktionalitet på forskellige platforme, således, at en given opgave kan løses på forskellige måder og i forskellige situationer, alt efter hvilket apparat brugeren benytter. Dette kræver et system, hvori det er indtænkt hvorledes systemet skal bruges i den enkelte situation, hvor brugergrænseflade og funktionalitet passer til situationen, således at brugeren kan udføre sit arbejde, uden at blive forstyrret unødigt af systemet, og med sikkerhed for, at hans data er sikre.

Det enkelte multiplatform system ser vi som værende to eller flere apparater, som tillader en bruger at udføre en sekvens af sammenhængende opgaver ved hjælp af disse apparater. Ideelt vil de enkelte apparater i et multiplatform system have samme opførsel, og føles som det samme system, på trods af apparaternes forskelligheder. Det er klart at et program som ikke reagerer konsistent på brugerinput, vil have dårlig brugbarhed. Dette gælder også for multiplatform systemer. Hvis en bruger kan benytte sin mentale model af et system for et apparat, i brugen af et andet, hæver dette hele systemets brugbarhed, idet kontinuiteten i systemets opførsel således tilbyder brugeren en fuldstændig konsistent, ”seamless”, overgang fra et apparat til et andet, og forstærker følelsen af at apparaterne tilhører det samme system.

1.1 Motivation

Vi har igennem vores arbejde med brugbarhed, fået en interesse i at betragte de omstændigheder der er betydende for at der opstår brugbarhedsproblemer. I denne forbindelse har vi set, at der er en lang række betydende faktorer for, om et edb-system er brugbart eller ej.

Det er en kendt sag, at ting skal ”ligne” det de gør – det der kaldes affordance [Nor90]. Dette gælder naturligvis også for edb-systemer, hvor grafik og farver kan bruges til at formgive en brugergrænseflades interaktionselementer, således at de støtter brugerens intuitive forståelse af hvad disse gør. Samtidig findes der en række regler – såkaldte ”gestaltlove” der beskriver den kognitive forståelse, som mennesker har, af at objekter kan være grupperende eller adskilte [Mol00]. Også ren tekst kan være mere eller mindre brugbar – udover at et højt eller lavt lixtal groft sagt kan siges at være fordelagtigt i forskellige situationer. Jakob Nielsen har for eksempel lavet en række undersøgelser af brugbarheden af selve *teksten* på websider [Nie97]. Der er lavet undersøgelser af selve de test man bruger til at evaluere brugbarhed, for at se hvad der påvirker disse, eksempelvis

om omgivelserne, typen af testpersoner eller multitasking har betydning for resultaterne [Kje05][Bec02].

Vi har set eksempler på, at mobile og stationære systemer er blevet aftestet, og i et vist omfang er de mobile systemers præstationer sammenlignet med de stationæres. Vi ser dog, i multiplatform systemer, en unik mulighed for at betragte og sammenligne egenskaberne for forskellige typer platforme, uden den interferens, som forskelligheden i platformenes software, normalt skaber. Således mener vi, at når et system kan bruges på flere typer platform, vil det kunne tydeliggøres, hvilke problemer i brugen der skyldes den specifikke platform, og hvilke der har baggrund i udformningen af systemets funktioner.

1.2 Problemformulering

Formålet med dette projekt er at undersøge brugbarhed i multiplatform systemer, hvordan man sikrer at et systems brugbarhed er så høj som muligt, og hvilke problemer der er i brugen af forskellige typer apparater, på trods af dette.

Hovedspørgsmål

- Hvilke problemer er der i at skabe og fastholde brugbarhed i multiplatform systemer?

Multiplatform systemer er specielle ved, at sådanne systemer skal køre på vidt forskellige platforme, og samtidig virke på en konsistent måde på tværs af disse platforme. Et system der har høj brugbarhed på en platform, bibeholder ikke nødvendigvis denne brugbarhed på en markant anderledes platform.

Delspørgsmål 1

- Hvad eksisterer i litteraturen omkring det at lave brugergrænseflader, som skaber og fastholder usability på multiplatform systemer?

For at kunne belyse denne problemstilling vil vi først undersøge HCI-litteraturen omkring multiplatform systemer, for at se hvilke tilgange der findes til udvikling af sådanne systemer, hvorledes disse forsøger at sikre brugbarhed, og hvilke begreber disse bruger i deres bestræbelser.

Delspørgsmål 2

- Hvorledes kan vi udvikle en brugergrænseflade til et multiplatform system?
 - Hvilken betydning har valget af udviklingsmetode?
 - Hvad skal vi være opmærksomme på i udviklingen, hvis vi vil sikre konsistens og brugbarhed i et multiplatform system?

Med udgangspunkt i delspørgsmål 2, vil vi fremstille et multiplatform system, som har den nøjagtig samme funktionalitet, på to forskellige platforme. Vi vil således kunne afprøve nogle af de værktøjer, som har til formål at sikre konsistens og kontinuitet, og brugbarhed generelt, på tværs af platforme. Dermed vil vi få mulighed for at vurdere effekten af disse, og vurdere om de var formålstjenstlige i det specifikke udviklingsforløb.

Delspørgsmål 3

- ❑ Er der problemer i forhold til brugbarhed, som adskiller den ene platform i et multiplatform system fra den anden?
 - Hvilke typer problemer er typiske for henholdsvis Tablet PC'er og Pocket PC'er, og hvad er grunden til at disse problemer opstår, på den specifikke type af platform?
 - Hvilke forskelle og ligheder er der mellem Ajour og amPHI, og hvorfor opstår disse?
 - Er der en sammenhæng mellem de fundne problemer, og de emner der behandles i litteraturen?

I denne del af projektet vil vi benytte vores system, sammen med Judex' amPHI-system, til at lave brugbarheds-tests, med henblik på at finde ud af, om der er bestemte aspekter af brugbarhed, som er karakteristiske for multiplatform systemer. Dette vil give os mulighed for at få et indblik i de egenskaber som den specifikke platform har, samtidig med, at vi vil få et indblik i, hvorvidt de valg vi har gjort i udviklingen har været fornuftige.

1.3 Afgrænsning

Dette projekt har fokus på de multiplatforme systemer, og på brugbarheden af disse. Dette betyder, at vi har fravalgt at beskæftige os med systemernes underliggende hardware, det metodiske i selve programmeringen af sådanne systemer, og de værktøjer der eksister til udvikling af disse. Ligeledes vil vi ikke arbejde med at programmere et reelt fungerende system, da vores interesse ligger i brugerens interaktion med brugergrænsefladen på det enkelte apparat. Dette betyder, at vi ikke vil arbejde med systemets "back-end" eller med den underliggende kommunikation mellem de forskellige dele af systemet.

Brugertestningsmæssigt vælger vi også at se på et enkelt forhold, falckredderes interaktion med systemet. Dette betyder, at der i dette projekt ikke vil være overvejelser i forhold til, om forholdene for de brugbarhedseksperimenter vi laver, har betydning for resultaterne.

2 Multiplatform systemer

Vi vil i dette afsnit sætte fokus på eksisterende litteratur indenfor HCI-området omkring multiplatforme brugergrænseflader. Vi vil primært fokusere dette studie på selve brugergrænsefladen, og på de guidelines og teorier der eksisterer her indenfor. Vores sekundære fokus vil være på hvornår brugergrænsefladerne rent udviklingsmæssigt bliver til. Nærmere betegnet ønskes det belyst, hvornår i udviklingsprocessen multiplatform systemers brugergrænseflader splittes op i forhold til de specifikke apparater der udvikles til. Målet med dette afsnit er at synliggøre, om der eksisterer retningslinier for, hvordan selve brugergrænsefladen designes for multiplatforme systemer, og hvornår i en udviklingsproces brugergrænsefladen bør designes i forhold til det specifikke apparat. Det underliggende programmeringssprog og dertilhørende programmeringsmodel er ikke af interesse for dette projekt.

2.1 Karakteristika

I [Sef04] beskriver forfatterne Seffah, Forbrig og Javahery deres syn på brugergrænseflader på multiplatform systemer, og på de udfordringer der ligger i dette forskningsområde. De bruger begrebet Multiple-User Interface (MUI), hvilket de beskriver som værende et aktivt system, som giver adgang til information og ydelser fra forskellige. En platform er i [Sef04] forstået som værende en kombination af henholdsvis computerhardware, operativsystem og et User-Interface (UI) toolkit, hvilket er tilsvarende til den definition der gives i [Flo04]. En platform kan være alt lige fra traditionelle kontor-pc'er, til bærbare computere, PDA'er og mobiltelefoner. [Sef04] opstiller følgende punkter for at give en mere dybdegående beskrivelse af et MUI.

Et MUI:

- Tillader at en enkelt bruger eller gruppe af brugere kan interagere med serverside services og information ved at bruge forskellige interaktionsredskaber.
- Tillader en enkelt bruger eller en gruppe af brugere at udføre en sekvens af sammenhængende opgaver på forskellige apparater.
- Præsenterer features og information der opfører sig ens på tværs af platforme, selvom hver platform/apparat har dens eget specifikke "look-and-feel".

- Føles som en variation af et enkelt interface for forskellige apparater med de samme muligheder.

MUI giver ifølge [Sef04] adskillige ”views” af den samme model, og koordinerer brugerens handlinger indsamlet på de forskellige apparater. Hvert ”view” kan ses som værende et komplet UI for en specifik platform, hvor alle disse ”komplette” UI’er former og udgør et enkelt MUI. Listen af features, måden at interagere på, og den viste information kan afvige fra den ene platform til den anden.

2.2 Usability faktorer

Ud over den gængse måde at forstå et MUI på, er det interessant at sætte fokus på, hvilke brugbarhedsmæssige overvejelser, der bør laves ved design af MUI’s på forskellige platforme. [Sef04] beskriver en række brugbarheds-faktorer der bør tages i betragtning i forbindelse med MUI design:

- *Måden der interageres på:* Grundet de mobile apparaters begrænsninger, må UI’et være markant anderledes fra det tilsvarende UI på en PC.
- *Skalerbarheden af information:* Typisk er applikationer, som kører på platforme der er begrænsede og har lav båndbredde, mere restriktive, da de ikke kan gengive det samme ”rige” informationsindhold, sammenlignet med UI’et på en mere traditionel UI platform.
- *Tilpasning af opgavestruktur:* Ved overførsel af opgavestrukturen fra en platform til en anden skal denne oftest tilpasses. Det er muligt, at opgaver udført på stationære apparater, er forskellige fra opgaver udført i mobile omgivelser. Ved MUI bør overførslen af opgaver opretholde support for den samme funktionalitet og feedback selvom bestemte features eller variationer er elimineret ved enkelte platforme.
- *Visuel meningsfuldhed:* Det kan være tilfældet, at UI’s med reduceret båndbredde i højere grad benytter sig af mere grafiske elementer med semantisk værdi for at kompensere for det manglende display. Forskelle i visuelle features apparaterne imellem kan skabe brugbarhedsproblemer. Inkonsistens af denne type kan forstyrre den analogiske overførselsproces, ved at forhindre brugere i at overføre deres indledende forståelse af apparatets ydelser.

- *Platformens restriktioner og muligheder:* På grund af forskellige apparaters restriktioner, specielt med båndbredde, må UI'et på et trådløst håndsat være fundamentalt anderledes end UI på et traditionelt UI applikation. Apparatets karakteristika og begrænsninger skal overvejes i forhold til deres umiddelbare effekt på tiden det tager at udføre opgaverne, for således at opnå den samme brugbarhed for mobile UI'er, som den der eksisterer på de mere traditionelle UI'er.
- *Opførsel:* Et MUI kan have forskelligt "look-and-feel" samtidig med at det opretholder den samme opførsel på tværs af forskellige platforme.
- *Brugerens opmærksomhed på trade-off:* Det er acceptabelt at have en simplificeret version af et program, der ekskluderer bestemte ekstra features, som er tilgængelige i en mere avanceret version.
- *Ydelsernes tilgængelighed:* Det er ikke nødvendigt at gøre alle features tilgængelige på alle apparater.

Ifølge [Sef04] er ovenstående karakteristika og begrænsninger ikke dele af nuværende udviklingsteknologier, men er interessante i forhold til MUI konceptet, og bør benyttes sammen med en udviklingsmodel, som guidelines til at designe eller evaluere aspekter af brugbarhed for et MUI.

2.3 Udvikling

Inden for udviklingen af multiplatform systemer eksisterer der en række forskellige udviklingsmodeller, som kan være givende at benytte, alt efter hvilken case man arbejder med. Et indgående kendskab til disse udviklingsmodeller er ikke interessant i dette projekt, men vi ønsker blot at give et mere generelt overblik over modellernes udviklingsforløb. Mere præcist ønsker vi her at belyse, hvornår i processen der designes i forhold til de specifikke platforme, hvis denne form for opdeling finder sted.

2.3.1 Diverse tilgange

I [Flo04] gives et overblik over de udviklingsmodeller der hidtil har været benyttet inden for MUI udvikling. Disse tilgange kan opdeles på følgende måde:

Platform-afhængig udvikling:

Denne form for udvikling tager udgangspunkt i den enkelte platform. Dette betyder at for hver platform der indgår i et system, designs der et unikt interface. Udviklingen af UI til forskellige platforme kører således uafhængigt af og sideløbende med hinanden.

Platform-uafhængig udvikling:

Platform-uafhængig udvikling består i alt sin enkelthed af design af et enkelt interface som vil kunne "køre" på adskillige platforme. Måden hvorved dette interface tilgås fra de forskellige platforme sker gennem disses indbyggede browsere. Ansvar for at få vist information korrekt på de forskellige platforme, er således ikke i hænderne på udviklerne, men afhænger af de platform-specifikke browsere. Udviklerne skal naturligvis tage højde for forskellige browsere og standarder på platformene, men den rent visuelle visning af interfacet, er i høj grad op til den enkelte browser.

Tilpasset platform-uafhængig udvikling:

Udviklingen af denne type sker ved udarbejdelsen af en beskrivelse der dækker alle de dele de forskellige platforme har til fælles. På baggrund af denne beskrivelse udvikles et grundlæggende interface, der typisk vil være tilsvarende til det der er beskrevet i "Platform-uafhængig udvikling". Forskellen består dog i, at der fra dette punkt udvikles yderligere beskrivelser, der tager højde for de unikke egenskaber den enkelte platform måtte have.

Modelbaseret platformudvikling:

I [Sef04] benyttes en tilgang kaldet "*Task model-based development*". Denne tilgang tager udgangspunkt i et model-baseret paradigme som benytter en central vidensdatabase til at gemme en beskrivelse af alle aspekter af et interface design. Den centrale beskrivelse kaldes for en model, og indeholder typisk information omkring de opgaver brugeren forventes at udføre ved brug af applikationen, data i applikationen, de kommandoer brugeren kan benytte, interfacets præsentation og opførsel, og en karakteristik af brugerne. Ideen er automatisk at producere UI prototyper ud fra en abstrakt generisk model.

En anden modelbaseret tilgang er "*design by graceful degradation*". Der argumenteres i [Flo04] for, at det er muligt at centrere design anstrengelserne omkring et enkelt "rod-interface" eller

”kilde-interface”. Dette interface eksisterer på en platform der kun i meget lille omfang er begrænset, typisk en PC, og ud fra dette kan yderligere brugergrænseflader fremstilles, ved at benytte en række regler.

2.4 Design by Graceful Degradation

Vi har valgt at benytte ”*design by graceful degradation*”. Vi har foretaget dette valg på baggrund af vores ønske om en tilgang til udviklingen, der kunne give os nogle konkrete regler og værktøjer at arbejde med. Vi har desuden set, at denne tilgang var mere velovervejet end mere ”tilfældige” og ad hoc tilgange. Endelig har vi set, at der i denne tilgang tages hensyn til, at der arbejdes med meget forskellige platforme. Tilgangen forsøger at sikre, at der er kontinuitet i den måde som disse platforme fungerer på, således at brugeren føler, at de forskellige platforme er dele af et sammenhængende system.

I ”*design by graceful degradation*” findes der en række transformationsregler, også kaldet for Graceful Degradation (GD) reglerne, som kan bruges på applikationens kilde-interface. Ud fra dette kan man designe interfaces på mere begrænsede platforme.

I [Flo04] beskrives begrænsede platforme som værende:

- Platforme hvis skærm har en lavere opløsning
- Platforme hvis skærm har en lignende opløsning, men hvor interfacets objekter skal være større (touch screen) eller hvor en del af displayet er brugt til andre formål (virtuelt keyboard).
- Platforme hvor færre brugergrænsefladeelementer er tilgængelige, da de eksempelvis har et reduceret toolkit til rådighed.
- Platforme hvor nogle brugergrænsefladeelementer er mindre brugbare, hvis disse platforme eksempelvis ikke har et keyboard.

Målet med GD reglerne er at skabe UI’er til forskellige platforme på baggrund af kilde-interfacet. Der eksisterer to typer GD regler, der kan benyttes på UI’et, hvoraf ”*layout transformationsreglen*” beskæftiger sig med transformation af layout-forholdet mellem grafiske elementer. Den anden type;

”grafisk element transformationsreglen”, beskæftiger sig med modificering af antallet af grafiske elementer, og disses natur.

Layout transformationsreglen:

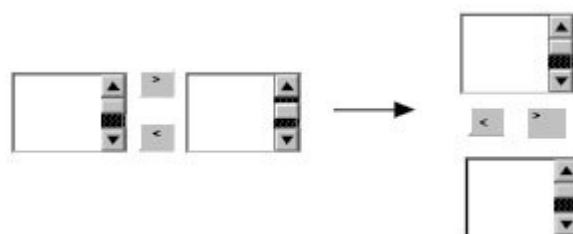
Der gøres ifølge [Flo04] under denne regel yderligere brug af tre regler som anvendes i forhold til layout forholdet. Disse er:

Dimensioneringsreglen: Denne regel modificerer et grafisk elements dimensioner og kan anvendes på hvilket som helst UI element. Der skal dog, ifølge [Flo04], tages højde for følgende aspekter:

- *Det enkelte elements natur:* Nogle elementer såsom et ”radio item” har en fastlåst størrelse, hvorimod en knap generelt godt kan skaleres.
- *Begrænsninger påkrævet af toolkits:* Mange toolkits tillader ikke programmøren at skalere grafiske elementer.
- *Menneskets begrænsede forståelse:* Ifølge [Flo04] har brugbarhedseksperimenter vist, at ikoner ikke kan skrumpes under en tærskel på 8 x 7 pixels. Alt herunder vil være for ulæseligt eller for umuligt at skelne fra andre ikoner.

For at skalere et element er der en række informationer som er nødvendige at kende til. Det er blandt andet nødvendigt at vide, hvor meget et element kan formindskes i forhold til højde og bredde. Denne minimumsgrænse viser sig ofte først når programmet designes, da eksempelvis en knaps minimumsgrænse afhænger af størrelsen på den label, der er på knappen. Minimumsgrænsen for en listboks afhænger på tilsvarende vis af længden af de valgmuligheder der er i listboksen.

Orienteringsreglen: Denne regel modificerer et elements retning uden at ændre ved dets størrelse og position, og benyttes primært når man skifter fra ”landscape mode” til ”portrait mode” eller omvendt. Figur 1 viser et eksempel herpå.



Figur 1 - Orienteringsreglen

Placeringsreglen: Denne regel modificerer positioneringen af et element, og er brugbar i følgende situationer:

- Hvis elementerne ikke passer i den ene dimension (horisontalt eller vertikalt), når der er fri plads tilbage i den anden dimension.
- Hvis elementerne ikke passer horisontalt, og man ønsker at undgå horisontal scrolling.
- Hvis der eksisterer ergonomiske regler eller konventioner på platformen der designes på - bestemte apparater har bestemte måder, hvorved menuerne er placeret.

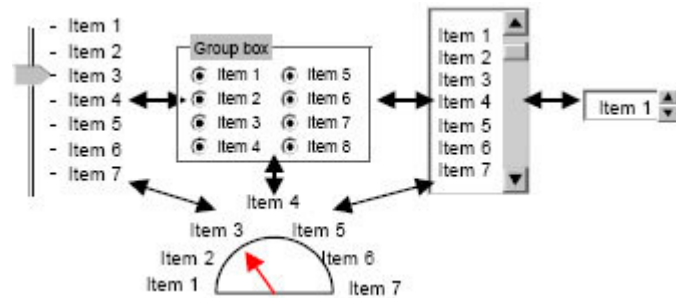
Grafisk element transformationsreglen:

Denne type regel beskæftiger sig, som tidligere nævnt, med modificering af antallet og naturen af grafiske elementer. Denne transformation kan ske ved hjælp af følgende tre regler:

Modifikationsreglen: Denne tager udgangspunkt i de grafiske elements ydre. Forstået på den måde, at den fysiske gengivelse af en semantisk feature kan modificeres. En teksts font eller farven på et element kan eksempelvis ændres.

Substitutionsreglen: Denne regel udskifter et interaktionselement med et alternativt interaktionselement, som muliggør den samme type interaktionsmuligheder. Man bør benytte substitutionsreglen, hvis en af de to følgende situationer opstår:

- *Manglende tilgængelighed:* Hvis interaktionselementet ikke er tilgængeligt på den nye platform, er det nødvendigt at erstatte det med et andet som er tilgængeligt.
- *Utilstrækkelig skærmstørrelse:* Hvis et interaktionselement er for stort til være på den nye platform, er det nødvendigt at erstatte det. Et eksempel herpå kan ses på Figur 2.



Figur 2 - Udskiftning af elementer

Elimineringsreglen: Denne regel er ganske simpel, idet den blot fjerner et grafisk element på grund af pladsbegrænsninger på den nye platform. Eksempelvis kan billeder fjernes fra UI'et på en mobiltelefon.

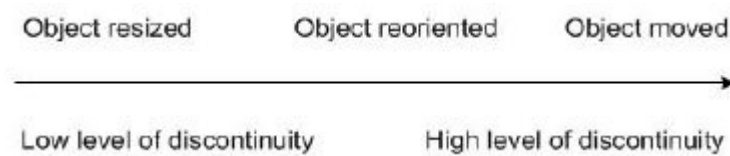
2.5 Kontinuitet

Multiplatform systemer har i modsætning til mange andre systemer en ekstra dimension, idet at de udover at skulle være "brugbare" hver især også bør være sammenhængende på tværs af platforme. Det bør være tydeligt, at en applikation kørende på to forskellige platforme, rent faktisk er den samme applikation, blot i en anden "indpakning". For at imødekomme og forstå denne problematik er der indenfor litteraturen blevet introduceret forskellige begrebsrammer. Vi har i dette afsnit inddraget to af disse begrebsrammer der på hver deres måde forsøger at sikre sammenhængende multiplatform systemer.

2.5.1 Graceful Degradation kontinuitet

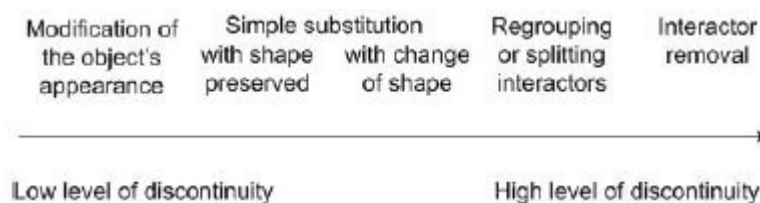
Som vi så tidligere ved GD reglerne, tilpasser man interaktionselementer fra et "rod-interface" til et andet mere begrænset interface. Disse regler sikrer ifølge [Flo04] en vis kontinuitet systemerne imellem. Dog har rækkefølgen hvori disse regler benyttes, en væsentlig betydning for hvor megen kontinuitet der reelt vil eksistere indenfor et multiplatform system. I [Flo04] introduceres derfor en prioriteringsrækkefølge for GD reglerne. Reglerne med en høj prioritet bør afprøves før regler med lavere prioritet tages i brug. Prioriteringsrækkefølgen ser ud som følger:

- *Layout transformationsreglen:* Den regel der forekommer at skabe mindst diskontinuitet er dimensioneringsreglen, derefter orienteringsreglen og til sidst placeringsreglen. Figur 3 viser prioriteringsrækkefølgen.



Figur 3 - Layout transformationsreglen

- *Grafisk element transformationsreglen*: Den simpleste form for modifikation af et interaktionselements udseende skaber kun lidt diskontinuitet (farveændring). Substitutionsreglen, hvor et interaktionselement udskiftes med et andet skaber mere diskontinuitet og mest diskontinuitet skabes, hvis et interaktionselement fjernes. Figur 4 viser prioriteringsrækkefølgen.



Figur 4 - Grafisk element transformationsreglen

2.5.2 Videnskcontinuitet

I [Den03] beskrives kontinuitet eller nærmere betegnet *videnskcontinuitet*, som et aspekt der bør opretholdes når den samme applikation benyttes på forskellige platforme. Hvor kontinuitet ved GD reglerne udelukkende fokuserer på regler inden for manipulering af de grafiske elementer, er der i [Den03] fokus på videnskcontinuitet i forhold til brugernes mentale model. Viden opnås ved brug af applikationen på ét apparat, og denne viden bør generelt være anvendelig ved interaktion med den samme applikation på et andet apparat. I [Nie02] tales der om to former for "konsistens" i brugergrænseflader, som vi ser som værende meget lig den forståelse af begrebet "videnskcontinuitet", som findes i [Den03]:

- *Intra-applikation konsistens*: Det der understøtter brugeren i at opretholde den samme mentale applikationsmodel på forskellige apparater.

- *Intra-apparat konsistens*: Det der hjælper brugeren til at skabe en strategi for, hvordan analogier tilpasses til den apparat-specifikke kontekst.

Hermed ser vi, at konsistens i brugergrænseflader hjælper brugeren til at bringe sin mentale model af én brugergrænseflade, med sig videre til en brugergrænseflade, hvor modellen således stadig er gyldig.

2.6 Vores overvejelser

I [Sef04] gives en anseelig mængde information, som udviklere bør tænke over og inddrage i deres beslutningsproces, når multiplatform systemer udvikles. Vores forståelse af denne information går meget i retning af ”commonsense” information, og kan derfor fortolkes på lige så mange måder som der er udviklere til. De brugbarhedsfaktorer der nævnes, lister en række værdifulde oplysninger, men hvordan disse oplysninger konkret skal bruges, gives der ikke noget svar eller bud på. [Flo04] kritiserer denne tilgang, ved at sige, at man som udvikler har meget lidt kontrol over selve designet af UI’et, da mange af de værktøjer der ligger til grund for en sådan model-baseret udvikling, ikke tillader udvikleren at ændre de interaktionselementer, som værktøjerne tilbyder.

Vi mener, at [Flo04]’s kritik af ”*task-model-based*” udvikling er berettiget. Vi finder det problematisk, at udviklingsværktøjet skal sætte så restriktive begrænsninger for, hvad man som udvikler har mulighed for at ændre på og tilpasse, da det netop kan være udformningen af et grafisk element i UI’et, der er afgørende for dets brugbarhed. Dermed vil senere rettelser, der måtte komme til UI designet som et resultat af brugertests, muligvis ikke kunne implementeres, grundet udviklingsværktøjets restriktioner.

I [Flo04] benyttes en alternativ tilgang, hvor et mere eller mindre ubegrænset UI transformeres til et UI på andre mere begrænsede platforme. I [Flo04] er argumentation herfor, at man på denne måde er sikret konsistens platformene imellem, da alle UI’s er ”nedarvet” fra et overordnet kilde-interface. Vi finder dog denne tilgang til udvikling af multiplatform systemer som værende problematisk, da brugbarhed stort set ikke nævnes. Spørgsmålet er, om [Flo04] ved udarbejdelse af GD reglerne har haft brugbarhed med i overvejelserne. Umiddelbart forekommer det ikke således, da reglerne grundlæggende handler om at skabe den nødvendige plads på mere begrænsede platforme, uden at tage hensyn til hvor ”brugbart” UI’et rent faktisk bliver. Dog ser vi begrebet ”kontinuitet” som et eksempel på, at brugbarhed er implicit indeholdt i brugen af GD-reglerne. I

[Flo04] afhænger målestokken for en succesfuld ”transformation” af, hvor lidt diskontinuitet der eksisterer mellem kilde-interfacet og det nye interface.

Vi mener ikke at kontinuitet nødvendigvis har en positiv indflydelse på brugbarhed. Det at tilsvarende interaktionselementer benyttes på tværs af platformen, er ikke umiddelbart ensbetydende med høj brugbarhed, ligesom det at fjerne et element, ikke nødvendigvis skaber lav brugbarhed. Til trods for dette mener vi dog, at brugen af kontinuitet i [Flo04] nærmer sig den forståelse af *videnskontinuitet*, der benyttes i [anonymous].

[anonymous] forståelse af den mentale model er væsentlig, da den ikke kun fokuserer på manipulering af fysiske elementer som i [Flo04]. Den giver en række abstrakte guidelines, som også har brugbarhedsaspektet indeholdt, samtidig med at brugerens mentale model, eller forståelse, af MUI’et tages med i overvejelserne – det er vigtigt at brugerens viden om brugen af et UI kan overføres til et andet UI i et MUI.

Vi mener, at de i [Flo04] præsenterede redskaber vil være nyttige til at transformere et almindeligt UI til et UI på en begrænset platform, idet tilgangen i modsætning til [Sef04] og [anonymous] giver konkrete arbejdsredskaber til udvikleren. Vi ser forståelsen af begrebet kontinuitet som værende vigtigt for udviklingen af multiplatform systemer, da dette er med til at sikre at brugeren kan relatere to forskellige platforme til hinanden. Samtidig mener vi dog, at det er vigtigt at have brugerens mentale model omkring selve arbejdet i forhold til systemet for øje, således at det sikres, at systemet rent faktisk svarer til brugerens forventninger og behov, da kontinuitet ikke i sig selv garanterer systemets brugbarhed.

3 Contextual Design

Til undersøgelse af falckreddernes arbejde, og det indledende design af vores system, har vi valgt at bruge "Contextual Design". Ud fra [Hol05] vil vi i dette kapitel vil vi først beskrive den metode vi har fulgt, og den proces vi er gået igennem, for at udvikle og designe vores system, for dernæst at diskutere de problemer der kan være i CD i forhold til andre udviklingsmetoder.

3.1 Beskrivelse af Contextual Design

Formålet med CD er først og fremmest at give udviklerne af software en god forståelse af det arbejde, som deres kunder skal lave, og dernæst at understøtte en kreativ udviklingsproces, hvor al implicit viden bliver gjort eksplicit, og deles mellem udviklerne. CD indeholder værktøjer til at lave feltundersøgelser, hvilket foregår på en sådan måde, at udvikleren får et indgående indblik i hvorledes kundens eller brugerens hverdag foregår. Det er en af CD's hjørnesten, at udviklerne skal forstå hvorledes brugeren arbejder, for at kunne lave edb-systemer, som kan understøtte dette arbejde. I Tabel 1 kan ses de værktøjer som bruges i Contextual Design. Vi har valgt at dele værktøjerne op i fire trin, som vi delvist har lånt fra [Ble03], dataindsamling, modellering, konsolidering og design. Disse vil vi beskrive i det følgende.

Dataindsamling	Contextual Inquiry Interviews
	Contextual Inquiry Interpretation Sessions
Modellering	Arbejdsmodellering
	Konsoliderede Sekvensmodeller
Konsolidering	Bygning af Affinity-diagram
	Brug af Kontekstuelle data til at lave Personaer
Design	Gennemgang af Affinity og Konsoliderede Sekvenser
	Visioning
	Storyboarding
	Test med Papirprototyper
	Papirprototype-interviews

Tabel 1 - Trin i Contextual Design

3.1.1 Dataindsamling

Dataindsamling kommer naturligt som det første i arbejdet med Contextual Design.

Dataindsamlingen foregår typisk som en række to-timers undersøgelser, som benævnes ”interviews”.

De guidelines der benyttes, har til formål at give interviewerens et så godt som muligt indblik i den kontekst som vedkommende undersøger, primært ved at observere og diskutere kundens arbejdsgange med kunden. I denne proces er det væsentligt for en succesfuld Inquiry, at interviewer og kunden arbejder sammen om at skabe den rette forståelse af arbejdet. Som [Hol05] siger, vil en kunde ikke kunne beskrive sit eget arbejde, uden at gøre det, da man naturligt udelader detaljer, som ikke virker væsentlige. Omvendt er det heller ikke tilstrækkeligt at observere kundens arbejdsgange, da en observatør ikke naturligt vil kunne forstå bevæggrundene bag de ting som kunden foretager sig. Af denne grund er det vigtigt for et vellykket interview, at interviewer og kunde indgår i et partnerskab, for at skabe den bedst muligt forståelse af det arbejde der foretages. Det er denne filosofi, der ligger til grund for hele CI-processen, og som også er gældende for interaktion med kunden i senere faser.

Udover de etnografiske undersøgelser, observationer og interviews indeholder CI værktøjer til at dele de erfaringer som udvikleren har gjort sig, under det enkelte interview. Det vigtigste kaldes for ”Interpretation Sessions” - fortolkningssessioner, hvor interviewer fortæller holdet om vedkommendes oplevelser, og der tages noter herom. Dette inkluderer almene observationer om arbejdsgangen, spørgsmål, og ideer til designet. Disse noter bruges senere i processen. I slutningen af den enkelte session indfanges arbejdsmodeller og ”Insights”, som er teamets tanker omkring interviewet, og den situation det beskriver.

3.1.2 Modellering

Efter at have færdiggjort fortolkningsprocessen bruges Arbejdsmodellering. Dette sker ved sekvensmodeller, som viser sekvensen i arbejdsgangene, artefaktmodeller, som indfanger de objekter, som bruges i arbejdet, og fysiske modeller, som viser hvorledes de fysiske omgivelser påvirker arbejdet. Sekvensmodellerne kan desuden samles under en Konsolideret Sekvensmodel, som samler kundens væsentlige kerneopgaver, såfremt der er behov for at konkretisere viden om dennes arbejdsgange.

3.1.3 Konsolidering

Et "Affinity-diagram" er det værktøj som [Hol05] bruger til at sammensætte de noter, som blev skabt under fortolkningssessionerne. Dette foregår ved at alle noter skrives ned på gule post-it notes, og grupperes i et stort diagram, hvorved centrale temaer for det edb-system man skal udvikle gerne skulle fremkomme. Affinity-diagrammets hovedformål er, at gruppere de enkelte observationer, således at udviklerne kan se, hvor der er områder i kundens arbejde, som det er relevant at tænke ind i designet af et edb-system. Noterne grupperes således under overskrifter, som har gerne skulle have en form for direkte relevans for det system man skal udvikle.

Bygningen af diagrammet foregår i flere trin, således at der først bygges kolonner af gule notes, som har en form for sammenhæng. Dernæst tilføjes blå notes, hvorpå denne sammenhæng noteres, og de gule notes gennemgås igen, for at se om der stadig er en fornuftig sammenhæng – ellers flyttes de. Når disse kategorier findes tilfredsstillende tilføjes lyserøde notes, som har til formål at vise nøgle-emner i de data som de underliggende blå notes repræsenterer. Endelig kategoriseres de lyserøde notes med grønne notes.

Efter at Affinity-diagrammet er lavet, kan der konstrueres "Personaer". Dette er en form for brugerprofiler, som er repræsentative for en gruppe brugere af det system, som man er ved at udvikle. Den enkelte Persona skal indeholde et brugs-scenarie, som viser et eksempel på en typisk arbejdsopgave, og desuden en liste af mål, opgaver og roller, som denne type bruger typisk har.

3.1.4 Design

Når de indsamlede data er konsolideret, i form af sekvensmodeller og affinity-diagram, kan man gå i gang med at skabe design-ideer. I CD foregår dette ved at udviklerne betragter de frembragte data, og tilføjer notes med ideer, der hvor de ser relevans for et design. På denne måde gøres det klart for andre, hvorfor en ide opstod, og hvilke notes der lå til grund for denne ide. På samme måde kan man, hvis man ser et "hul" i de data der udgør modellen eller affinity'en, skrive en note, og gøre opmærksom på dette. CD tillader så, at der indsamles yderligere data, som tilføjes til den relevante model. Denne gennemgang af de indsamlede og behandlede data giver et grundlag for alle, som skal deltage i det videre arbejde.

Når alle deltagere i projektet har gennemgået de tilgængelige data, kan der skabes "Visions". Dette er en proces som kombinerer "grounded brainstorming" og storytelling [Hol05, s. 213], og resultatet

er billeder, som viser arbejdsgange med det system man vil udvikle, samtidig med at ideer til brugergrænseflader og funktioner føjes til. Der udvikles en række visioner, hvis indhold deles mellem alle medlemmer af teamet, hvorefter de kerneelementer der er indeholdt i disse visioner, konsolideres i en endelig vision. Hvis der er antagelser eller uopklarede spørgsmål i delvisionerne, bør de så vidt muligt afklares først.

Storyboards er det sidste skridt i designprocessen, inden det reelle interface skal fremstilles. I dette led i processen vises det, hvordan en bruger vil bruge systemet, samt præcis hvilke funktioner brugeren vil komme til at have brug for, for at kunne udføre sit arbejde. Hvert storyboard fortæller en historie, ud fra en af de sekvenser der blev lavet tidligere i CD-processen.

Det sidste skridt i CD-processen er at udvikle og evaluere en papirprototype. Dette foregår ved at der itereres over et mockup af et interface, lavet på eller af papir. Hvis man har lavet storyboarding, er det let at tage skridtet til mock-ups, da man allerede har identificeret elementer til brugergrænsefladen, og blot mangler at formgive og placere disse. Også her kan det være en fordel at benytte post-it notes, da elementerne så let kan omrokers.

Endelig kan denne prototype fremvises for potentielle brugere. Ved at gennemgå prototypen med disse, kan man verificere, at udviklernes forståelse af arbejdet har været korrekt, og at det system man udvikler faktisk vil kunne understøtte kundens arbejde. Post-it note mock-up'en har de samme fordele som en konventionel papirprototype. Den er uformel og indbyder i endnu højere grad til diskussion, da brugergrænsefladens elementer kan omrokers efter forgodtbefindende.

3.2 Brug af Contextual Design

Da Contextual Design er en række værktøjer, hvor [Hol05] selv har valgt at dele processen op i tre mulige forløb, har vi valgt at benytte de dele af CD, som vi ser som gunstige for udviklingen, mens vi har fravalgt andre. De dele som vi har brugt har vi i visse tilfælde modificeret, så de passer vores omstændigheder bedst muligt.

Aktiviteter	Lightning Fast	Lightning Fast +	Focused Rapid CD	Vores valg
Contextual Inquiry Interviews	X	X	X	X
Contextual Inquiry Interpretation Sessions	X	X	X	X
Arbejdsmodellering			X	
Konsoliderede Sekvensmodeller			X	
Bygning af Affinity-diagram	X	X	X	X
Brug af Kontekstuelle data til at lave Personaer			X	
Gennemgang af Affinity og Konsoliderede Sekvenser	X	X	X	X
Visioning	X	X	X	X
Storyboarding			X	
Test med Papirprototyper		X	X	X
Papirprototype-interviews		X	X	X

Tabel 2 - Typer af Contextual Design

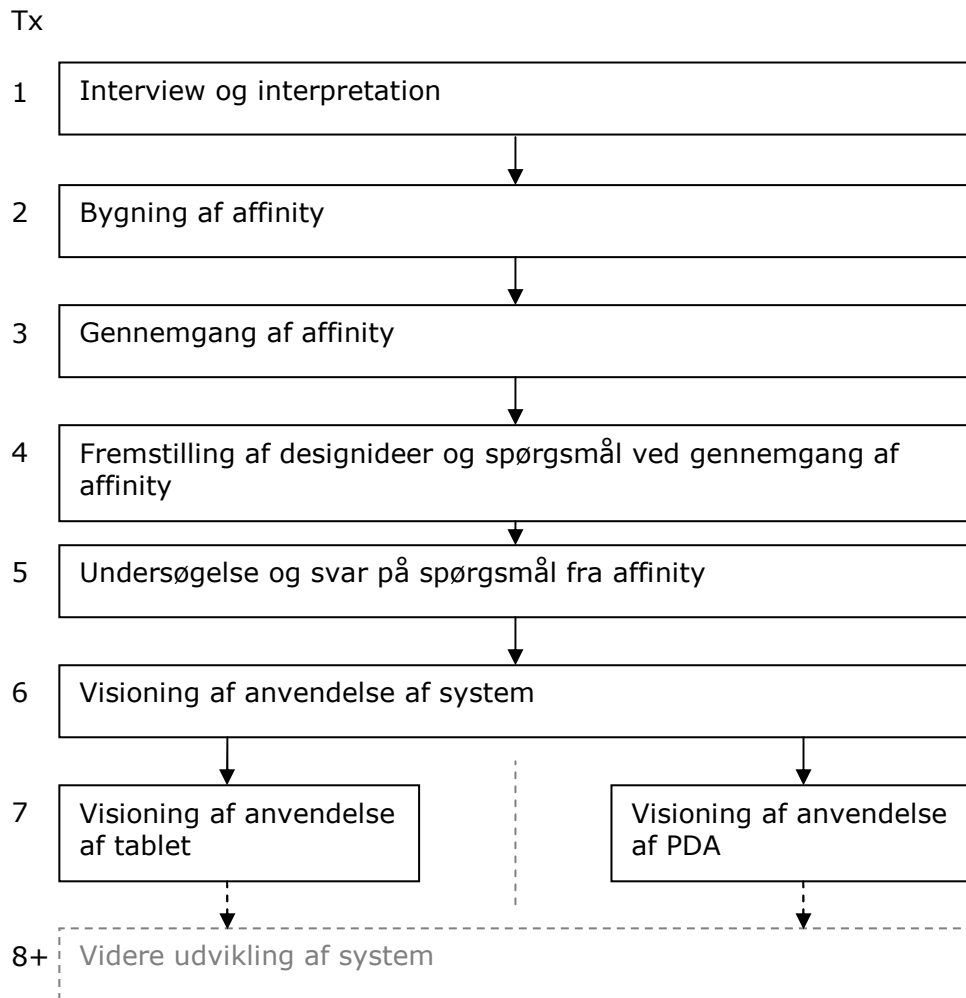
I Tabel 2 kan ses de typer som [Hol05] har delt Contextual Design op i, sammen med den tilgang som vi har valgt. "Focused Rapid CD" er den grundlæggende type, men også den mest ressourcekrævende. Som det kan ses, har vi benyttet det samme udvalg af værktøjer, som Holtzblatt et al. betegner som "Lightning Fast +".

Dette betyder for os, at vi fravælger at lave modellering, ligesom vi fravælger at beskrive de typiske brugere og brugsmønstre i "Personaer" og "Storyboards".

Typen har, udover udvalget af værktøjer, også betydning for hvor mange bruger-interviews man baserer sit design på, [Hol05] foreslår 8-12 interviews i "Focused Rapid CD", mens der i "Lightning Fast +" foreslås 6-12. Typen har altså betydning, ikke bare for hvilke værktøjer man bruger, men også hvor megen tid man bruger på den enkelte aktivitet.

3.2.1 Contextual Inquiry Interviews

For os er Inquiry-delen den mest interessante del af CD i dette projekt, da vi ikke har et behov for at dele vores forståelse med andre udviklere. Dette betyder også, at vi har valgt at bruge en del tid på denne, således at vores forståelse af falckreddernes arbejde bliver så god som muligt.



Figur 5 - Forløb i Contextual Design

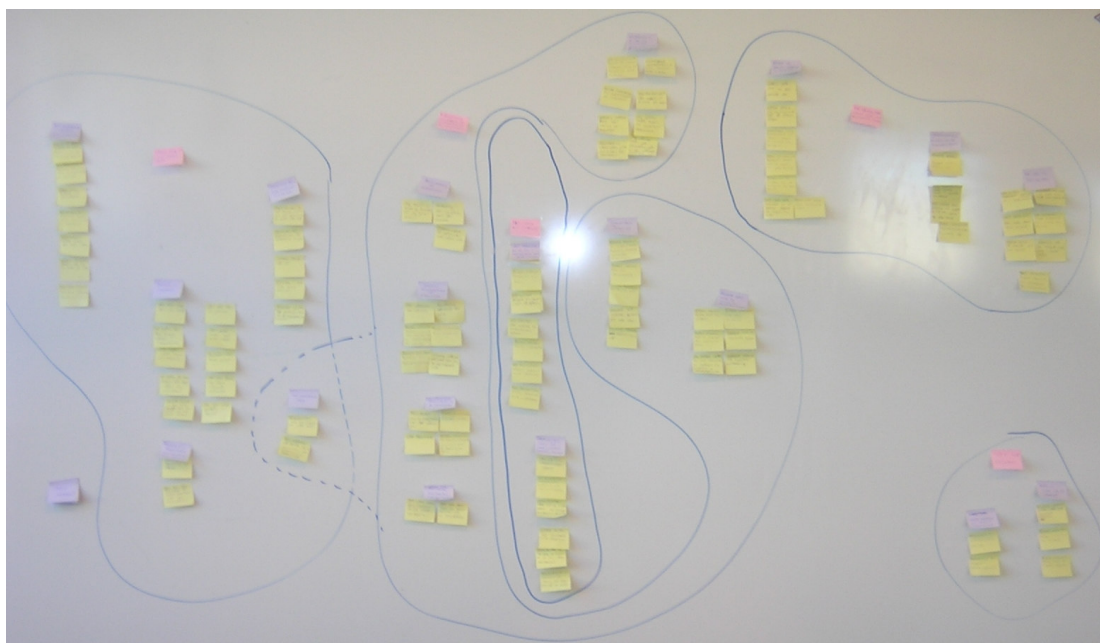
Vi planlagde vores Contextual Inquiry således, at vi så vidt muligt skulle have mulighed for at observere falckredderne i arbejde. Dette betød, at det ikke var praktisk muligt at følge de retningslinjer, som [Hol05] havde fremsat. Dette betød, at vi planlagde to ”interviews” á otte timer, hvor vi fulgte redderne, og observerede og diskuterede deres arbejde med dem. I sagens natur arbejder falckreddere ikke hele tiden, og der var derfor også en del ventetid, hvor der reelt ikke var noget at observere. Arbejdets uforudsigelige natur betød da også, at vi endte med at lave et ekstra interview, da et af de planlagte viste sig at være på en dag hvor der ikke foregik meget ”interessant” – falckreddere kan kun udføre deres arbejde hvis der er nogen der kommer til skade eller bliver syge.

Efter hvert interview lavede vi ”Interpretation Sessions” hvor den der havde været på besøg hos Falck, fortalte om sine oplevelser, mens den anden skrev noter omkring de observationer der fremkom.

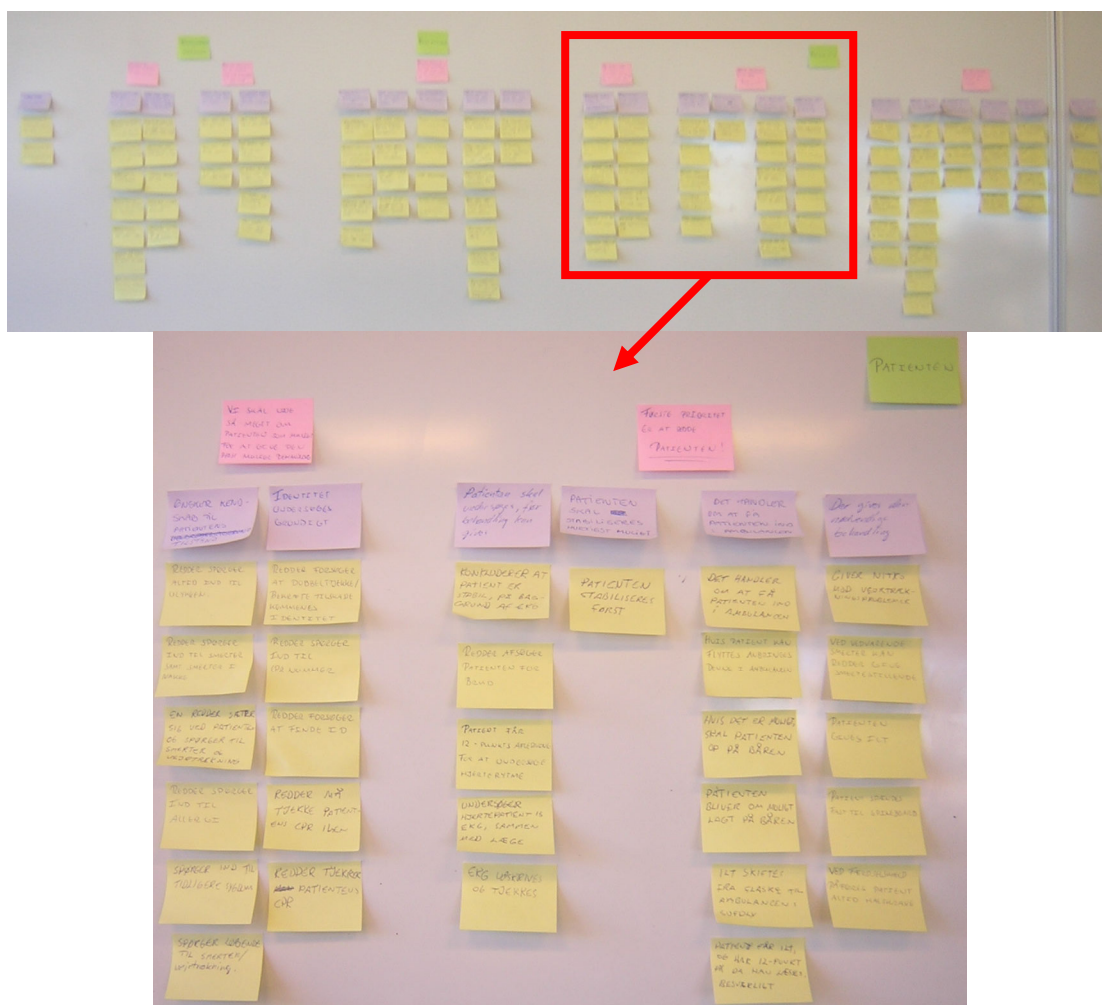
3.2.2 Bygning af Affinity-diagram

Vi fravalgte store dele af CD-modelleringsprocessen, da de fordele som disse aktiviteter skaber, primært er rettet mod at præsentere og dele viden med udenforstående udviklere. I stedet valgte vi at bringe udenforstående ind, for at arbejde med ”Affinity-diagrammet”, og lade dem diskutere og analysere dette. Ved at præsentere disse udenforstående for vores data, har det været vores forventning at få en række design-ideer til det næste skridt i processen. I dette trin brugte vi 3 timer med tre andre speciale-studerende, som først fik en udførlig præsentation af casen for vores projekt, hvorefter de gennemgik de noter vi havde lavet, og grupperede disse i et ”Affinity-diagram”(Se Figur 6), ifølge de retningslinier der er foreskrevet i [Hol05]. Vi var begge til stede under hele sessionen, for at kunne komme med forklaringer på forkortelser, fagudtryk og eventuelle tvetydige noter.

Da alle noter var grupperet, og der var fundet overskrifter til de enkelte grupper, samt overoverskrifter til yderligere gruppering, blev alle noter reorganiseret, således at beslægtede grupper blev placeret sammen, og affinity-diagrammet dermed blev mere overskueligt (Se Figur 7).

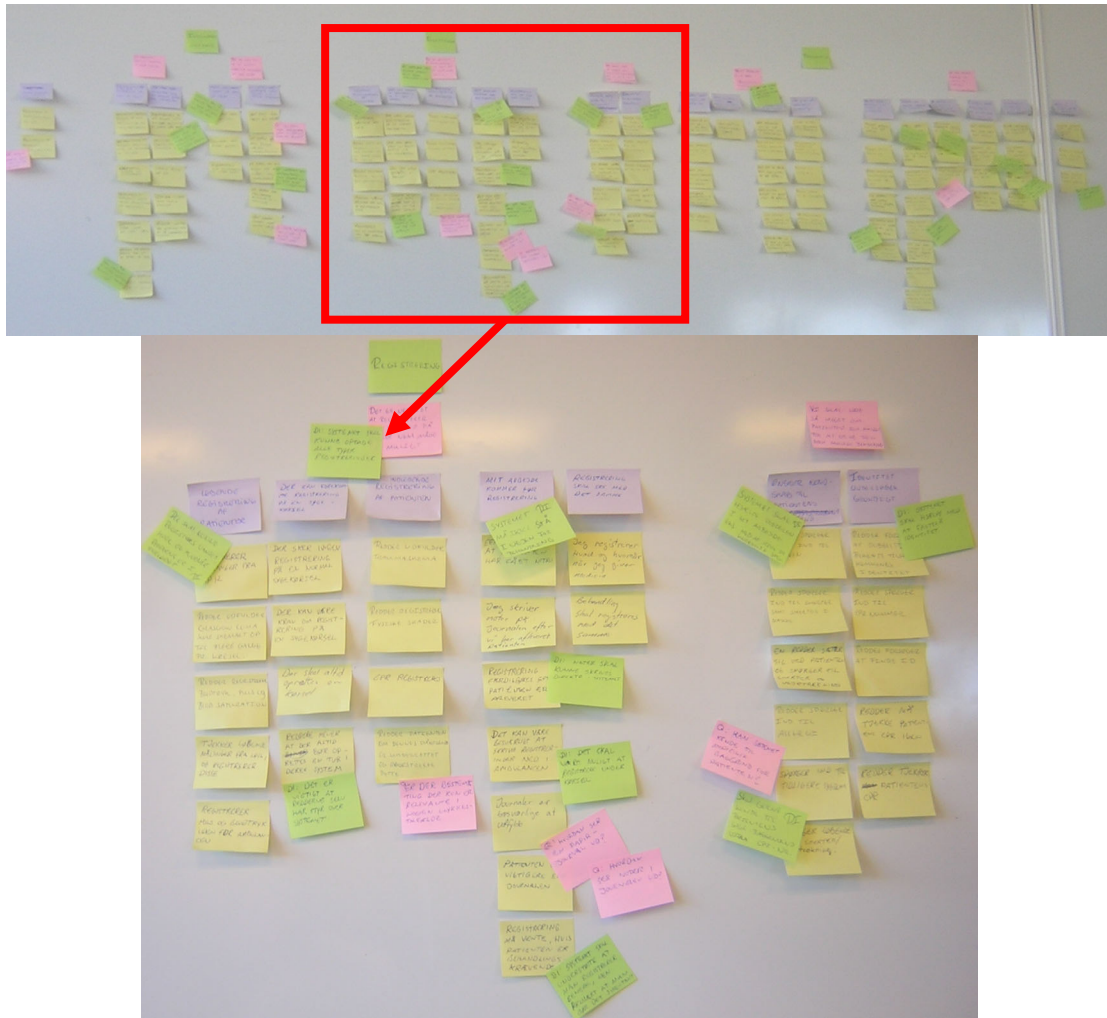


Figur 6 - Affinity, første sortering



Figur 7 - Affinity, sorteret i grupper, med overskrifter

Som afslutning på sessionen gennemgik vi affinity-diagrammet, og diskuterede hvordan et system til at understøtte det arbejde som falcckredderne har, kunne se ud. Ved at gennemgå diagrammet minutiøst, var det muligt at se, hvor der var muligheder eller huller, i de data vi havde indsamlet. Ud fra dette blev der placeret yderligere noter på diagrammet, som indikerede ideer til design, eller spørgsmål som var nødvendige at få afklaret, før et system kunne konstrueres (Se Figur 8).



Figur 8 - Affinity med spørgsmål og design-ideer tilføjet

De endelige grupperinger af affinity-noter kan ses i bilag A.

Tabel 3 viser de spørgsmål som opstod i forbindelse med fremstilling af Affinity-diagrammet. For at kunne være sikre på, at vores forståelse af falckreddernes arbejde var tilstrækkelig, var det nødvendigt, at disse spørgsmål blev besvaret til vores tilfredsstillelse, inden udviklingen af et system kom i gang.

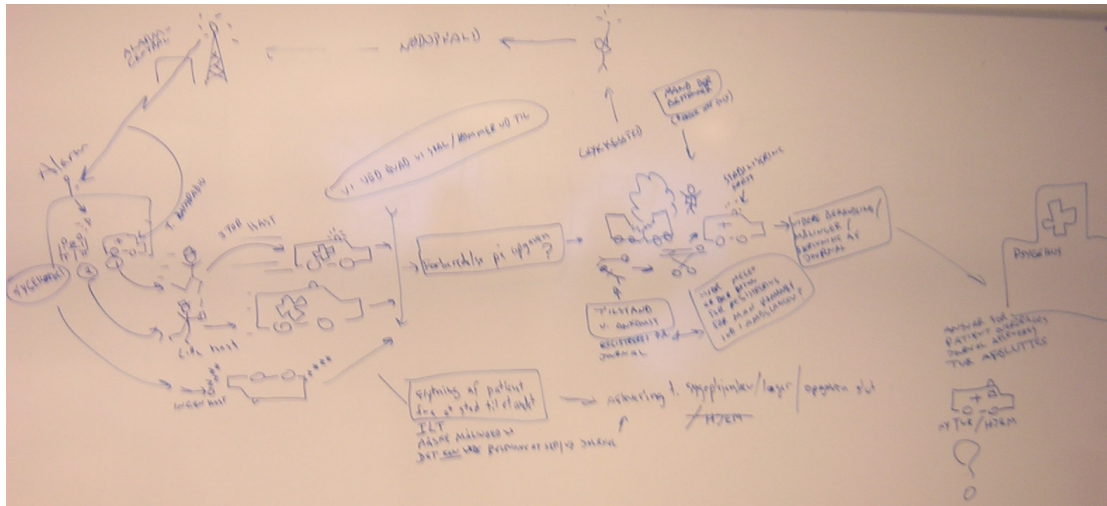
Problem	Løsning
Hvordan kan der deles op efter opgavetype? Modulopbygning?	D
Er der bestemte ting der kun er relevante i nogen ulykkestilfælde	J
Hvordan ser en papirjournal ud?	J
Hvordan ser noter i journalen ud?	J
Hvilke muligheder har vi for at snakke sammen med andet udstyr?	-
Hvilken type hardware skal vi bruge? Hvad er muligt?	D
Hvilke numre har reddere? Hvordan har disse relevans?	R
Kan vi få numre fra dataradioen, eller skal de indtastes manuelt?	-
Hvad er en kørsel 2 i forhold til en sygekørsel / kørsel 1?	R
Kan systemet kende til patientens medicinske baggrund?	J
Må alle falckreddere det samme?	R

Tabel 3 - Spørgsmålsnotes i Affinity-diagram

Nogle af spørgsmålene var designmæssige (D), og som sådan ikke umiddelbart lette at besvare, mens andre blot krævede en snak med en falckredder (R). Endnu andre spørgsmål krævede hjælp fra vores samarbejdspartner, Judex (J), hvor vi i det ene tilfælde måtte lave et større efterforskningsarbejde i Judex' database, for at få den information vi havde brug for. Dataudtrækket og arbejdet med at skabe et context-aware system kan ses i bilag C og appendix A. Desuden var der to spørgsmål som ville kræve indsigt i ambulancens nuværende udstyr, hvilket vi valgte at afgrænse os fra at undersøge, da det lå uden for formålet med dette projekt (-).

3.2.3 Visioning

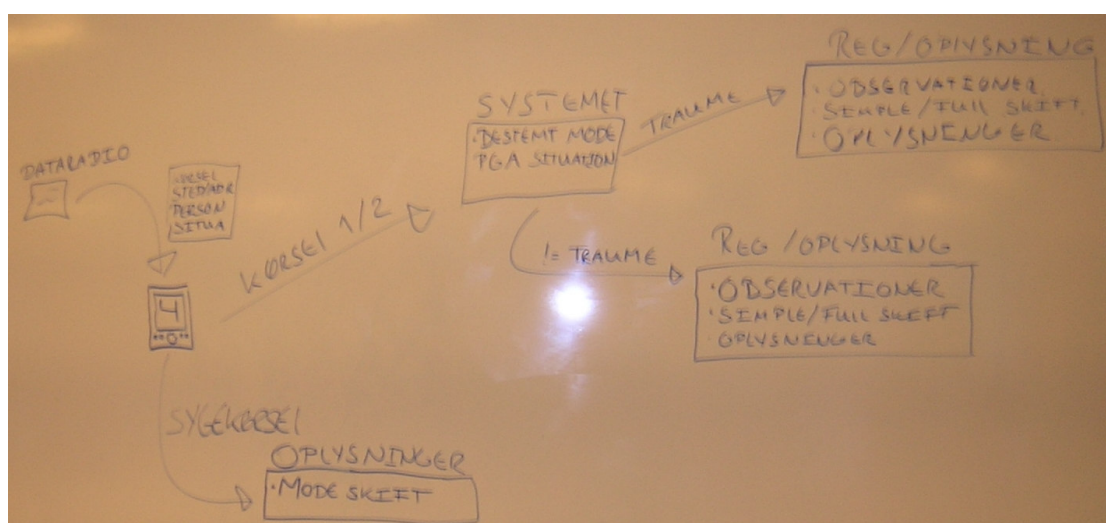
Ved hjælp af den indsigt og de ideer vi fik, ved at fremstille vores Affinity-diagram, og ved at undersøge de områder, hvor vores indsigt var utilstrækkelig, lavede vi to visioner, for at klargøre og afbilde reddernes arbejdsområde, og den måde hvorpå vi kunne skabe et system, som kunne assistere reddernes arbejde.



Figur 9 - Vision over redderes arbejde

Vi lavede tre visioner, som hver havde til formål at afdække en del af falckreddernes arbejde. Den første, (Se Figur 9) er en gennemgang af hvad der foregår, når redderne får en kørsel. Visionen viser grundlæggende, at redderne får en kørsel fra alarmcentralen, og forholder sig forskelligt til denne, alt efter kørselens type. Visionen viser desuden hvilke arbejdsopgaver redderne kan have ved et færdselsuheld, hvordan de behandler en patient, registrerer denne behandling, og bringer patienten til et hospital, hvorefter de melder fri, og er klar til en ny tur.

Vision nr. 2 (Se Figur 10) viser hvorledes vi forestillede os, at data kommer fra Falcks EVA-system til et mobilt system, hvori redderne kan indtaste deres observationer og handlinger i forbindelse med deres patient, frem for at skulle benytte en papir-journal.



Figur 10 - Vision over registrering med mobilt system

Tanken i denne vision var desuden, at der er en grundlæggende forskel mellem ulykker med traume (eksempelvis færdselsuheld), andre udrykninger (eksempelvis hjerteanfald eller kritisk sygdom), og sygekørsler. Vores arbejde med denne mulighed kan ses i appendix A.

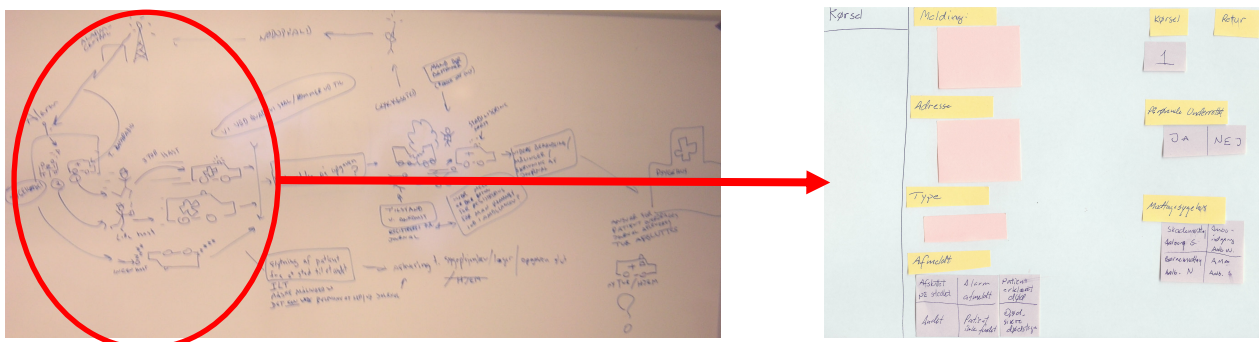
3.2.4 Mock-ups

Da vi igennem affinity og visioning havde skabt os en forståelse for, hvilke funktioner vores system skulle have, lavede vi mock-ups over, hvordan disse funktioner kunne mappes over på knapper i en brugergrænseflade.

Af hensyn til forståeligheden af disse mock-ups, lavede vi følgende konventioner for elementer i mock-up brugergrænsefladen. Da vores beholdning af post-it notes var svingende, er det dog ikke alle mock-ups der har fulgt disse konventioner.

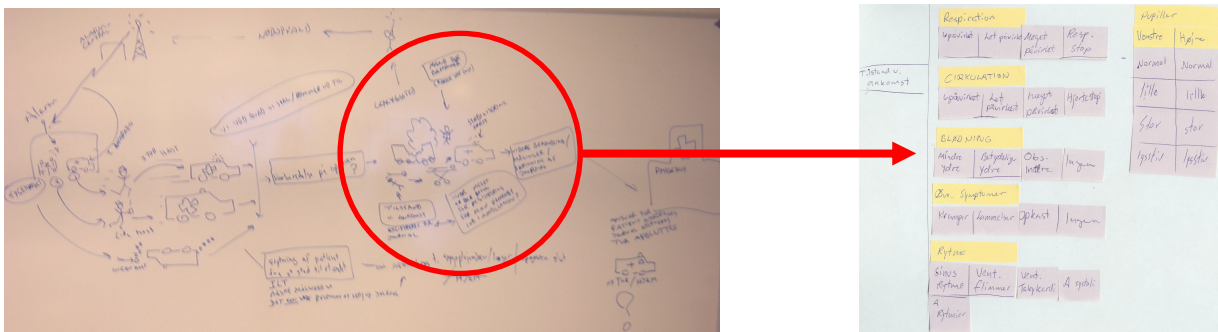
1. Blå – knapper som brugeren kan trykke på
2. Gul – labels eller overskrifter, som beskriver andre elementer
3. Grøn – felter, der indeholder de data som brugeren indtaster eller vælger
4. Lyserød – felter, der indeholder data som systemet har genereret på baggrund af indtastede oplysninger

Udover post-it notes blev der benyttet en almindelig kuglepen. Det hele blev lavet på almindelige hvide A4-ark. Figur 11 viser hvorledes en del af en vision kunne bruges direkte, til at lave en mock-up. I dette tilfælde kunne vi ud fra visionen se, at melding og kørselsinformation kommer fra alarmcentralen, direkte til ambulancen. Dette har vi valgt at samle under ét skærbillede, således at redderne på en overskuelig måde kan se disse informationer om deres opgave.



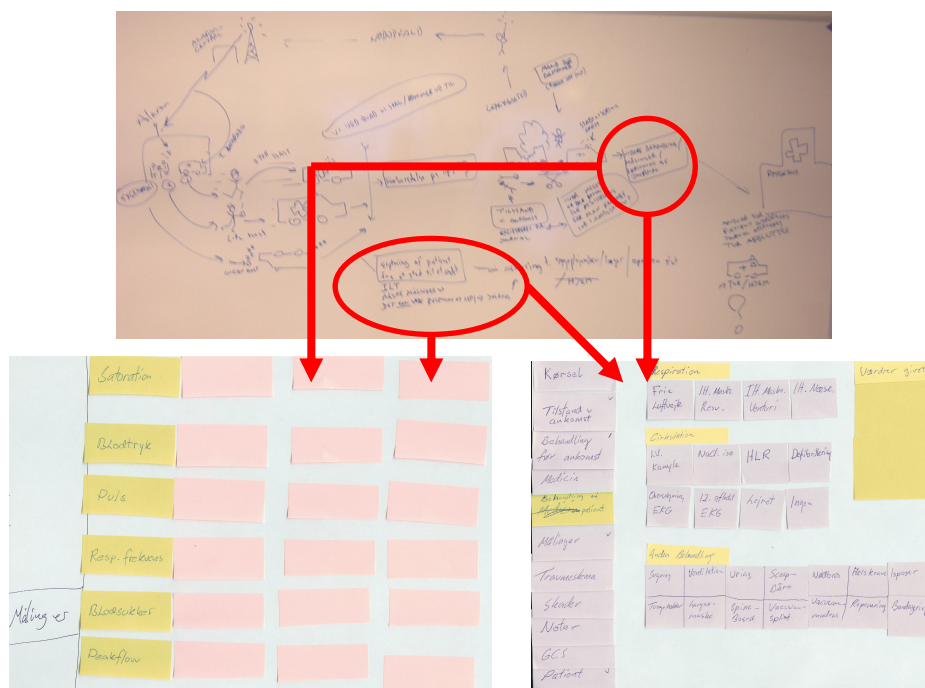
Figur 11 - Vision → mock-up, kørsel

På samme måde har vi set, at der er visse ting, der foregår når redderne ankommer til et skadested. Disse ting er således også samlet under ét skærbillede, hvilket kan ses på Figur 12.



Figur 12 - vision → mock-up, tilstand ved ankomst

Behandlingen af en patient er en mere kompliceret del af systemet, end de to foregående eksempler. Da en patient kan være skadet på mange forskellige måder, kan dette ikke blot samles under et enkelt skærbillede. Ud fra vores vision så vi, at der var brug for to grundlæggende ting, ”målinger” og ”behandling af patient”. Disse to funktioner er tæt forbundet, men vi så det ikke som værende praktisk at samle den under et, da især ”behandling” indeholder en stor mængde valgmuligheder. Samtidig vidste vi, at redderne løbende tager en række målinger, som de baserer deres evaluering af patientens tilstand efter. Dette betød, at vi måtte skabe en måde, som ville gøre det nemt for redderne at følge med i udviklingen i patientens tilstand.



Figur 13 – vision → mock-up, behandling og målinger

3.3 Vores erfaringer med Contextual Design

Selve CD-processen er tydeligvis meget gennemtænkt af [Hol05]. Således var der en klar tråd igennem aktiviteterne, således at interpretation var direkte afhængig af interview, og resulterede i råmaterialerne til affinity-diagrammet, og det færdige affinity-diagram, med spørgsmål og design-ideer lagde ligeledes direkte op til skabelsen af visioner for systemet. Vi fravalgte at lave Storyboarding, og dette har muligvis været betydende for, at vi ikke følte den samme kontinuitet mellem vision og mock-ups, som vi ellers følte i resten af forløbet. Visionerne gav os en klar forståelse af hvad systemet skulle kunne, men skridtet fra konceptuelle ideer omkring funktioner til brugergrænseflade var stort. Vi var således på et tidspunkt i tvivl om, om CD reelt havde nogen betydning for det færdige produkt, da vi i princippet kunne tage den papir-journal, som falckredderne normalt bruger til at notere data om en patient, og lave denne om til et computer-system. Som man kan se i afsnit 3.2.4, var dette dog ikke tilfældet, da visionerne fra CD i nogen grad lå direkte til grund for mock-ups. Vigtigere var dog nok den forståelse af arbejdsgangen, som vi havde skabt os, igennem arbejdet med CD. Denne betød, at vi ofte kunne lave kvalificerede beslutninger angående en hensigtsmæssig opbygning af brugergrænsefladen, frem for at gætte.

Det var fra starten klart for os, at der var visse dele af CD der var mindre relevante for os. Vores udviklingsteam bestod af to mand, og CD er tydeligvis ikke gearret til så små enheder, da mange af dens værktøjer eksplicit fokuserer på at der skal være 4 eller flere personer til stede. Dette ”problem” mærkede vi da også igennem hele processen, hvor vi i et enkelt tilfælde valgte at bringe andre studerende ind i processen, blot for at have det nødvendige antal deltagere til en aktivitet. CD er, i vores øjne, primært til for at dele viden og ideer mellem udviklere, igennem interpretation sessions, affinity et cetera. Vi oplevede af denne grund også en del af de aktiviteter vi foretog os, som delvist overflødige, da der ikke var nogen medarbejdere, som vi kunne dele den skabte viden med. Vi oplevede dog også, at den grundige gennemgang af indsamlede data, og eksplicitte behandling af viden, betød at misforståelser og huller i vores viden blev blotlagt på et tidligt tidspunkt, hvilket var en stor fordel senere i udviklingsprocessen. Endelig oplevede vi, at vi ved at have flere deltagere i konstruktionen af Affinity-diagrammet, fik væsentligt flere kreative bud på hvorledes et sådant system kunne fungere.

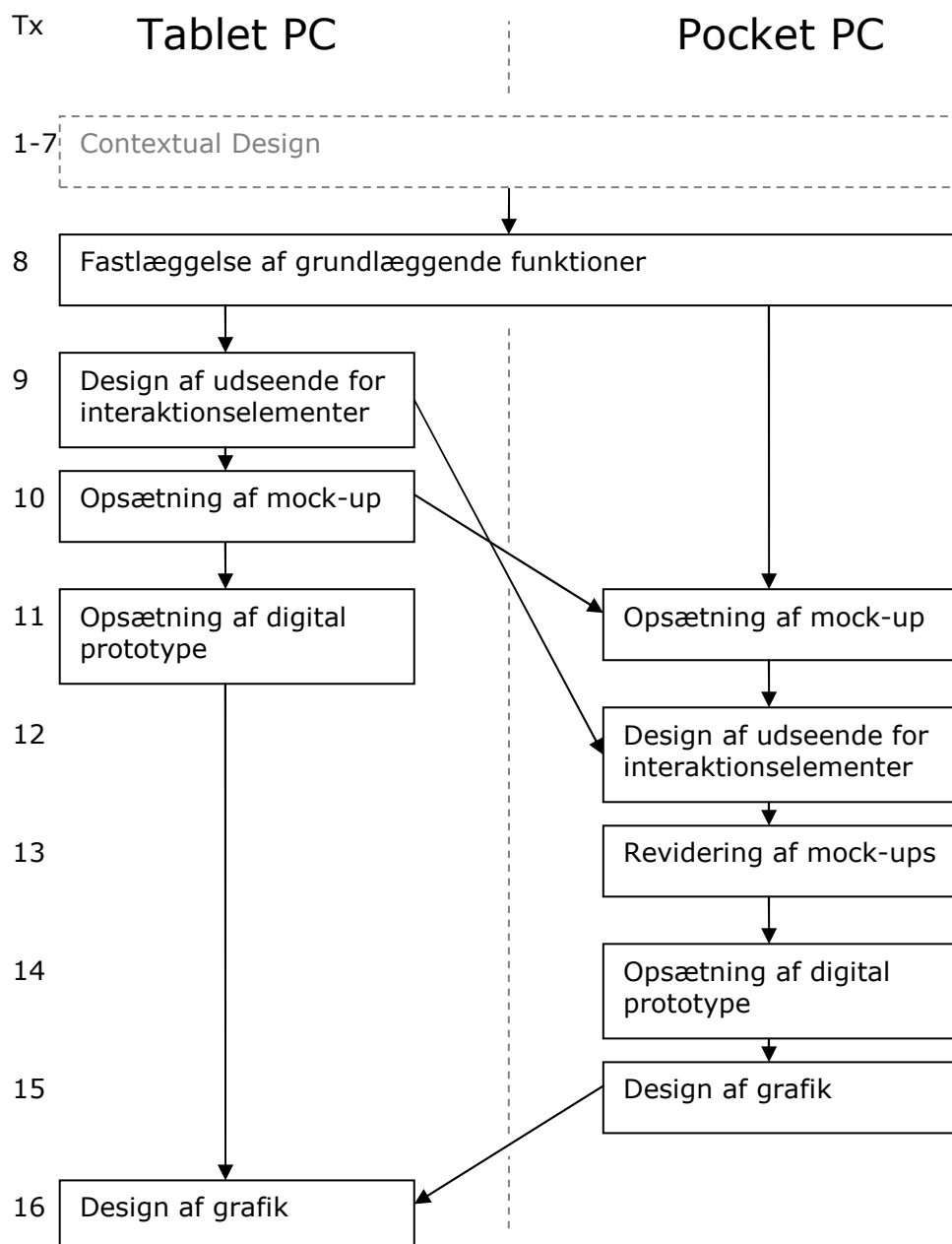
Alt i alt mener vi at Contextual design har været en givende og interessant metode at arbejde med, men mener også, at den vil egne sig bedre til udvikling af koncepter for lidt større projekter, hvor mulighederne for at skabe kreativitet og vidensdeling vil komme langt bedre til sin ret. Det er

naturligvis begrænset hvor mange der kan deltage i det kreative arbejde med CD, men ting som arbejdsmodeller, personaer, visioner og storyboards kan i princippet spredes til en meget stor medarbejderstab, og på den måde være med til at guide forståelsen for den enkelte udvikler, i en arbitrært stor organisation.

4 Udvikling af programmet

I dette afsnit præsenterer vi udviklingen af vores system. Systemet består grundlæggende af to separate enheder, dels et program til en Tablet PC, som er tænkt til stationær brug, og dels et program til Pocket PC. Vores system blev navngivet ”Ajour”, som en sammentrækning af ordet ”Ambulancejournal”.

Som det kan ses i Figur 14, har vi i processen foretaget nogle spring fra den ene platform til den anden. Vi her forklare hvad vi har gjort i de enkelte trin, og diskutere bevæggrundene for de valg vi har foretaget. For at illustrere opdelingen i aktiviteterne under udviklingen, har vi valgt at opdele Figur 14 således, at Tablet PC’en er til venstre, og Pocket PC’en aktiviteter er til højre. Desuden er aktiviteterne delt op efter tidsindex (Tx), således, at aktiviteter med separate kasser, men med samme tidsindex har foregået i to parallelle forløb, mens aktiviteter der deler en kasse på tværs af platformene, har foregået samlet for begge platforme.



Figur 14 - Forløb i udvikling af programmer

8. Fastlæggelse af grundlæggende funktioner

Den grundlæggende funktionalitet blev primært fastlagt ud fra den vision for systemet, som vi havde skabt. Desuden benyttede vi de ting vi kunne se i Judex' database, og en kopi af den journal, som Falck-reddere udfylder i dag, når en kørsel kræver registrering af data (se bilag B). Dette havde den betydning, at Ajour grundlæggende ville blive til en elektronisk version af papir-journalen. Vi så dog nogle muligheder i det elektroniske medie, som en papirjournal aldrig ville kunne tilbyde. En klar mulighed ville være, at tilføre systemet informationer udefra.

Dette kunne ske direkte fra ambulancens dataradio, således at kørselsinformationer kunne benyttes. Dette ville være en forholdsvis begrænset funktionalitet, som ville gøre kørselstype og melding tilgængelig i systemet, samt eventuelle telefonnumre og adresser.

En anden mulighed ville være, at forbinde systemet direkte til telemedicinsk netværk. Vi har observeret, at man på Judex, igennem en VPN-forbindelse, er i stand til at finde data om den enkelte patient, ud fra patientens CPR-nummer.

9. Design af udseende for interaktionselementer (Tablet PC)

Designet af interaktionselementer var et ganske kort trin i udviklingsprocessen. Vi besluttede her, så vidt muligt at følge MS Windows standarden [msdn], for på denne måde at støtte brugerens intuition omkring interaktionselementernes udseende. I Tabel 4 kan ses de krav som Windows standarden stiller til brugergrænseflader.

5.1	Support standard system size, color, font, and input settings
5.2	Ensure compatibility with the High Contrast option
5.3	Provide documented keyboard access to all features
5.4	Provide notification of the keyboard focus location
5.5	Do not rely exclusively on sound
5.6	Do not place shortcuts to documents help, or uninstall in the Start Menu
5.7	Support multiple monitors

Tabel 4 - Krav til Brugergrænseflade under Windows standarden

Standarderne omkring selve interaktionselementerne følges automatisk, når man benytter standard-elementer fra Visual Studio, hvilket vil sige at vi opfylder punkterne 5.1 og 5.2. De resterende punkter er ikke relevante for Ajour, da brugen af systemet ikke foregår ved hjælp af keyboard, der ingen lyd bruges, og det generelt ikke er meningen, at brugeren skal have adgang til det underliggende operativsystem. Hvis systemet skulle bruges på en almindelig PC, ville alle syv punkter dog være opfyldt.

Vi besluttede, at vi så vidt muligt ville spejle alle tekster fra papirjournalen over i Ajour, og ikke benytte ikoner. Dette betyder, at vi har fravalgt at bruge grafiske mnemer til at støtte brugernes hukommelse, men i stedet har satset på tekstuel klarhed, begrundet i, at det er essentielt at der

ikke er tvetydighed omkring knappernes funktioner – specielt set i lyset af, at det allerede på dette tidspunkt var klart, at systemet ville blive overvejende baseret på knapper.

Desuden besluttede vi at overveje nøje, om interaktionselementernes visuelle struktur var fornuftig, og om de var grupperet rigtigt i forhold til de såkaldte ”gestaltlove” [Mol00].

10. Opsætning af mock-up (Tablet PC)

I dette trin påbegyndte vi at opbygge selve brugergrænsefladen til Tablet PC’en. Dette gjorde vi ved hjælp af post-it notes, således at et element, når det først var lavet, kunne flyttes rundt på ”skærmen”, indtil den korrekte placering var fundet. Således blev alle væsentlige skærbilleder fremstillet på papir, inden vi gik videre til næste trin. Vi valgte at udskyde fremstillingen af mock-ups til Pocket PC’en, da vores fokus i dette trin var på at få samtlige identificerede funktioner omdannet til elementer i brugergrænsefladen. Vi indså desuden, at vi med det større interface på Tablet PC’en, nemmere kunne flytte rundt på elementerne, og skabe den mest hensigtsmæssige opbygning, både i form af grupperinger eller gestalter, og i det, at knappernes rækkefølge skulle forekomme naturlig.

11. Opsætning af program (Tablet PC) / opsætning af mock-up (Pocket PC)

I dette trin opdelte vi udviklingen. Dels gik vi i gang med at implementere de allerede fremstillede mock-ups i MS Visual Studio 2003 / C#, dels påbegyndte vi at fremstille de mock-ups der skulle bruges til Pocket PC’en, ud fra de mock-ups der var lavet til Tablet PC’en.

Ud fra de mock-ups, der var lavet til Tablet PC’en, fremstillede vi de tilsvarende mock-ups til Pocket PC’en. Dette arbejde bestod primært i at bestemme, hvorledes hele sider med knapper kunne opdeles, således at sammenhængen mellem enkelte knapper blev bevaret, og at det var klart for brugeren, at der var yderligere funktioner at finde på undersider.

Interaktionselementernes rækkefølge var allerede besluttet i trin 10, så denne var ikke diskutabel på dette trin. Vi havde desuden på dette tidspunkt besluttet os for, at ville benytte Graceful Degradation reglerne til udvikling af Pocket PC’en. Vi valgte derfor at implementere interaktionselementerne således, at de i forhold til disse regler, ville skabe mindst mulig diskontinuitet¹.

¹ Se eventuelt afsnit 2.5 side 18

12. Design af udseende for interaktionselementer (Pocket PC)

I dette skridt besluttede vi hvordan elementerne i brugergrænsefladen skulle se ud på Pocket PC'en. Vi tog udgangspunkt i de beslutninger vi havde taget for Tablet PC'en, men måtte samtidig tage hensyn til de begrænsninger der eksisterer på en Pocket PC. Vi lavede generelt elementerne væsentligt større en normen på Pocket PC'er, da Ajour gerne skulle kunne betjenes med fingrene, frem for med den normale pen.

13. Revidering af mock-ups (Pocket PC)

Det var klart for os fra starten af trin 11, at vi ikke kunne være sikre på hvordan et tænkt element på en mock-up, ville se ud når det blev implementeret. Dette var ikke noget problem på Tablet PC'en, men betød at sideopsætningen flere steder måtte revideres på Pocket PC'en. Efter at have lavet et par forsøgs-sider i Visual Studio, gik vi derfor tilbage, og reviderede vores mock-ups, så de kom til at stemme mere overens med hvad der rent faktisk kunne implementeres. Dette betød ligeledes at vi måtte anvende andre Graceful Degradation regler, der i mindre grad sikrede kontinuitet.

14. Opsætning af program (Pocket PC)

I dette trin blev de reviderede mock-ups implementeret i Visual Studio 2003 / C# Compact Framework.

15. Design af grafik (Pocket PC)

Da hele programmet på Pocket PC var blevet gjort funktionsdygtig, gennemgik vi den, og besluttede hvilke typer grafik der var nødvendige, og hvor. Generelt valgte vi en minimalistisk tilgang til grafik, mens vi gik efter lyse farver som baggrund og knapper, med stor kontrast til markerede knapper og tekst. Et stort problem var, at "C# Compact Framework", som bruges på Pocket PC'en ikke tillader knapper med mere end en linie tekst, hvilket krævede at vi selv tegnede de knapper der havde brug for dette. Dette forsinkede både udviklingen, og betød dårligere afviklingshastighed på Pocket PC'en.

16. Design af grafik (Tablet PC)

I det sidste udviklingstrin overførte vi de grafiske beslutninger som vi havde taget på Pocket PC'en, til Tablet PC'en. En mulighed havde været at bruge såkaldte "tabbed panes", men vi følte at dette valg ville være for inkonsistent med Pocket PC'en. Vi fravalgte at benytte den hierarkiske opbygning, som set på Pocket PC'en, da der var rigelig plads til at præsentere alle sider i programmet i et lag, på Tablet PC'en.

5 Programmer

Vi vil i dette afsnit beskrive og illustrere resultatet af den forudgående design- og udviklingsproces. Processen udmundede i tre programmer; et program kørende på en Tablet PC, et kørende på en Pocket PC, med samme funktionalitet som Tablet PC udgaven, og til sidst et program kørende på en Pocket PC, men med reduceret funktionalitet i forhold til de to andre. Den reducerede version kan ses nærmere beskrevet i appendix B. Der blev brugt omkring 150 - 170 mandetimer på disse programmer der sammenlagt består af ~26.000 linier kode. Koden består udelukkende af de tre programmers brugergrænseflader og den dertil hørende begrænsede funktionalitet. Hardwaren i de to benyttede apparater bestod af:

HP iPAQ Pocket PC h1940:

- 64 MB RAM
- 266 MHz
- 240x320 pixels TFT skærm
- Microsoft Windows Pocket PC 2003 Professional



Figur 15 - HP iPAQ Pocket PC h1940

Samsung SyncMaster 173VT:

- 17 tommer
- Touch Screen
- Max opløsning: 1280x1024 60Hz
- Koblet til stationær PC:
 - 650 MHz Athlon processor
 - 128 MB RAM



Figur 16 - Samsung SyncMaster 173VT

5.1 Tablet PC

Udvikling samt implementering af Tablet PC'en fandt sted i trinene 8 - 11², hvilket lå før udvikling af Pocket PC'erne. Vi valgte at implementere Ajour Tablet i en opløsning på 800x600 pixel, da denne opløsning benyttes på Judex' Tablet PC udgave.

Papirjournal

I trin 8 var det blevet besluttet, hvilke kilder vi ville basere Ajour på – vision, Judex' database og papirjournalen. Det stod dog hurtigt klart at det i høj grad ville være papirjournalen der ville skabe basis for selve indholdet i registreringsdelen, da denne, med en enkelt undtagelse, i forvejen indeholder et komplet billede af den information falckredderne har behov for at registrere. Vi valgte at inddele systemet i ti menupunkter, hvilket kan ses på Figur 17. De ti menupunkter fik følgende betegnelser

Kørsel, **Patient**, **Behandling før ankomst**, **Tilstand ved ankomst**, **Skader**, **Behandling af patient**, **Medicin**, **Målinger**, **GCS** og

Traumeskema. Traumeskemaet er ikke repræsenteret på Papir Journalen, men er på et separat skema, som vi har tilføjet til systemet.

Navigation

The form is titled 'F 000000 Ambulancejournal'. It is divided into several sections:

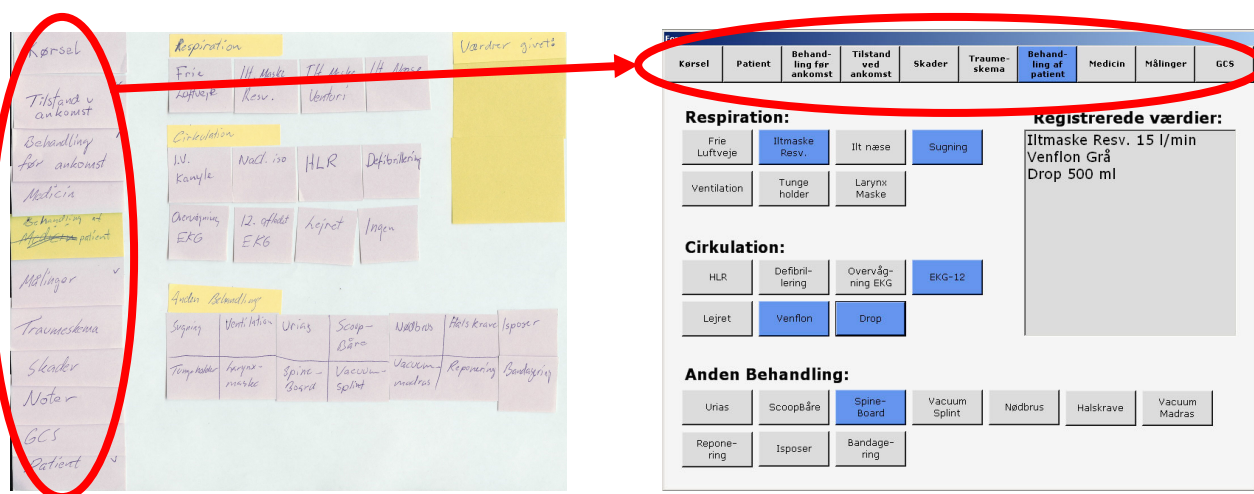
- Top Section:** Patient information including name, address, and contact details.
- Medical History:** Sections for 'Behandling før ankomst' (Treatment before arrival) and 'Tidligere sygdomme' (Previous diseases).
- Vital Signs and Observations:** Sections for 'Tilstand ved ankomst' (Condition on arrival), 'Målt kl.' (Measured vitals), 'Saturation', 'Blodtryk' (Blood pressure), 'Puls', 'Resp. frekv.' (Respiratory frequency), 'Rytme' (Rhythm), 'Blodsukker' (Blood sugar), and 'Peakflow'.
- Treatment and Medication:** Sections for 'Behandling af patient' (Patient treatment) and 'Medicin' (Medication).
- Other Sections:** 'GCS' (Glasgow Coma Scale), 'Påretningsundersøgt' (Wound examination), 'Lokalisering af skader' (Localization of injuries), and 'Notater' (Notes).

The form includes various checkboxes and input fields for recording patient status and treatment.

Figur 17 - Papir Journal

² Se Figur 14 under designforløb

På baggrund af disse menupunkter gik vi i trin 10 i gang med at organisere og opstille mock-ups. En af de første beslutninger der skulle tages, var hvordan navigationen mellem de ti menupunkter skulle fungere. Det blev i denne fase besluttet at placere de ti menupunkter som knapper i venstre side af grænsefladen, hvilket kan ses på Figur 18. Bevæggrundene for dette valg var at skabe en lettilgængelig og overskuelig måde at tilgå navigationen på. Ved selve implementeringen foretog vi dog en ændring, da vi manglede plads horisontalt, men havde tilstrækkelig vertikal plads. Af denne grund blev navigationen flyttet op i toppen af brugergrænsefladen. Knappernes rækkefølge blev i første omgang lavet ud fra den rækkefølge som falckredderne i praksis behandler deres patienter i, men vi valgte efterfølgende at ændre denne, således at den stemte overens med den rækkefølge, som det var nødvendigt at benytte på Pocket PC'en. Det er vores erfaring, at redderne sjældent laver de dele af registreringen, der kan vente, indtil andre og mere vigtige opgaver er løst. Et eksempel herpå kan være, at "tilstand ved ankomst" i princippet kan vente til patienten er afleveret på sygehuset, mens "målinger" skal laves og registreres løbende.

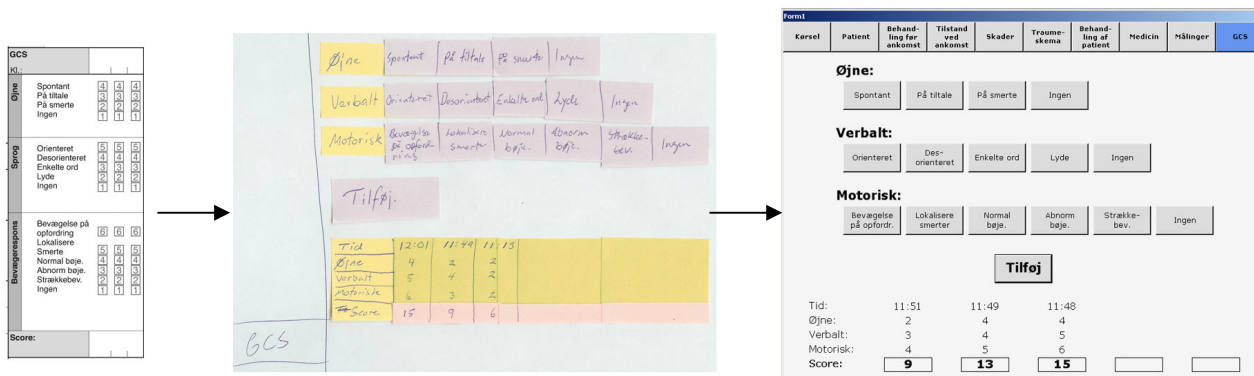


Figur 18 - Navigation på mock-ups → program

Den store udfordring ved trin 10 var endvidere at fastlægge og skabe en brugergrænseflade, hvor de ti menupunkters indhold blev præsenteret på en sammenhængende måde ved hjælp af en række, af os valgte, interaktionselementer. Desuden stod vi med en papirjournal, hvorpå måden at registrere, var meget statisk. Har falckredderne først valgt en værdi kan denne reelt ikke "slettes" igen. Desuden var vi klar over, at det var muligt at skabe forbindelse mellem falckcentralen, sygehuset og vores system, hvilket ligeledes åbnede for en mere dynamisk registrering.

Designbeslutninger

Springet fra papirjournal til prototype viste sig at være en større todelt proces, bestående af at gå fra den indledende vision og papirjournalen til mock-up, samt at gå fra mock-up til implementeret prototype. På Figur 19 er udviklingen af GCS menupunktet illustreret. Typisk var det største spring rent grænseflademæssigt springet fra papirjournal til mock-up, hvilket Figur 19 illustrerer. Vi fandt i dette tilfælde grænsefladen på papirjournalen for værende meget tætpakket og til dels uoverskueligt. Vi valgte derfor at lave en række knapper, der blev benyttet ved hver scoring, frem for at benytte nye hver gang, som er tilfældet ved papirjournalen. Endvidere kunne vi på denne måde i bunden af grænsefladen skabe detaljerede scoringer, hvor det vil være nemt og overskueligt for falc kredderne at se resultaterne af foretagne scoringer. Desuden valgte vi, i modsætning til papirjournalen, at liste de seneste scoringer længst mod venstre, da disse blandt andet ligger til grund for en værdiangivelse i Traumeskemaet. Vi har valgt at benytte dette princip gennemgående, for at skabe en så konsistent som mulig brugergrænseflade. Således rykker de ældste scoringer mod højre, ud af skærmen, i takt med at nye kommer til, mens listbokse med medicin og deslige tilføjer de nyeste indtastninger i toppen.



Figur 19 - GCS udviklingsforløb

En andet designvalg der kom ud af denne proces var, at vi stedet for "krydsmarkeringer" ville benytte "knapmarkeringer". På denne måde vil eksempelvis "Frie luftveje" blive skrevet på selve knappen og når knappen blev trykket på, ville den blive markeret med en blå farve. Denne form for markering giver i langt højere grad falc kredderen et umiddelbart billede af situationen frem for det et beskedent kryds er i stand til. Det blev endvidere besluttet, at falc kredderne ligeledes skulle kunne afmarkere en knap. Denne designbeslutning blev taget for at imødekomme eventuelle fejlindtastninger der højst sandsynligt vil opstå, når der skal registreres i ambulance der bevæger sig med høj hastighed. Figur 20 illustrerer "Behandling af patient" -udviklingsforløbet hvor nævnte design ideer blev benyttet.



Figur 20 - Behandling af patient udviklingsforløb

På papirjournalen i Figur 20 er der desuden plads til værdiangivelse ved flere af knapperne. For at imødekomme denne form for registrering valgte vi at introducere en popup boks, som vist på Figur 21.

The figure shows a value entry popup dialog box. It contains a 'Respiration:' section with a 'Frie Luftveje' button and a 'Ventilation' button. A 'Cirkulation' section has 'HLR' and 'Lejret' buttons. An 'Anden Be' section has 'Urias', 'ScoopBare', 'Board', 'Splint', 'Nødbrus', 'Halskrave', and 'Vacuum Madras' buttons. The 'Registrerede værdier:' section shows 'Iltmaske Resv.' with a value of '15 l/min'. There are 'OK' and 'Fortryd' buttons at the bottom.

Figur 21 – Værdiangivelse

Som det fremgår af ovenstående figur, bestod "Iltmaske Resv." af multichoice-værdier, hvilket betød, at vi uden større problemer kunne liste disse, således at falkredderne var fri for manuelt at gå ind og indtaste værdierne, sådan som det er tilfældet på papirjournalen. Ved valg af en værdi og et efterfølgende tryk på "Ok" knappen, bliver værdierne sat ind i "Registrerede værdier:" tekstboksen, sammen med tidspunkt for værdiangivelsen. I overensstemmelse med papirjournalen, valgte vi endvidere at multichoice-knapper skal være markerede efter første værdiangivelse, selvom der reelt kan registreres adskillige værdier for samme knap over tid.

Som nævnt var vores grundlæggende design ide at skabe en sammenhængende brugergrænseflade på tværs af de ti menupunkter, hvilket ofte betød et stort spring fra vision og papirjournal til mock-

up. I enkelte tilfælde var der dog en god overensstemmelse disse to imellem, hvilket eksempelvis fremgår af Figur 22. Denne form for overensstemmelse var som nævnt ikke mulig på tværs af alle punkter, men i de tilfælde, hvor det var, anså vi det for værende meget positivt, idet at vi således sikrede kontinuitet papirjournalen og prototypen imellem.



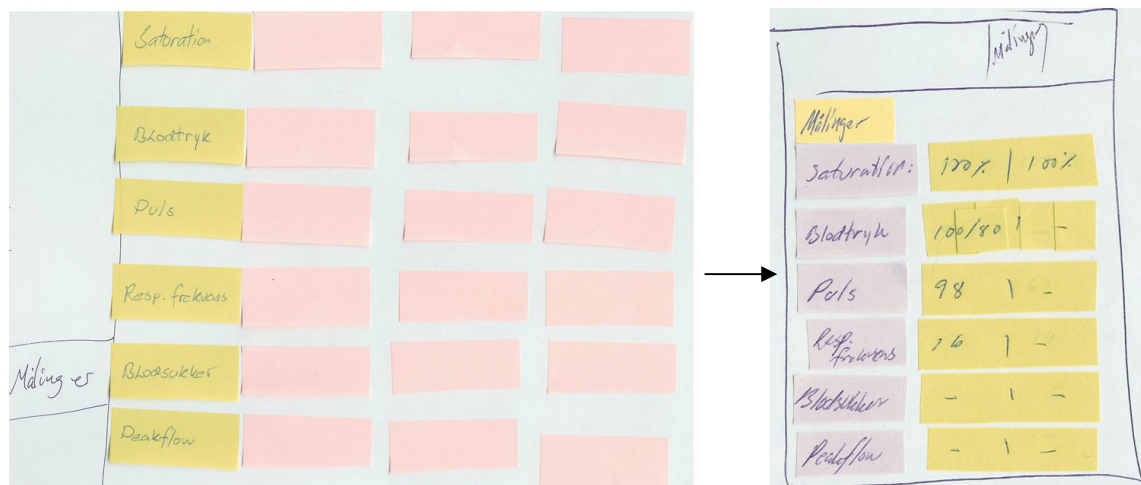
Figur 22 - Målinger udviklingsforløb

5.2 Pocket PC

Fra trin 11 – 14 foregik udviklingen af den første Pocket PC prototype. Denne prototype har funktionalitet svarende til den der findes på Tablet PC'en. Design ideerne bag Pocket PC'en tog hovedsageligt sine udgangspunkter i de allerede udviklede mock-ups til Tablet PC'en. Graceful Degradation reglerne blev benyttet til at sikre kontinuitet mellem Tablet PC'en og Pocket PC'en. Pocket PC prototypen blev designet i en opløsning på 240x320, hvilket svarer til opløsningen på Judex' Pocket PC.

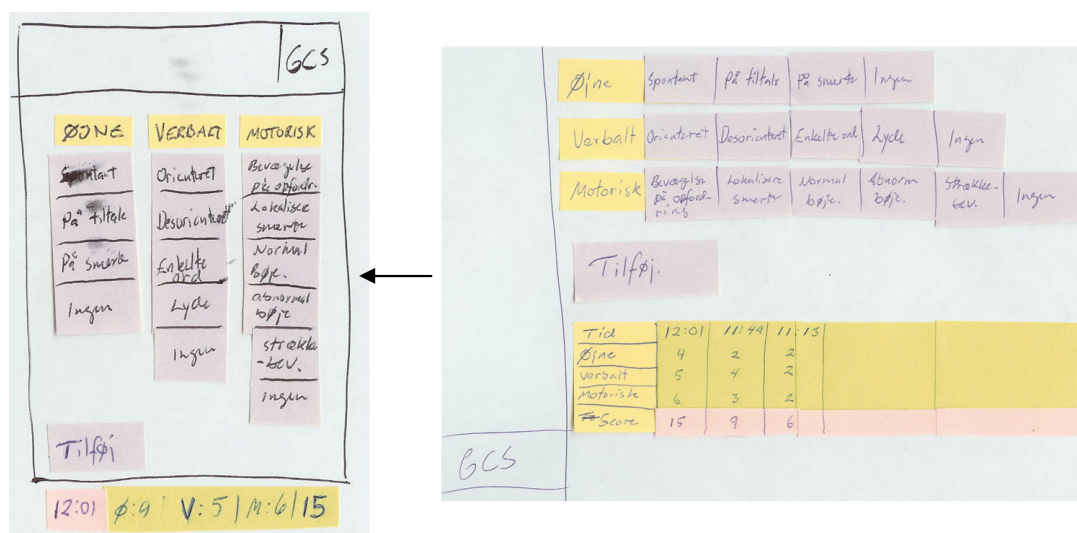
Designbeslutninger

I trin 11 begyndte udviklingen af mock-ups til Pocket PC'en, på baggrund af Tablet PC'ens mock-ups. I adskillige tilfælde benyttede vi os af dimensioneringsreglen. Dette betød, at elementerne på Tablet PC'en blev skaleret ned til Pocket PC'ens interface. Et eksempel på dette kan ses på Figur 23.



Figur 23 - Målinger fra Tablet PC til Pocket PC

Den største hurdle ved at gå fra Tablet PC'en til Pocket PC'en var ikke så meget at elementerne skulle skaleres ned. Dette var en forholdsvis ligetil procedure, der "kun" krævede overvejelser i forhold til tekstens størrelse og font. Problematisk var det i højere grad, at vi på Tablet PC'en gik fra et display på 800x600 pixel til et display på Pocket PC'en på 240x320 pixel. Vi gik dermed fra et display der havde landskabs-proportioner til et display med portræt-proportioner, hvorfor det var nødvendigt at omrokere flere af brugergrænsefladens elementer. For at gøre dette anvendte vi orienteringsreglen³. Måden dette blev gjort på, var som vist på Figur 24, hvor knapperne på Tablet PC'en er arrangeret horisontalt, men grundet skærmens dimensioner måtte arrangeres vertikalt på Pocket PC'en.

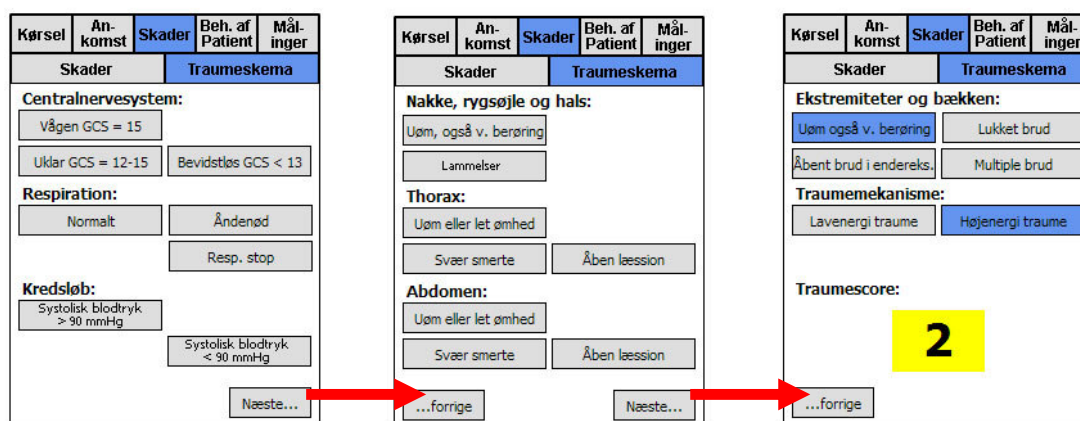


Figur 24 - GCS fra Tablet PC til Pocket PC

³ Se eventuelt afsnit 2.4 side 15

Da vi indså at vi ikke havde med en lineær dimensionering at gøre, og således ikke bare kunne skalere brugergrænsefladens elementer ned, måtte vi tage en beslutning, hvorvidt det var ønskeligt at benytte os af scrollbars eller undersider for at få vist alt indhold. Hensigten med systemet har hele tiden været, at man skulle kunne benytte sine fingre til at navigere rundt med. Dette ville kræve en scrollbar med en forholdsvis stor bredde, hvilket ville betyde en yderligere pladsbegrænsning horisontalt, hvor der i forvejen var mangel på plads. Desuden havde vi hele vejen gennem design processen benyttet os af knapper til at angive indholdsskift, hvorfor vi anså brugen af knapper som navigationselementer for at være et mere konsistent valg, som i højere grad ville støtte brugerens intuition. Det var derfor naturligt at introducere undersider, hvor man ved hjælp af en knap ”Næste...” ville skifte over til et nyt indhold.

Et eksempel på disse undersider kan ses på Figur 25. Havde vi valgt at benytte os af scrollbars ville vi have introduceret et element der kun i meget sjældne tilfælde vil kunne eksistere på Tablet PC'en.



Figur 25 - Underside navigation i Traumeskemaet

Vi ønskede for så vidt muligt at undgå at benytte mere drastiske indgreb på Pocket PC'en i form af eksempelvis substitutionsreglen, hvor man udskifter et interaktionselement med et andet element med tilsvarende funktionalitet. Dette blev dog en nødvendighed på grund af den utilstrækkelige skærmstørrelse på Pocket PC'en. Som vist på Figur 26 havde vi til Tablet PC'en lavet et keyboard, hvorved man kunne indtaste medicin en patient måtte have fået inden falckreddernes ankomst til ulykkesstedet.

The screenshot shows a medical form titled 'Form1' with tabs for 'Kørsel', 'Patient', 'Behandling før ankomst' (selected), 'Tilstand ved ankomst', 'Skader', 'Traumeskema', 'Behandling af patient', 'Medicin', 'Målinger', and 'GCS'. A modal window titled 'Angiv medicin' is open, featuring a touch keyboard with a numeric keypad (0-9, /), an alphabetic keypad (a-z, Æ, Ø), and a row with 'Stor', 'Delete', and a hyphen key. Below the keyboard are 'OK' and 'Fortryd' buttons. A 'Lægmand' button is visible at the bottom left of the form.

Figur 26 - Touch keyboard på Tablet PC

The screenshot shows a medical form on a Pocket PC with a text field containing 'Nitro 5mg'. Below the field is a keypad with numeric keys (1-9, 0), a key for 'a/A', and a 'Slet' key. At the bottom are 'OK' and 'Fortryd' buttons.

Figur 27 - Keypad på Pocket PC

Det var dog klart, at vi grundet Pocket PC'ens skærmstørrelse ikke ville kunne presse et sådant keyboard ind. Derfor valgte vi at lave et alternativt keyboard, som vist på Figur 27, baseret på en mobiltelefons keypad. Dette bevirkede naturligvis en noget anden form for interaktion end på Tablet PC'en, men dog med en vis forhåbning om, at brugen af noget så familiært som en mobiltelefons keypad ville støtte brugerens intuition i forhold til interaktionen.

Navigation

Udover de enkelte siders transformation fra Tablet PC mock-ups til Pocket PC mock-ups måtte vi også tage en beslutning i forhold til den overordnede navigation på Pocket PC'en. Som vist på Figur 28 består den overordnede navigation i systemet på Tablet PC'en af ti knapper i toppen af displayet.

Kørsel	Patient	Behand- ling før ankomst	Tilstand ved ankomst	Skader	Traume- skema	Behand- ling af patient	Medicin	Målinger	GCS
--------	---------	--------------------------------	----------------------------	--------	------------------	-------------------------------	---------	----------	-----

Figur 28 - Overordnet navigation på Tablet PC

Det var tydeligt at det ikke ville være muligt at skalere ti knapper med en samlet bredde på 800 pixels ned til en samlet bredde på 240 pixels. Vi valgte derfor at benytte os af en kombination af dimensionerings- og placeringsreglen. Resultatet heraf, som kan ses på Figur 29, blev en form for hierarkisk navigation i toppen på PDA'en, hvor fem knapper allerøverst angav overordnede kategorier.

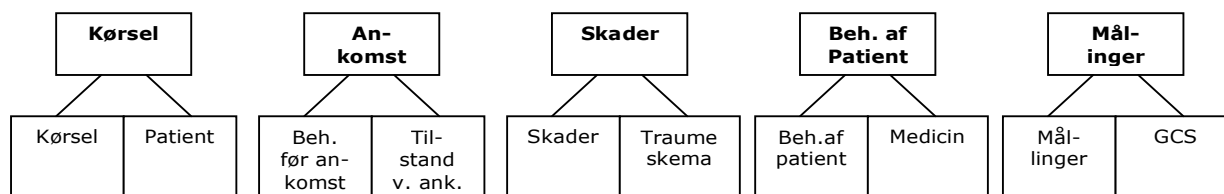
Inden mock-ups blev lavet til Pocket PC'en var det vores idé at opbygge rækkefølgen af den overordnede navigation på Tablet PC'en efter falckreddernes typiske registreringsforløb. Således havde vores indledende navigation følgende rækkefølge: Kørsel, Patient, Behandling før ankomst, Tilstand v. ankomst, Skader, Beh. af patient, GCS, Målinger, Medicin og Traumeskema. Dette var dog kun tænkt som en hjælp til redderne, da vi har observeret, at reddernes måde at registrere på langt fra altid følger denne "logiske" rækkefølge.

Vi besluttede at lave en opbygning på Pocket PC'en, hvor menupunkterne var opdelt i et hierarki. Som det kan ses på Figur 28, lavede vi fem hovedpunkter, med hver to underpunkter, således at overskriften for et par af punkter var beskrivende for begge punkter. Hver af de 5 gange 2 underpunkter svarende til knapperne vist i Figur 28. Resultatet af disse overvejelser kan ses på Figur 29.

Kørsel	An- komst	Skader	Beh. af Patient	Mål- inger
Beh. før ankomst		Tilstand v. ankomst		

Figur 29 - Overordnet navigation på Pocket PC

Under hver kategori blev der placeret yderligere to knapper der angav specifikke menupunkter svarende til knapperne vist i toppen af Figur 28. Pocket PC'ens hierarkiske opbygning kan ses på Figur 30



Figur 30 - Hierarkisk opbygning Pocket PC

6 Beskrivelse af eksperiment

I dette kapitel vil vi beskrive hvorledes vores brugbarhedseksperimenter er opbygget og udført. Således vil læseren i dette kapitel finde relevante detaljer omkring hvilke testpersoner der har deltaget, og hvorledes vi har indsamlet data til senere analyse. Formålet med at lave disse brugbarhedseksperimenter var, at undersøge hvilke brugbarhedsproblemer der var at finde på de forskellige apparater, samt at se på arbejdsbelastning, effektivitet og formålstjenstlighed, således at vi kunne besvare problemformuleringens delspørgsmål 3.

6.1 Planlægning af test

I dette afsnit beskrives planlægningen af vores eksperimenter. Denne planlægning er baseret på elementer taget fra [Rub94]. Herefter klargøres eksperiment-opgavernes udformning, samt eksperimenternes omgivelser, testpersoner og udstyr. Til sidst forklares, hvilke målinger der blev registreret under eksperimenterne.

Selve testene var planlagt således, at vi tog ud til Falck i Nibe, og lavede så mange test som muligt. Normalt var dette maksimalt to test per dag, med en enkelt undtagelse, da en redder tilbød at komme ind på sin fridag. Så vidt det var muligt lavede vi de test der kunne laves, og behandlede data i den uundgåelige ventetid, når redderne arbejdede.

Det primære eksperiment omfattede to multiplatform systemer, og bestod således af fire apparater. For at få det mest interessante sammenligningsgrundlag, og for at hver falckredder ikke skulle forstyrres unødigt mange gange, lod vi hver falckredder teste to apparater umiddelbart efter hinanden, hvorefter vi ventede mindst en dag, før denne falckredder kunne teste de to andre apparater.

Vi valgte, for at der skulle være mindst mulig indlæringseffekt mellem de to apparater, at hver testsession bestod af en Pocket PC det ene system og en Tablet PC fra det andet.

Opdelingen var lavet således, at hver falckredder skule gennemgå de fire apparater, i en unik rækkefølge. De førnævnte begrænsninger betød at der blev otte mulige rækkefølger, hvoraf seks blev udvalgt ved lodtrækning. Den planlagte fordeling af testpersoner på apparater kan ses i Tabel 5, mens den endelige fordeling kan ses i Tabel 6.

Antal Testpersoner	Apparater, rækkefølge
1	Ajour Tablet – amPHI Pocket, Ajour Pocket – amPHI Tablet
1	Ajour Pocket – amPHI Tablet, amPHI Pocket – Ajour Tablet
1	amPHI Pocket – Ajour Tablet, amPHI Tablet – Ajour Pocket
1	amPHI Tablet – Ajour Pocket, amPHI Pocket – Ajour Tablet
1	Ajour Pocket – amPHI Tablet, Ajour Tablet – amPHI Pocket
1	Ajour Tablet – amPHI Pocket, amPHI Tablet – Ajour Pocket
0	amPHI Pocket – Ajour Tablet, Ajour Pocket – amPHI Tablet
0	amPHI Tablet – Ajour Pocket, Ajour Tablet – amPHI Pocket

Tabel 5 - Fordeling af testpersoner på apparat-sammensætninger

Testperson nr.	Apparater, rækkefølge
4	Ajour Tablet – amPHI Pocket, Ajour Pocket – amPHI Tablet
3	Ajour Pocket – amPHI Tablet, amPHI Pocket – Ajour Tablet
6	amPHI Pocket – Ajour Tablet, amPHI Tablet – Ajour Pocket
5	amPHI Tablet – Ajour Pocket, amPHI Pocket – Ajour Tablet
2	Ajour Pocket – amPHI Tablet, Ajour Tablet – amPHI Pocket
1	Ajour Tablet – amPHI Pocket, amPHI Tablet – Ajour Pocket

Tabel 6 - Endelig fordeling af testpersoner

6.2 Testpersoner

Testpersonerne var alle reddere på falckstationen i Nibe, og arbejder med en ældre version af amPHI systemet til daglig. Grunden til at valget faldt på disse reddere var, at netop redderne i Nibe ofte er sat i beredskab, og derfor kan have tid til at indgå som testpersoner i vores eksperiment, i modsætning til redderne på de større stationer.

Nr.	Alder	Ansættelse hos Falck	Uddannelse	Erfaring med edb generelt	Erfaring med Falcks edb-systemer	Test udført
1	53 år	32 år	Redder	Næsten ingen	Ingen	Ajour Tablet – amPHI Pocket (20/5) amPHI Tablet – Ajour Pocket (22/5)
2	57 år	25 år	Redder	Meget lidt	Lidt	Ajour Pocket – amPHI Tablet (21/5) Ajour Tablet – amPHI Pocket (24/5)
3	33 år	6 år	Redder	Nogen	Nogen	Ajour Pocket – amPHI Tablet (21/5) amPHI Pocket – Ajour Tablet (24/5)
4	56 år	26 år	Redder	Lidt	En del	Ajour Tablet – amPHI Pocket (22/5) Ajour Pocket – amPHI Tablet (24/5)
5	42 år	18 år	Redder niveau 2	Ikke ret meget	Ingen	amPHI Tablet – Ajour Pocket (23/5) amPHI Pocket – Ajour Tablet (25/5)
6	38 år	16 år	Redder niveau 2	Meget lidt	Lidt	amPHI Pocket – Ajour Tablet (23/5) amPHI Tablet – Ajour Pocket (25/5)

Tabel 7 - Data for falckreddere + dato for test

6.3 Pilot og prototype-test

Denne test blev foretaget med to Specialestuderende fra Aalborg Universitet, og havde til formål at afklare testlederens rolle i forhold til testpersonerne, samt at konstatere om der var stavfejl i såvel opgaver som brugergrænsefladen, som kunne rettes inden den egentlige test. Endelig var der også fokus på, om det var teknisk muligt at gennemføre opgaverne ved brug af Ajour. Testen foregik i Universitetets Usability-laboratorie, frem for hos Falck, men var ellers grundlæggende identisk med den test der blev benyttet til selve eksperimentet, således at en studerende testede Ajour Tablet og amPHI Pocket, mens den anden testede Ajour Pocket og amPHI Tablet. Resultatet af pilottesten var en bedre forståelse af hvorledes testlederen burde agere i forhold til testpersonen, samt at en enkelt opgave, som var uklart formuleret, blev delt op i to opgaver og rettet til. Desuden blev enkelte opgaver kortet en smule ned, da de enkelte apparater i pilottesten i gennemsnit tog 31 minutter og

55 sekunder at gennemgå, hvilket ville betyde, at hver session ville blive på over en time. Endelig blev et antal tydelige tekstfejl i brugergrænsefladen rettet. Der blev ikke lavet analyse af de optagne videoer, hvorfor der heller ikke blev fundet eller rettet egentlige brugbarhedsproblemer.

6.4 Opgaver

Opgaverne var lavet på en sådan måde, at alle typer funktioner blev testet. For at gøre dette var opgaverne lavet som to scenarier, hvor redderne først skulle forestille sig at de havde med et færdselsuheld at gøre, og dernæst at de var ude ved en ældre mand med hjerteproblemer. Opgaverne kan ses i bilag D.

Vi havde tidligere gjort os den erfaring, at falckreddere i gennemsnit brugte 7 minutter og 33 sekunder på at gennemføre seks opgaver, i det amPHI-system som de havde brugt i halvandet år [And05, bilag F]. I den forsøgsopstilling, som vi havde valgt i dette eksperiment, skulle redderne gennemgå 2 x 7 opgaver for to apparater, hvilket ville betyde, at de skulle bruge cirka 17 minutter og 37 sekunder, eller ifølge pilottesten, hvor brugerne ganske vist ikke kendte domænet, 1 time, 8 minutter og 44 sekunder.

Logisk set burde redderne være langsommere end med den version af amPHI, som de kendte, da de ikke havde set de apparater der indgik i testen. Samtidig forventede vi dog, at Judex' amPHI havde fået forbedret brugbarhed, da det havde gennemgået videreudvikling siden det sidst var blevet brugbarhedstestet. Kombineret med, at vi i [Kje03] havde set, at brugere ikke holder op med at opleve brugbarhedsproblemer, blot fordi de har brugt et system i længere tid, mente vi dog, at det var meget sandsynligt, at testpersonerne ikke ville være hurtigere i brugen af amPHI, end de var i det foregående brugbarhedseksperiment.

Vi forventede desuden, at Ajour ville være en brugbarhedsmæssig forbedring af det gamle amPHI. Vi vidste, at en stor del af pilottestens tidsforbrug bundede i testpersonernes dårlige forståelse af domænet, og vi forventede af denne grund, at det gennemsnitlige tidsforbrug i hver testsession ville komme til at ligge omkring 15-20 minutter, eksklusiv NASA TLX test (se Bilag J.4 og J.5) og diskussion af det aktuelle apparat.

6.5 Omgivelser og udstyr

Da det ville være problematisk for Falck at få deres folk til at komme ind til universitetets Usability-laboratorie, valgte vi at foretage alle test hos Falck i Nibe. Dette betød at vi måtte benytte os af vores mobile laboratorie, således at de Pocket PC'erne blev filmet med et påspændt minikamera, mens Tablet PC'erne blev filmet med håndholdt kamera. Da hver testsession bestod af en Tablet PC og en Pocket PC, blev hver testperson præsenteret for to apparater i starten af hver testsession, hvorefter det ene blev sat til side, indtil der skulle bruges, i testens anden del. Disse opstillinger kan ses på Figur 31.



Figur 31 – Forsøgsopstillinger

6.6 Databehandling

6.6.1 Gennemsyn af videooptagelser

For at lokalisere og definere de brugbarhedsproblemer, som hver testperson stødte på, gik vi hver for sig igennem videooptagelserne af alle testsessionerne, og noterede alt hvad testpersonen gjorde i logfiler. For at undgå at blive påvirket af rækkefølgen, valgte vi hver for sig at blande alle optagelserne sammen, og derefter gennemgå dem i tilfældig rækkefølge. Problemerne blev primært fundet ud fra de steder hvor testpersonen gik i stå, blev forvirret, irriteret eller direkte foretog sig noget forkert. Sekundært blev problemer fundet ud fra de ytringer som testpersonen kom med under testen. Et eksempel på dette er følgende problem:

5. Tilstand ved ankomst: Problemer med at finde "Pupiller".

Dette problem blev registreret i gennemgangen i det øjeblik, at det blev klart, at testpersonen ledte efter noget, og var blevet forsinket mere end et par sekunder i at udføre opgaven. At dette problem er baseret på mere end blot forsinkelse, kan ses på, at testperson 2 godt er klar over at han ikke kan finde det søgte, siger "hjælp mig", og ender med at registrere det forkerte (se bilag F).

Generelt blev problemerne kategoriseret som det kan ses i Tabel 8. Vi tillod os at vurdere forsinkelse strengere end vi normalt ville finde rimeligt, da nogle af vores opgaver kunne løses på omtrent 2 sekunder, hvorfor en forsinkelse på 30 sekunder bliver et meget anseeligt irritationsmoment. Eksempelvis brugte testperson 2 i alt 58 sekunder på denne del af testen, og blev så forsinket af problemet, at han endte med at udføre opgaven forkert. Desuden så vi disse effektivitetsreducerende problemer som problematiske i forhold til at systemerne i den virkelige brug vil komme til at tage falckredderens tid fra patienten, som er det primære fokusområde. Således må det være et væsentligt succeskriterium for disse systemer, at enkle registreringer kan udføres meget hurtigt og uden problemer.

	Forsinket	Irritation / Irrationalitet	Forventning vs. Faktisk
Kosmetisk	< 30 sekunder	Lettere	Mindre forskel
Alvorlig	~1 minut	Middel	Betydelig forskel
Kritisk	Fuldstændigt (går i stå)	Stærk	Kritisk forskel

Tabel 8 - Kategorisering af problemer

6.6.2 Evaluatoreffekt

Vi lavede hver en logfil for hver test, samt en problemliste for hvert apparat. Således blev der lavet otte problemlister og 48 logfiler, som efterfølgende blev skrevet sammen. Disse kan ses i bilag E.2 og bilag F. Dette skete ved, at vi gik problemlisterne igennem og diskuterede, hvilke problemer der skulle med i den endelige problemliste, hvorefter logfilerne blev sammenlignet og sammenskrevet. Hvis der var uenighed om problemer, eller forskelligheder i logfilerne, gennemgik vi i fællesskab den relevante del af videoen, og fandt frem til den mest korrekte fremstilling af den hændelse eller det forløb der var uenighed om. Generelt var problemlisterne meget homogene, hvorfor det kun var

nødvendigt at opsplitte et enkelt problem, som var meget bredt, og således kom til at dække for mange forskellige forhold, uden at det i tilstrækkelig grad beskrev det problematiske i disse forhold. Dette problem blev splittet op og tilpasset til andre problemlister, som det kan ses i Figur 32.

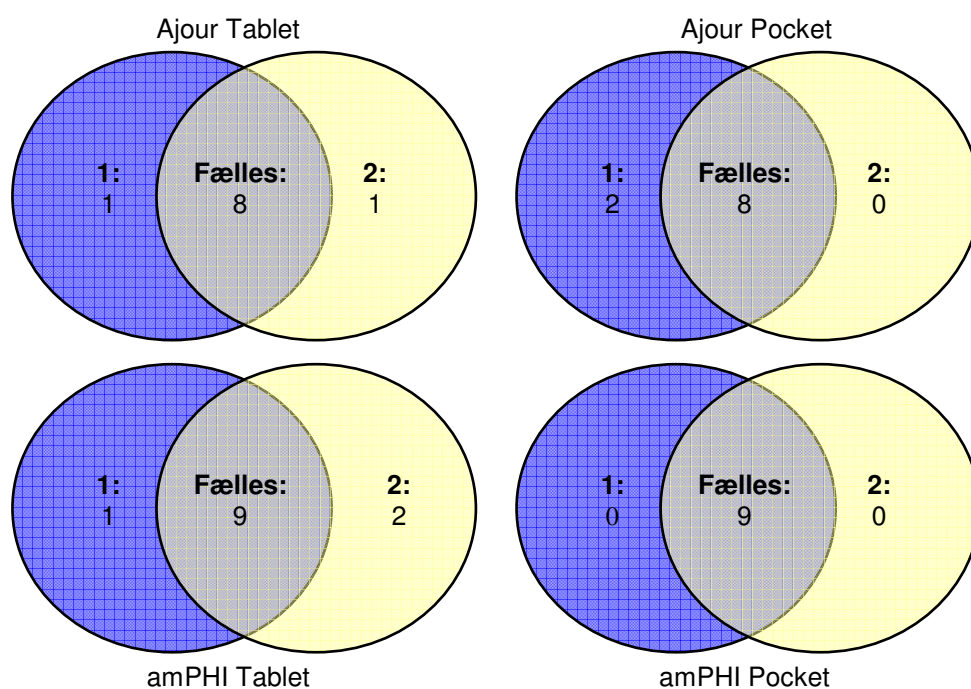
B2: kan ikke finde ønsket menupunkt

- 5: Tilstand ved ankomst: Problemer med at finde "Pupiller".
- 6: Behandling før ankomst/Førstehjælp: Problemer med at finde "Registrer medicin".
- 7: Patient/Tidligere Sygdom: Kan ikke finde "oplyst af patient/pårørende".
- 19: Traumeskema: Kan ikke finde traumeskemaet

Figur 32 - Opsplitning af brugbarhedsproblem

Problembeskrivelserne blev desuden tilpasset, så de havde den samme formulering, på tværs af de fire apparater, i de tilfælde hvor der var tale om ækvivalente problemer.

Problemer der forekom for flere testpersoner, men med forskellig grad, blev altid indført i de samlede problemlister med den størst observerede alvorlighed, således at et problem oplevet af både testperson 1 og 2, men som henholdsvis kosmetisk og alvorligt, blev klassificeret som alvorligt. Dette forhold gør sig også gældende for den endelige samling af problemlisterne, som kan ses i bilag E.1



Figur 33 - Evaluatoreffekt

Som det kan ses på Figur 33, fandt evaluator 1 og evaluator 2 omtrent lige mange problemer. Fælles for de problemer der kun blev fundet af en evaluator var, at de typisk kun var oplevet af en enkelt testperson, eller var et biprodukt af et andet problem, hvor en evaluator havde set det som et problem, mens den anden havde delt problemet i to diskrete problemer. Følgende er eksempler på disse typer problemer:

25: Patient: Forstår ikke hvad "Vis tastatur" gør.

Dette problem opstod fordi testpersonen brugte et stykke tid på at overveje hvordan man skulle komme videre til at indtaste de data, som opgaven krævede. For en person kunne dette tolkes som en pause, hvori testpersonen blot overvejede skærbilledets indhold, mens det for en anden kunne se ud som om, at testpersonen ikke forstod knappens formål. Ved eftersyn var det klart, at testpersonen havde et reelt problem med at forstå mulighederne i denne knap, hvorfor dette blev angivet som et problem.

20: Førstehjælp: Kan ikke indtaste tal under "Medicin fritekst"

14: Førstehjælp: Forstår ikke "123" knappen på keyboard. Denne reagerer forkert.

Disse to problemer opstod i forlængelse af hinanden. Problem nr. 20 viser således testpersonens umiddelbare problemer med at forstå og bruge en del af systemet, men derudover tilføjede den ene evaluator et ekstra problem, da systemet desuden havde en direkte funktionsfejl i forbindelse med den knap der skulle bruges til at udføre opgaven.

6.6.3 Arbejdsbelastningsdata

Arbejdsbelastningen blev målt ved, at hver testperson, efter endt test med det enkelte apparat, udfyldte et skema med vedkommendes vurdering af arbejdsbelastningen for de enkelte faktorer, på en skala fra 0-100 %, hvorefter vedkommende fik en stak på 15 sedler, hvorpå der skulle sættes kryds ved den faktor, som vedkommende vægtede højest.

Det første skridt i behandlingen af de indsamlede arbejdsbelastningsdata, bestod i at registrere værdi og vægt for hver faktor. Udover at øge overskueligheden ligger registreringen til grund for de efterfølgende udregninger af arbejdsbelastningen. Udregningen af den samlede arbejdsbelastning blev udført på baggrund af en tilpasset værdi for hver af de seks faktorer. Denne tilpassede værdi er et udtryk for, hvor stor en belastning hver faktor havde på testpersonen under eksperimentet, multipliceret med de vægte, som testpersonen tillagde disse faktorer. Ved at lægge de seks tilpassede værdier sammen, og dividere disse med femten, opnås således den samlede

arbejdsbelastning. Dette svarer til de femten vægte testpersonerne fordelte på faktorerne. Herudover har vi ligeledes beregnet gennemsnittet for de seks testpersoner ved hvert eksperiment, for på den måde at kunne belyse eventuelle ligheder og forskelle eksperimenterne imellem.

6.6.4 Brugbarhedsmålinger

Ved gennemsyn af videoerne af de fire eksperimenter blev der udover brugbarhedsproblemer, indsamlet en række andre data fra testpersonerne. En nærmere forklaring af, hvad disse data indeholder følger her:

- **Formålstjenestelighed** – Formålet med disse data er at belyse, hvor præcist og fuldendt testpersonerne fuldførte eksperimenterne på de fire apparater. Opgaven blev gennemgået under hvert apparat, og antallet af ikke-fuldførte opgaver registreret i log filen. Hvorvidt de enkelte opgaver blev registreret som gennemført eller ej, blev vurderet af evaluatorene hver for sig, og diskuteret på plads ved sammenfletningen af problemlisterne.
- **Effektivitet** – Tiden for løsningen af de enkelte opgaver. Tiden blev for den enkelte opgave registreret fra det tidspunkt, hvor testpersonen havde påbegyndt selve opgaven, til tidspunktet hvor denne tydeligt markerede, at opgaven nu mentes at være fuldført. Inkluderet i denne tid er også spørgsmål fra testpersoner og hjælp fra testleder forstået på den måde, at tidsregistreringen ikke blev stoppet, hvis testpersonen havde spørgsmål undervejs eller i værste tilfælde, at testlederen måtte give testpersonen hjælp. Det blev med andre ord set på som en integreret del af opgaveforløbet.
- **Tilfredshed** – Er baseret på brugernes kommentarer og bemærkninger i forbindelse med eksperimenterne og de efterfølgende interviews. De enkelte testpersoner kom under eksperimenterne med en række kommentarer omkring deres vurdering af systemerne. De efterfølgende interviews blev benyttet til at få disse kommentarer uddybet og nærmere specificeret. Kommentarerne blev i alle tilfælde kategoriseret i forhold til om kommentaren var positiv eller negativ i forhold til systemerne. Kommentarer som gik igen ved den samme testperson fra eksperiment til eksperiment, blev ved hvert enkelt eksperiment listet som enten positiv eller negativ kommentar. Den enkelte kommentar blev kun talt med en gang per apparat.

7 Testresultater

I dette kapitel vil vi præsentere resultaterne af det brugbarhedseksperiment, som vi har foretaget med amPHI og Ajour.

Kapitlet er opdelt, således at vi først præsenterer og analyserer de brugbarhedsproblemer der er fundet. Dernæst præsenteres resultaterne af vores undersøgelse af den arbejdsbelastning der er forbundet med brugen af apparaterne, i form af en NASA TLX "workload" undersøgelse, og endelig viser vi resultaterne af de data der er opsamlet i forhold til systemets formålstjenstlighed og effektivitet, samt testpersonernes tilfredshed. Det gælder for alle afsnit, at vi lægger vægt på at finde forskelle og ligheder i de forskellige platforme, på tværs af systemerne, med henblik på at kunne besvare vores problemformulerings delspørgsmål 3.

Vi har brugt "ANalysis Of VAriance between groups" (ANOVA) til at konstatere hvorvidt der er statistisk signifikante forskelle mellem resultaterne af test med de forskellige apparater. Når resultaterne benævnes mindre signifikante, er det med et signifikansniveau på 10 % eller derunder. Når de benævnes signifikante, er det med et signifikansniveau på 5 % eller derunder. Når de benævnes stærkt signifikante, er det med et signifikansniveau på 1 % eller derunder.

7.1 Brugbarhedsproblemer

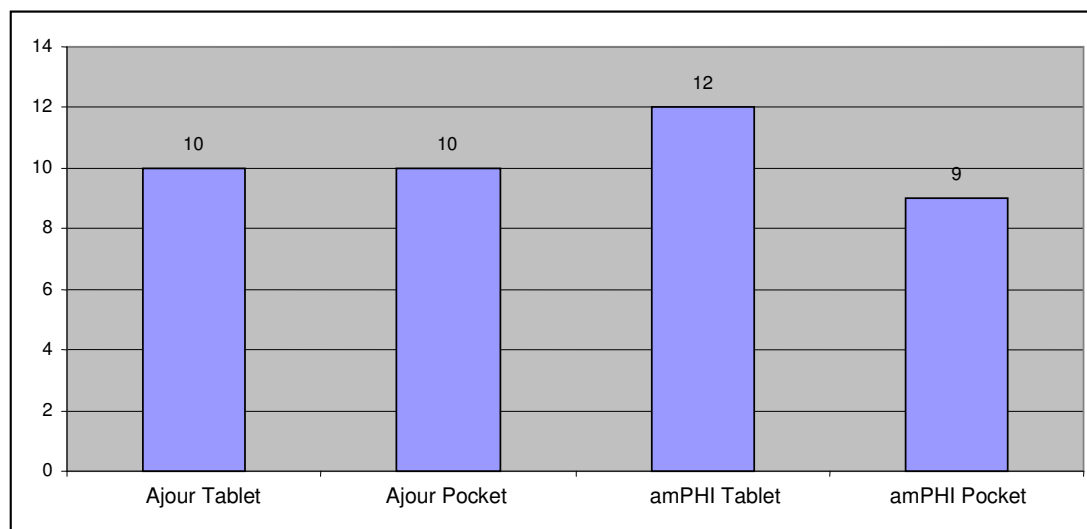
I dette afsnit benytter vi mængden af oplevede brugbarhedsproblemer, til at sammenligne de fire apparater, som vi har testet. Først sammenlignes det samlede antal problemer der er fundet under hver test, hvorefter mængderne af kosmetiske, alvorlige og kritiske problemer, og de totale antal oplevede problemer sammenlignes. Derefter diskuterer vi brugbarhedernes typer, og slutter af med at samle op på de fundne resultater, og konkludere på disse.

7.1.1 Antal brugbarhedsproblemer

Antal brugbarhedsproblemer fordelt på apparater

Som det kan ses i bilag E, fandt vi frem til, at testpersonerne i alt oplevede 27 forskellige brugbarhedsproblemer, hvoraf 4 var kritiske, 11 var alvorlige og 12 var kosmetiske. Af disse var 19 unikke for et enkelt apparat, mens 4 blev fundet på 2, 2 på 3 og 2 på alle 4 apparater. Fordelingen af

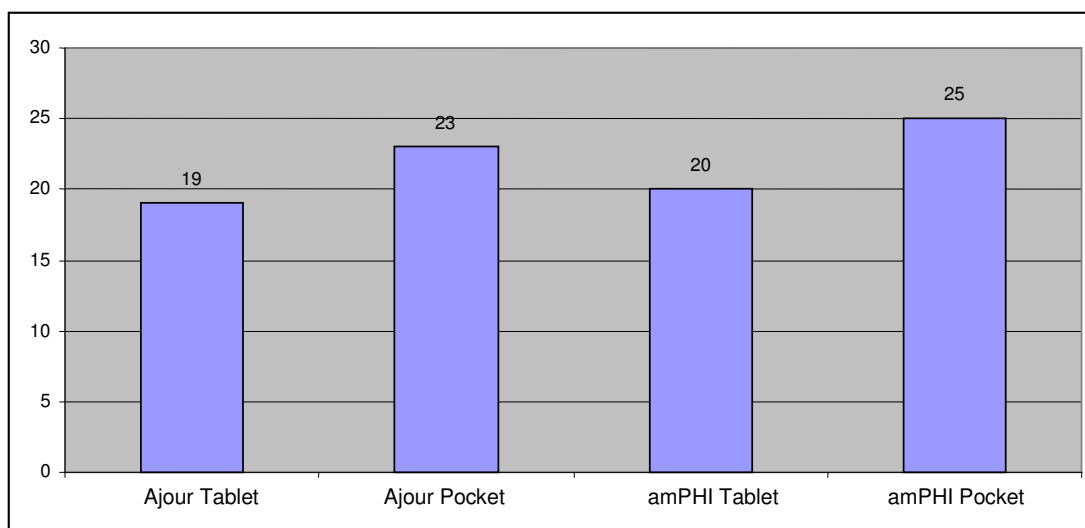
fundne problemer var dog meget jævn. Som det kan ses på Figur 34, blev der fundet 10 forskellige brugbarhedsproblemer på begge apparaterne i Ajour-systemet, mens der blev fundet 12 på amPHI Tablet og 9 på amPHI Pocket.



Figur 34 - Antal brugbarhedsproblemer på apparater

Det blev fundet flere problemer på amPHI Tablet, end på de andre apparater, men på Figur 35 kan det dog ses, at testpersonerne ikke havde problemer flere gange i brugen af amPHI Tablet.

Tværtimod oplevede testpersonerne, at der var problemer 20 gange, hvilket er en enkelt gang mere end i brugen af Ajour Tablet, og henholdsvis 2 og 5 oplevede problemer mindre end Ajour Pocket og amPHI Pocket. På Figur 35 kan det desuden ses, at der forekommer færre problemer på Ajour, i forhold til amPHI. Der er ikke tale om en stor forskel, da amPHI Tablet har 2 unikke problemer mere end Ajour Tablet (20 % flere), og et enkelt tilfælde mere af, at der opleves problemer (5.3 % flere), mens Ajour Pocket har et unikt problem mere end amPHI Pocket (11 % flere), og amPHI har to oplevede problemer mere end Ajour Pocket (8,7 % flere).



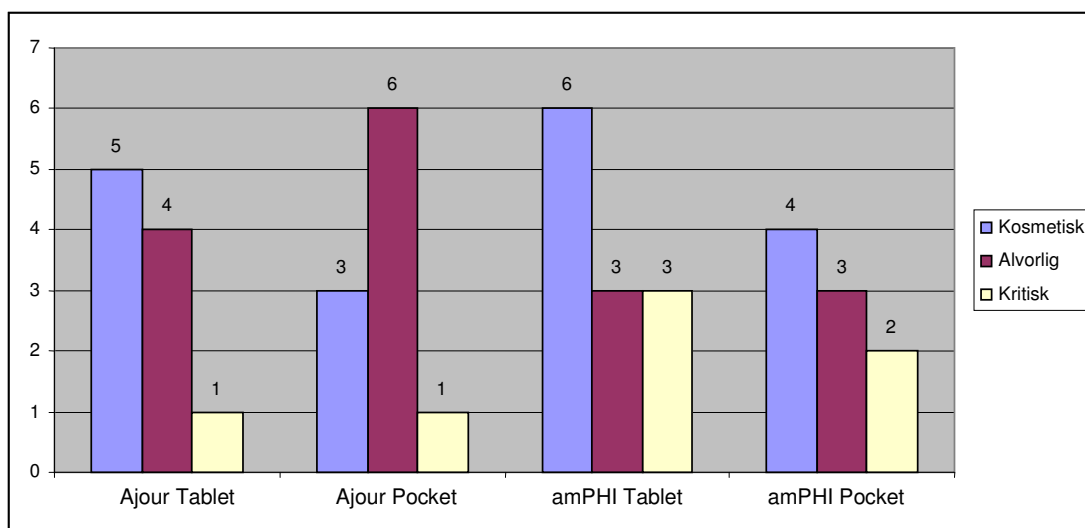
Figur 35 - Totalt antal oplevede problemer

Vi ser, ved at betragte antallet af oplevede brugbarhedsproblemer, at Ajour Pocket og amPHI Pocket, har henholdsvis 10 % og 25 % flere forekomster af problemer i brugen end amPHI Tablet, selvom amPHI Tablet har 20 % flere unikke brugbarhedsproblemer end Ajour Tablet og Pocket, og 33,3 % flere end amPHI Pocket. Dette indikerer for os, at man i brugen af små enheder i multiplatform systemer, selvom det konkrete indhold skulle være ækvivalent med indholdet på en større enhed, vil opleve flere tilfælde af problemer, end i brugen af større apparater. Dette underbygges desuden af, at de problemer, som er fundet på Pocket PC'erne, i gennemsnit er oplevet 2,3 gange på Ajour Pocket og 2,78 gange på amPHI Pocket i løbet af hele eksperimentet, hvor problemerne på Tablet PC'erne kun er oplevet 1,9 og 1,67 gange i gennemsnit. Vi ser altså, at selvom vi har fundet færre brugbarhedsproblemer på de små enheder, så har disse problemer hvert for sig været til større gene, end de problemer der blev fundet på Tablet PC'erne.

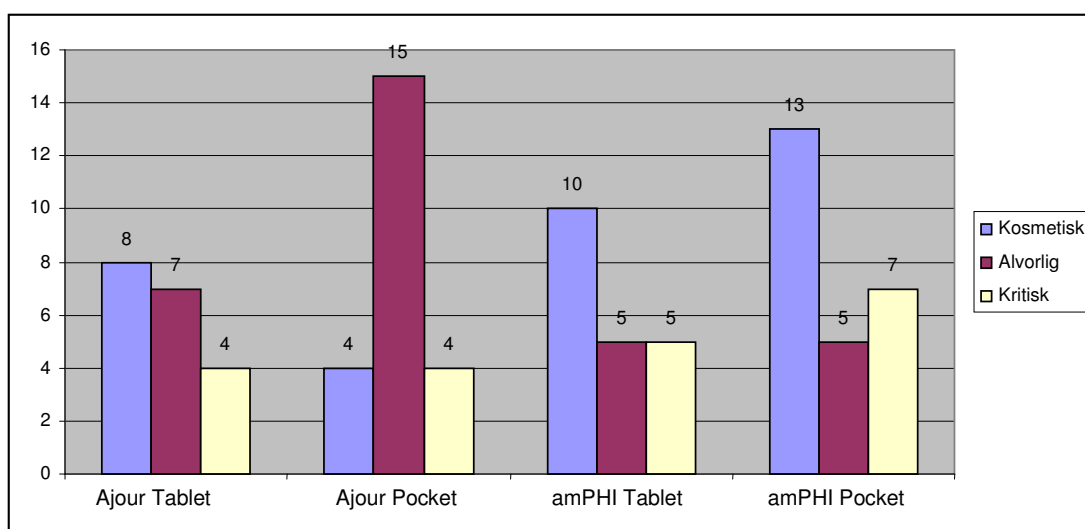
Antal brugbarhedsproblemer fordelt på apparater, opdelt efter kategori

Hvis man betragter Figur 36, kan det ses, at Judex' amPHI står for de fleste kritiske problemer, med 5 af 7. amPHI har desuden flest kosmetiske problemer (10 / 18), men kun 6 af 16 alvorlige problemer. Vi ser det mest interessante i dette, som værende den forholdsvis store mængde kritiske problemer i amPHI(71 %), og de mange alvorlige problemer i Ajour (62,5 %).

På Figur 36 kan man se, at de fleste kosmetiske problemer er at finde på Tablet PC'erne, med 11 af 18. Mens de andre typer fordeler sig mere jævnt, med 7 mod 9 i alvorlige problemer, og 4 mod 3 kritiske.



Figur 36 - Brugbarhedsproblemer på apparater, opdelt efter kategori



Figur 37 - Totalt antal oplevede problemer, opdelt efter kategori

Hvis man ser på Figur 37, kan man se, at de 2 kritiske problemer der blev fundet i Ajour, opleves hele 4 gange hver i løbet af eksperimentet. amPHI's 5 kritiske problemer bliver til sammenligning kun oplevet 2,4 gange hver i gennemsnit. Det samme gør sig til dels gældende for de alvorlige problemer, hvor testpersonerne i Ajour har oplevet hvert problem 2,2 gange, mens hvert problem kun er oplevet 1,67 gange i amPHI. Omvendt forholder det sig dog med de kosmetiske, hvor hvert problem i gennemsnit er oplevet 1,5 gange i Ajour, mod 2,3 gange i amPHI. Ud fra disse resultater kan vi se, at selvom Ajour ikke har mange kritiske problemer, så opleves disse ganske mange gange af testpersonerne, mens de alvorlige og kosmetiske problemer i gennemsnit ikke opleves ret mange gange i løbet af eksperimentet. Dette tyder på, at der er mange problemer som reelt ikke opleves

mere end en enkelt gang. Generelt er der ikke nogen signifikant forskel på antallet af oplevede problemer ($F_{3,8}=0,14$, $p=0,93$),

Hvis vi betragter Figur 36 og Figur 37, med henblik på at adskille de to typer platform ad, kan vi se, at der findes 4 kritiske problemer på Tablet PC'erne, mens der kun findes 3 på Pocket PC'erne, og at de 4 opleves 2,25 gange mens de 3 opleves hele 3,67 gange hver. De alvorlige problemer er der 7 af på Tablet PC'erne, mens der er 9 på Pocket PC'erne. Tablet PC'ernes problemer opleves kun 1,7 gange hver i gennemsnit, mens Pocket PC'ernes problemer opleves 2,22 gange. Tablet PC'erne har 11 kosmetiske problemer som sammenlagt opleves 18 gange(1,64), mens Pocket PC'erne har 7 som opleves 17 gange(2,43). Der er ikke nogen signifikant forskel på det gennemsnitlige antal gange de enkelte problemer er oplevet, apparaterne imellem ($F_{3,8}=0,65$, $p=0,61$).

Alt i alt kan vi her se, at selvom der findes flere problemer af en type, på en bestemt type platform, tyder intet i dette eksperiment på, at testpersonerne af den grund har flere problemer. Typisk for resultaterne er, at de steder hvor der er fundet få brugbarhedsproblemer, er disse problemer gået igen et tilsvarende større antal gange, således at testpersonerne har haft omtrent lige mange problemer med at bruge de forskellige apparater. Dermed kan vi sige, at der ikke er nogen signifikant forskel i antallet af brugbarhedsproblemer, de to platforme imellem.

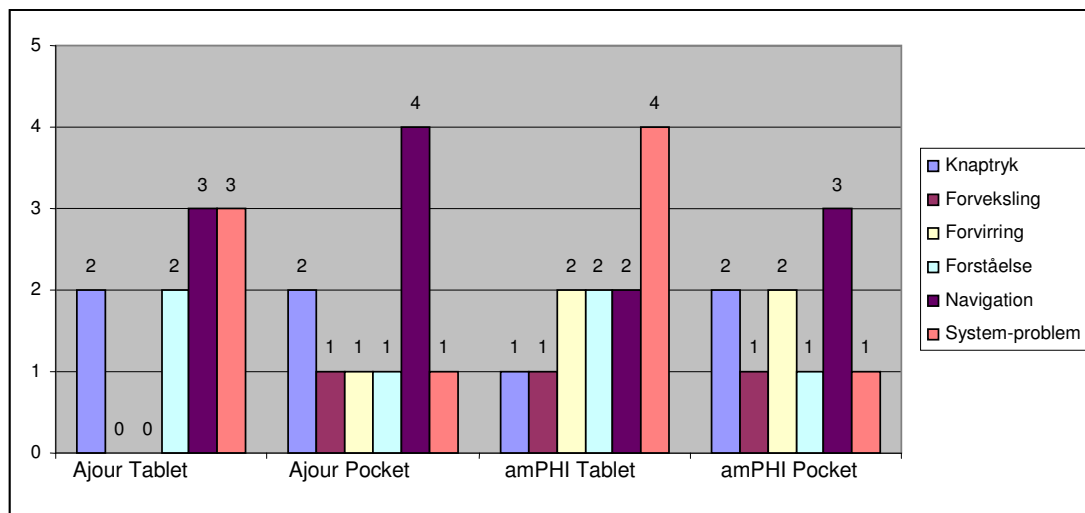
7.1.2 Typer af brugbarhedsproblemer

For at se, om der er nogen specielle former for brugbarhedsproblemer der er specielt påtrængende på en af de to typer platforme, som vi har benyttet os af i dette eksperiment, vil vi i dette afsnit lave en analyse og gruppering af de brugbarhedsproblemer der er fundet. Ud fra dette vil vi således kunne sige, om bestemte typer af problemer i udpræget grad tilhører en bestemt type apparat.

Vi har valgt at dele de fundne problemer op i følgende typer:

1. Knaptryk – Testpersonen trykker ikke på en knap som er nødvendig for opgavens udførelse. Typisk tror brugeren at opgaven er udført alligevel.
2. Forveksling – Testpersonen tror en knap eller et menu punkt er noget andet end det reelt er, hvilket kan betyde at han forsinkes i opgavens udførelse, eller løser denne forkert.
3. Forvirring – Testpersonen forvirres af systemets opbygning eller opførsel. Typisk bliver vedkommende i tvivl om hvad systemet har gjort, eller hvor i systemet han befinder sig.

4. Forståelse – Testpersonen har en anden forståelse af hvordan systemet fungerer, end den som udvikleren har haft da han lavede det.
5. Navigation – Testpersonen kan ikke finde den knap eller det menupunkt han leder efter.
6. System-problem – Systemet gør noget som er direkte forkert. Typisk dækker denne kategori over tilfælde hvor systemet opfører sig direkte forkert, eller ikke reagerer overhovedet.



Figur 38 - Typer af brugbarhedsproblemer, fordelt på apparater

Som det kan ses på Figur 38, er fordelingen af problemtyper ikke helt jævn på tværs af apparaterne. ”Knaptryk” - problemerne er rimeligt jævnt fordelt over apparaterne, hvor alle har 2 problemer, med undtagelse af amPHI Tablet, som kun har 1. Denne lighed i antallet af problemer ser vi som en indikator for, at amPHI generelt har fået de fleste af denne type problemer rettet, mens Ajour måske nok undgår nogle af de problemer som stadig er i amPHI, men til gengæld har visse ”børnesygdomme”, som er blevet rettet i amPHI.

Brugbarhedsproblemer som resulterer i, at testpersonen forveksler elementer i systemet, er der generelt ikke mange af. Apparaterne Ajour Pocket, amPHI Tablet og amPHI Pocket har hvert et enkelt eksemplar af disse. Vi ser to ting i disse problemer. For det første kan vi se, at Pocket PC’erne begge har et sådant problem. Dette oplever vi som et eksempel på, at brugerne af sådanne små enheder let kan komme til at misforstå hvilken del af et system de arbejder med. Dette giver os en ide om, hvor vigtigt det er at vise brugeren konteksten for det aktuelle skærbillede, så denne hele tiden har en forståelse af hvor i systemet vedkommende befinder sig. For det andet ser vi, at begge systemer i amPHI har et sådant problem. Dette kan give en ide om, at amPHI har problemer med at vise, hvad de enkelte dele af systemet helt præcist har som formål.

”Forvirring” er klart mest fremtrædende i amPHI. Ajour Pocket har et enkelt tilfælde, mens amPHI Tablet og Pocket har 2 tilfælde. Dette siger os, at der må være helt grundlæggende elementer i amPHI, som har betydning for, at brugerne bliver forvirrede, og dermed kommer til at lave fejltagelser, eller bliver forsinkede i udførslen af deres arbejde. At der også findes forvirring i testen af Ajour Pocket tyder på, at der også her er elementer, som forvirrer brugerne. Da selve indholdet i Ajour Tablet og Pocket er det samme, må denne forvirring enten skyldes den måde hvorpå navigationen er indrettet, eller selve det, at Ajour Pocket er en Pocket PC.

Brugbarhedsproblemer som omhandler testpersonernes forståelse af systemet, er jævnt fordelt på Ajour og amPHI, på en sådan måde, at der er 2 problemer på hver Tablet PC, og 1 problem på hver Pocket PC. Dette viser os, at der stadig er elementer i begge systemer, som er udviklet forkert i forhold til falckredderes reelle måde at tænke på. At fordelingen af problemer ikke er jævn på tværs af apparater kan undre, og skyldes, at testpersonerne har flere problemer med at bruge det ”fulde” onscreen tastatur, som findes på Tablet PC’erne, end med at bruge det begrænsede telefon-tastatur, som findes på Pocket PC’erne.

Navigations-problemerne er den gruppe af problemer, der er mest gennemgående for alle fire apparater⁴. Der er fundet et problem mere af denne type på hvert apparat i Ajour, end i de eksperimenter der involverede amPHI, ligesom begge Pocket PC’er har et problem mere end deres tilsvarende Tablet PC’er. Dette viser os, at testpersonerne generelt havde lettere ved at finde rundt i amPHI, end i Ajour, hvilket ikke er overraskende, da de kan siges at have en del øvelse i en tidligere version af amPHI, men da navigationen især i Tablet PC’erne er meget ens, siger det os også, at nogle af de bestræbelser vi har gjort os for at forsimple navigationen, ved at eliminere menupunkter, kun har betydet flere problemer for testpersonerne.

Vi ser desuden, at testpersonerne har sværere ved at navigere rundt i Pocket PC’erne. Dette ser vi som en indikation af, at disses navigation ikke er lige så god som på Tablet PC’erne, hvor samtlige menupunkter altid er synlige på skærmen. Forsøget med at lave et navigationshierarki på Ajour Pocket, som gjorde dele af navigationen synlig på skærmen, kan således ikke siges at have været mere anvendelig end den tilgang, med en hovedside med undersider, som bruges i amPHI Pocket.

⁴ se eventuelt bilag E, problem nr. 5, 6, 7 og 19

Der blev oplevet flere system-problemer på Tablet PC'erne end på Pocket PC'erne. Umiddelbart er dette svært at forklare, da denne type problemer ikke burde have nogen relation til platformstypen – interaktionselementerne er bygget tilsvarende op på de to typer apparat, og en fejl der opstod på en platform, burde have en rimelig sandsynlighed for også at opstå på en anden, indenfor det samme multiplatform system.

Der er ingen signifikant forskel i hvilke af denne type brugbarhedsproblemer der er fundet på de fire apparater ($F_{3,20}=1,28$, $p=0,94$).

7.1.3 Delkonklusion

I dette afsnit har vi set, at der i det eksperiment vi har foretaget med de fire apparater, er fundet i alt 27 forskellige brugbarhedsproblemer, og at der blev fundet flere problemer på amPHI Tablet, end på de andre apparater. Vi så desuden, at det ikke var i den del af eksperimentet, hvor amPHI Tablet blev benyttet, at testpersonerne oplevede det største totale antal problemer, men at dette i stedet var i brugen af Ajour Pocket og især amPHI Pocket. Ud fra dette kan vi konkludere, at selvom de små enheder i de aftestede multiplatform systemer i gennemsnit har haft færre brugbarhedsproblemer, har disse problemer været til større gene end problemerne på Tablet PC'erne, og dermed betydet at Pocket PC'erne har været mere problematiske at bruge. Der blev dog ikke fundet nogen signifikant forskel på apparaterne.

Ser man på problemkategorierne hver for sig, kan det ses, at hvert brugbarhedsproblem i gennemsnit opleves mellem 1,67 og 2,4 gange hver i løbet af hele eksperimentet. En enkelt undtagelse fra dette er dog, at de to kritiske problemer som opleves på Ajour, opleves hele fire gange hver, hvilket vil sige, at de i gennemsnit opleves af fire ud af seks testpersoner. Dette tager vi som en indikation af, at disse problemer er særligt kritiske. At de resterende problemer i gennemsnit kun fremkommer i under halvdelen af tilfældene, ser vi som en indikation af, at disse er mindre problematiske, da de fleste brugere således vil kunne bruge systemet, uden at støde ind i disse. Heller ikke her blev der fundet statistisk signifikans.

I analysen af problem-typer har vi set, at Ajour har en række "børnesygdomme", som ikke er til stede i amPHI. Vi kan se at amPHI-systemet generelt har brugbarhedsproblemer, hvor brugeren forvirres, og at begge Pocket PC'er har "forvirrings"-problemer. Vores tolkning af dette er, at amPHI ikke på tilstrækkelig vis giver brugeren overblik, eller at amPHI indeholder elementer, som direkte forvirrer brugeren, ligesom at forvirringen på Ajour's såvel som amPHI's Pocket PC må skyldes navigationens indretning, samt dårlig indikation af, hvilket menupunkt der er aktivt.

Vi kan se, ud fra vores resultater, at der var flest navigationsproblemer i Ajour, samt at Pocket PC'erne begge havde et problem mere end deres tilsvarende Tablet. Ud fra dette kan vi konkludere, at de forsøg vi har gjort os, for at forbedre og forsimple navigationen ikke har været virksomme, og i nogen tilfælde sandsynligvis har givet bagslag, da de strider imod brugernes mentale model af systemet. Vi kan desuden konkludere, at der generelt er flere problemer på Pocket PC'erne, hvilket kan skyldes den mangel på overblik, som brugeren har, når hele navigationen ikke kan vises, på samme måde som der er mulighed for på en Tablet PC. Alt i alt har vi set, at der er indikationer af, at der opleves flere problemer i brugen af Pocket PC'erne, dog uden at der er signifikant forskel på de to typer platform.

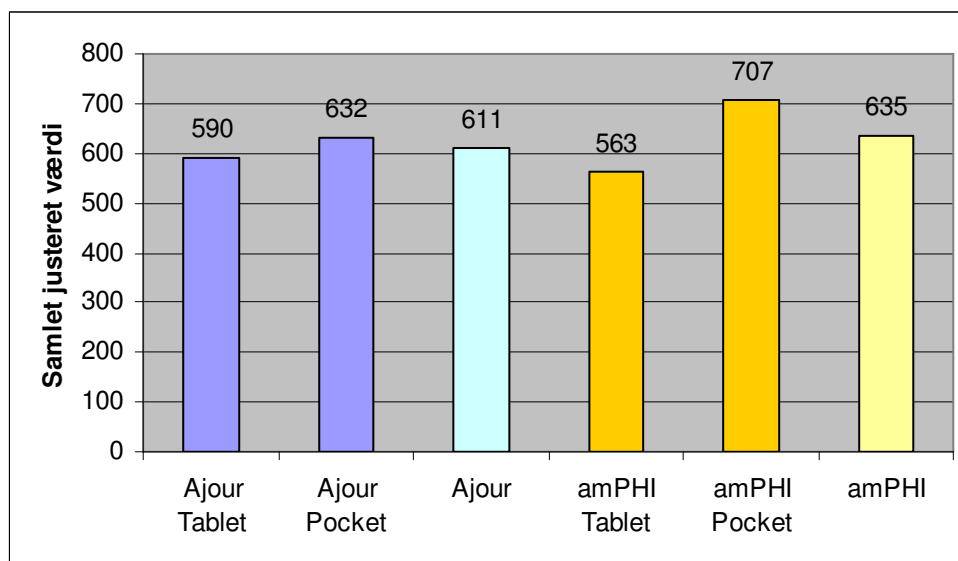
7.2 Arbejdsbelastning

I dette afsnit beskriver og analyserer vi de resultater, som vi har fået ud af NASA TLX arbejdsbelastningsmålingen. Resultaterne betragtes med hensyn til at sammenligne de forskellige apparater, platforme og systemer, og desuden kigges der på forskelle testpersonerne imellem. Endelig betragtes hver af arbejdsbelastningens enkelte faktorer for sig. Til sidst i afsnittet konkluderes der på de analyserede resultater.

7.2.1 Analyse

Samlet arbejdsbelastning

Forskellen på den samlede arbejdsbelastning for de fire apparater, som kan ses på Figur 39, er beskeden. amPHI Pocket har en arbejdsbelastning på 707, Ajour Pocket en på 632, Ajour Tablet en på 590 og amPHI Tablet en på 563. Arbejdsbelastning ved amPHI Pocket er således oplevet 10,6 procent større end Ajour Pocket, 16,6 procent større end Ajour Tablet og 20,4 procent større end amPHI Tablet. Denne forholdsvise beskedne forskel understøttes af ANOVA ($F_{3,10}=0,69$, $p=0,57$), som viser, at der ikke er signifikant forskel på de fire apparaters samlede arbejdsbelastninger.



Figur 39 - Gennemsnitlig samlet arbejdsbelastning per eksperiment

Vi kan se, at der er en vis sammenhæng mellem platform og arbejdsbelastning. Tablet PC'erne har begge en lavere arbejdsbelastning end deres tilsvarende Pocket PC'er. Således har Ajour Tablet en 4,8 % lavere arbejdsbelastning end Ajour Pocket, og amPHI Tablet har en arbejdsbelastning der er 20,4 % lavere end amPHI Pocket. Selv mellem de to apparater i amPHI, som har den største forskel, er der ikke signifikant forskel ($F_{1,10}=1,38$, $p=0,27$)

Det er endvidere interessant at se nærmere på apparat i forhold til platformstype. Gøres dette kan det ses, at arbejdsbelastning ved Ajour Tablet er 5,6 % større end ved amPHI Tablet. I forhold til Pocket PC'en er arbejdsbelastningen ved amPHI Pocket 10,6 % større end Ajour Pocket.

Ser vi på arbejdsbelastningen i forhold til system, kan vi se at amPHI, har en samlet gennemsnitlig arbejdsbelastning på 635, hvilket er 3,8 % større end Ajour, som har en samlet gennemsnitlig arbejdsbelastning på 611.

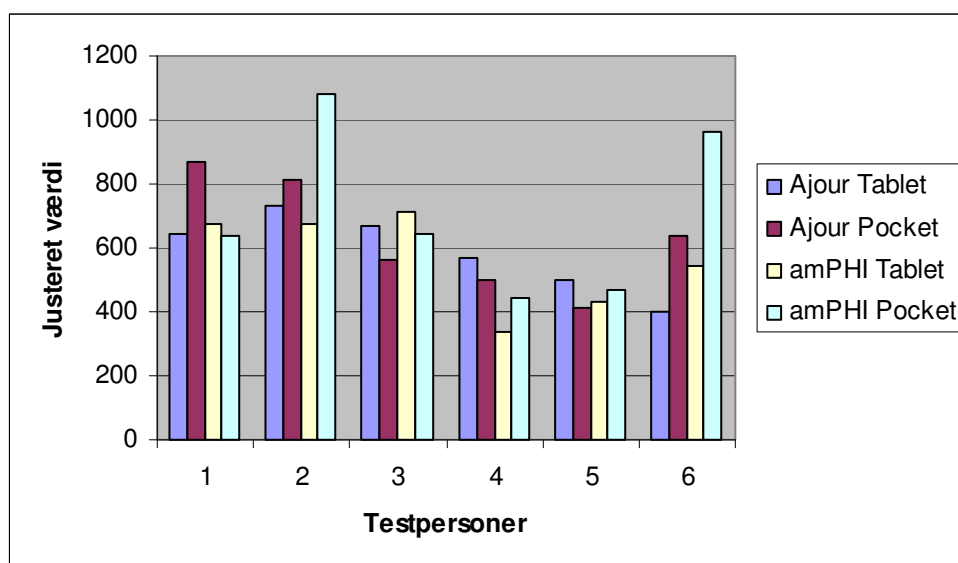
Arbejdsbelastning og testpersoner

Arbejdsbelastningen eksperimenterne imellem er som nævnt ganske ens, hvilket ikke umiddelbart giver noget klart billede af arbejdsbelastningens betydning for det enkelte apparat. Det er derfor interessant at se nærmere på den enkelte testpersons arbejdsbelastning i forhold til apparat, for derigennem at kunne få et klarere billede af arbejdsbelastningen. Ser vi først nærmere på arbejdsbelastningen for de seks testpersoner, kan vi ved hjælp af ANOVA ($F_{5,18}=4,138$, $p=0,011$) se, at testpersonernes oplevelse af arbejdsbelastningen er signifikant forskellig. Denne forskel er illustreret på Figur 40, hvor testpersonerne er opstillet i forhold til de fire apparater. Ud af denne

figur kan man umiddelbart se, at testperson 1, 2 og 3 oplever arbejdsbelastningen som værende generelt højere på tværs af apparaterne i forhold til de andre testpersoner. I den anden ende ser vi, at testperson 4 og 5 oplevede de forskellige tests som generelt mindre belastende at udføre.

I forhold til det enkelte apparat er det svært at sige noget entydigt, idet der er stor forskel på testpersonernes oplevelse af de forskellige eksperimenters indbyrdes relation. Af væsentlig udslag er dog værd at nævne amPHI Pocket, som for testperson 2 og 6 vægtes betydeligt højere end deres oplevelse af de tre andre apparater.

Ser vi nærmere på forholdet mellem platforme, kan vi se, at arbejdsbelastningen ved Tablet PC'erne generelt vægtes tilsvarende med enkelte undtagelser. Testperson 4 vægter arbejdsbelastningen ved Ajour Tablet som værende 40,4 % højere end ved amPHI Tablet, og testperson 6 vurderer arbejdsbelastningen ved amPHI Tablet til at være 26,6 % højere end ved Ajour Tablet. Ved test på Pocket PC'en er der stor variation af de enkelte testpersonernes vurdering af arbejdsbelastningen på de to apparater. Som allerede nævnt oplever testperson 2 og 6 et stor udslag i arbejdsbelastning for amPHI Pocket, hvor Ajour Pocket ikke har tilsvarende udslag.

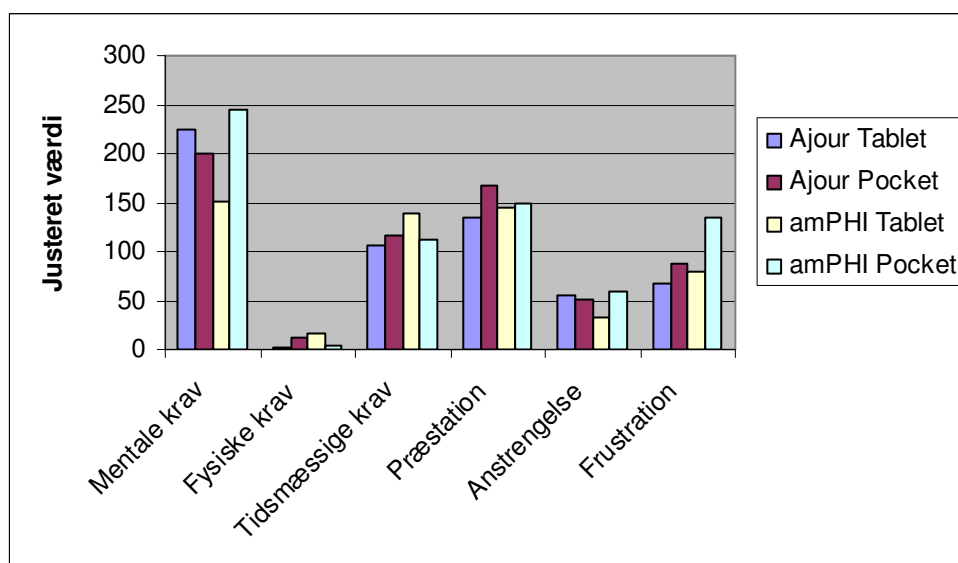


Figur 40 - Arbejdsbelastning per person og per eksperiment

Arbejdsbelastning og faktorer

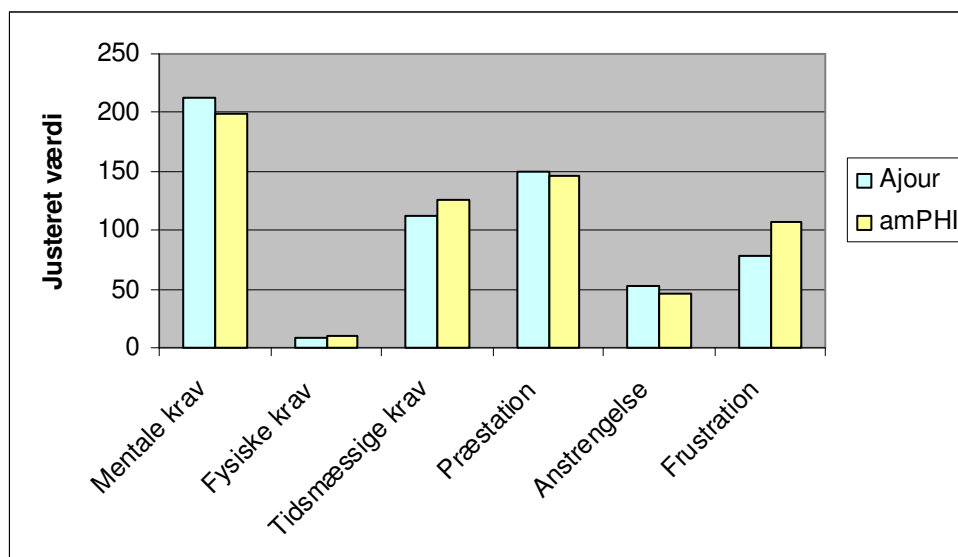
Arbejdsbelastningen i forhold til faktorer giver et mere nuanceret billede af arbejdsbelastningen i forhold til det enkelte apparat, hvilket fremgår af Figur 41. De mentale krav vægtes 61 % højere ved amPHI Pocket end ved amPHI Tablet. Den mentale belastning for amPHI Tablet vurderes generelt lavere end for de andre apparater, dog uden at dette er signifikant ($F_{3,20}=0,62$, $p=0,61$).

I forhold til de fysiske, tidsmæssige, præstationsmæssige og anstrengelsesmæssige faktorer er forskellen mellem apparaterne beskeden. Ved amPHI Pocket opleves frustration dog som havende betydelig mere indflydelse på arbejdsbelastningen end ved de tre andre apparater. amPHI Pocket opleves således som 34,8 % mere frustrerende end Ajour Pocket, 41,5 % mere frustrerende end amPHI Tablet og 49,6 % mere frustrerende end Ajour Tablet. Denne forskel er ikke signifikant ($F_{3,20}=0,45$, $p=0,72$).

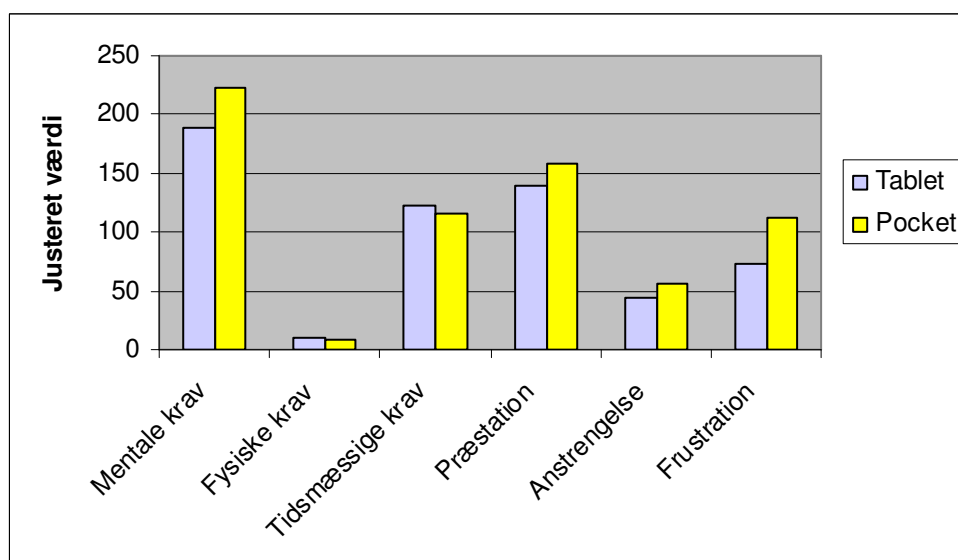


Figur 41 - Arbejdsbelastning i forhold til faktorer

Ser man på Figur 42, kan man se, at de mentale krav er lidt højere i brugen af Ajour end i amPHI, men dog næsten ens. De fysiske krav er vurderet meget lavt i relation til de andre faktorer, og er stort set ens for Ajour og amPHI. De tidsmæssige krav er ved Ajour vurderet 11,2 % lavere end ved amPHI. Testpersonernes præstation ved de to systemer er vurderet næsten ens, hvilket ligeledes er tilfældet ved anstrengelse. Dog er Ajour vurderet mere anstrengende at bruge end amPHI. Endelig er amPHI vurderet til at være 27,1 % mere frustrerende for testpersonerne at bruge end Ajour. En signifikansanalyse viser dog, at der ikke er nogen signifikant forskel på de to systemer ($F_{1,10}=0,007$, $p=0,93$).



Figur 42 - Arbejdsbelastning i forhold til system



Figur 43 - Arbejdsbelastning i forhold til platform

På Figur 43 kan man se, at der generelt ikke er nogen stor forskel i arbejdsbelastningen, platformene imellem, hvilket en signifikansanalyse også viser ($F_{1,10}=0,15$, $p=0,71$). Dog kan vi se, at Tablet PC'erne typisk har en lavere arbejdsbelastning end Pocket PC'erne, hvilket specielt kommer til udtryk ved frustration, som er 34,1 % større.

7.2.2 Delkonklusion

Vi har i dette afsnit set at den overordnede arbejdsbelastning for de fire apparater er forholdsvis ens. Dog kan vi se at arbejdsbelastningen for Pocket PC'erne er ligget på 632 og 707, hvor Tablet

PC'ernes ligger på 590 og 563, hvorfor vi kan se en indikation af, at det generelt er forbundet med en større arbejdsbelastning at benytte en Pocket PC til at udføre en given opgave, givet at henholdsvis Pocket PC og Tablet PC indeholder den samme funktionalitet.

Vi kan ved arbejdsbelastningen i forhold til de enkelte faktorer se, at testpersonerne har vægtet frustration højere for amPHI Pocket, end for de resterende tre apparater. Herved kan vi konkludere, at amPHI Pocket var betragtelig mere frustrerende at bruge end de andre apparater. Desuden kan vi se, at testpersonerne har vægtet de mentale krav højere ved amPHI Pocket og Ajour Tablet end ved de resterende to apparater. Foruden disse få udsving kan vi dog konkludere at testpersonerne ved de fleste faktorer har vægtet arbejdsbelastningen nogenlunde ens.

Generelt har vi kunnet se, at amPHI Pocket har en højere arbejdsbelastning end Ajour Pocket. I forhold til Tablet PC er det dog et andet billede der viser sig, da arbejdsbelastningen på amPHI Tablet er lavere end Ajour Tablet. Alt i alt kan vi se, at brugen af amPHI Tablet ikke er forbundet med nogen stor arbejdsbelastning. Ajour Tablet er lidt hårdere at bruge, mens Pocket PC'erne, og specielt amPHI Pocket er forbundet med den største belastning.

Vi kan konkludere, at der reelt ikke er nogen signifikant forskel i arbejdsbelastningen, de fire apparater imellem, men at der dog er indikationer for, at arbejdsbelastningen er større ved brug af Pocket PC'er, end ved brugen af Tablet PC'er, og at dette især udmønter sig i form af frustration. Vi kan desuden konkludere, at der ikke er nogen signifikant forskel i arbejdsbelastningen, Ajour og amPHI imellem.

7.3 Måling af usability

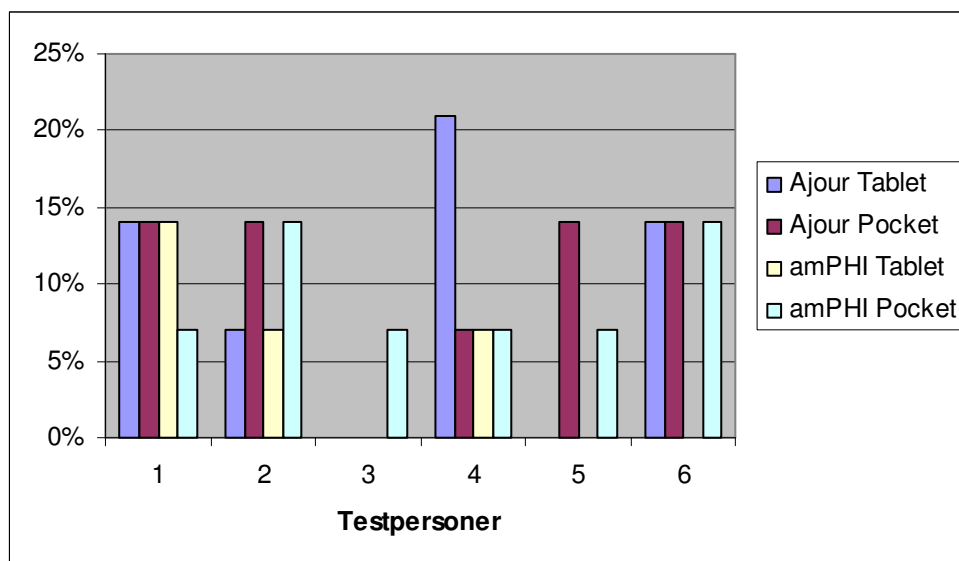
I dette afsnit beskrives først, hvordan databehandlingen fandt sted, med henblik på at undersøge formålstjenestelighed, effektivitet og tilfredshed for amPHI og Ajour. Dernæst analyseres resultaterne af denne undersøgelse i den nævnte rækkefølge, hvilket slutteligt leder hen mod en afsluttende konklusion på baggrund af de tre målingers resultater.

7.3.1 Analyse

Formålstjenstlighed

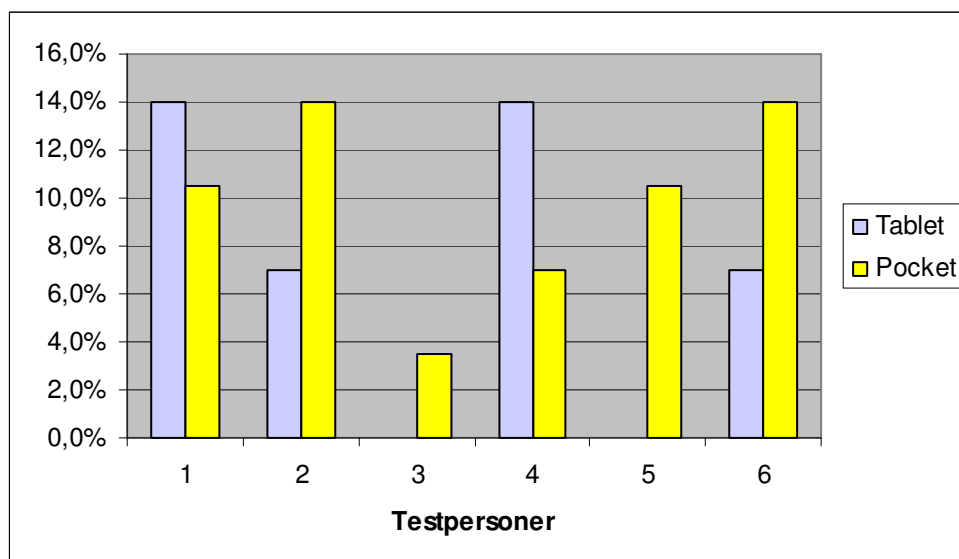
Vi vil i dette afsnit belyse, hvor præcist og fuldendt testpersonerne udførte eksperimenterne på de fire apparater. Dette gøres først med henblik på at sammenligne Tablet PC apparaterne med hinanden, samt at sammenligne Pocket PC apparaterne med hinanden. Dernæst sammenlignes Tablet PC'erne med Pocket PC'erne. Til sidst foretages en sammenligning mellem amPHI systemet og Ajour systemet.

På Figur 44 kan vi se testpersonernes procentvis ikke-gennemførte opgaver. Af denne figur kan vi se, at testpersonerne 4 og 6 gennemfører flere opgaver på amPHI Tablet end på Ajour Tablet. Ved de resterende testpersoner gennemføres et tilsvarende antal opgaver på både amPHI Tablet og Ajour Tablet. Ved sammenligning af Ajour Pocket og amPHI Pocket kan vi se at Ajour Pocket ved testperson 1 og 5 har flere ikke-gennemførte opgaver end amPHI Pocket, hvilket på tværs af alle testpersonerne gør at der løses beskedent flere opgaver på amPHI Pocket end på Ajour Pocket. Overordnet set er der flest ikke-gennemførte opgaver på Ajour Pocket med 10,5 %, dernæst kommer Ajour Tablet og amPHI Pocket med 9,3 % og til sidst amPHI Tablet med 4,7 % ikke-gennemførte opgaver. Hvis dette analyseres med ANOVA viser der sig et signifikansniveau på 39 %, hvilket ikke er signifikant ($F_{3,20}=1,06$, $p=0,39$).



Figur 44 - Procentvis ikke-gennemførte opgaver

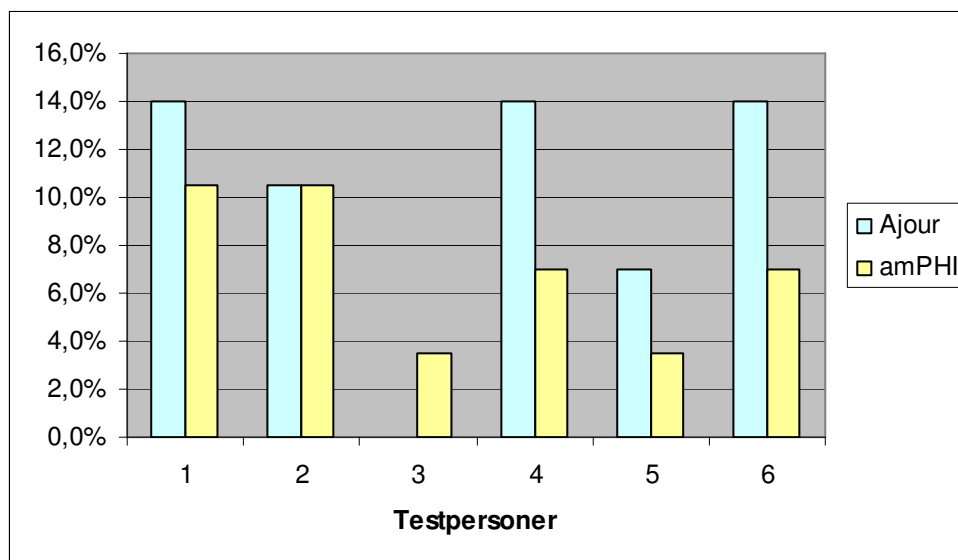
Ser vi nærmere på forholdet mellem Tablet PC kombinationen og Pocket PC kombinationen, kan vi på Figur 45 se en forskel, der dog ikke entydigt peger i en bestemt retning. Testpersonerne 1 og 4 gennemfører færrest opgaver ved Tablet PC'erne, hvor de resterende testpersoner gennemfører færrest opgaver ved Pocket PC'erne. Overordnet set er der en tendens til, at testpersonerne løser færre opgaver ved brug af Pocket PC apparaterne end ved brug af Tablet PC apparaterne.



Figur 45 – Procentvis ikke-gennemførte opgaver i forhold til platform

På Figur 46 er de procentvis ikke-gennemførte opgaver for amPHI og Ajour angivet. Der kan her ses en større mængde gennemførte opgaver med amPHI end med Ajour. Alle testpersoner med

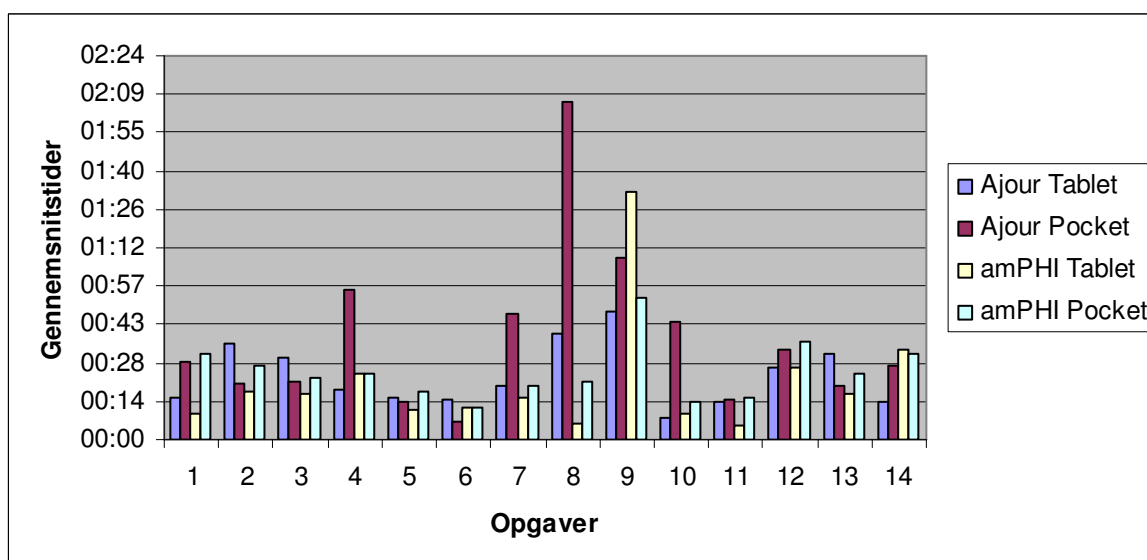
undtagelse af testperson 3, har flere eller lige mange ikke-gennemførte opgaver på Ajour som på amPHI. Gennemsnitligt gennemfører testpersonerne således ikke 9,9 % af opgaverne ved Ajour, hvor dette tal er nede på 7,0 % for amPHI apparaterne.



Figur 46 – Procentvis ikke-gennemførte opgaver i forhold til vores system

Effektivitet

I dette afsnit analyseres tiderne (se bilag F) der blev registreret ved gennemgang af videoerne. Først anskues gennemsnitstider for hvert apparat i forhold til de enkelte opgaver. Dernæst anskues tiderne i forhold til apparaternes platform, og slutteligt vil vi anskue tiderne i forhold til Ajour.



Figur 47 - Gennemsnitstider for opgaver

Figur 47 illustrerer gennemsnitstiderne for testpersonernes udførelse af opgaverne på de fire apparater. ANOVA ($F_{3,52}=2,001$, $p=0,125$) viser, at der ikke er signifikant forskel på gennemsnitstiderne for opgaveudførelsen på de fire apparater.

Forholdet mellem Tablet PC'erne viser en klar tendens mod, at opgaveudførelsen på amPHI Tablet er hurtigere end den tilsvarende opgaveudførelse på Ajour Tablet. I 9 ud af 14 opgaver er dette tilfældet, hvor forskellen dog oftest er begrænset. Enkelte store udsving er dog at finde ved opgaverne 8 og 9, hvor Ajour Tablet ved førstnævnte opgave tidsmæssigt lægger betragteligt højere end amPHI Tablet. Ved den 9. opgave er situationen dog lige omvendt da opgaven udføres betydeligt hurtigere på Ajour Tablet end på amPHI Tablet. Desuden er det værd bemærke at udførelsen af amPHI Tablet ved opgave 9 tidsmæssigt tager betydeligt længere end nogen anden opgave udført på amPHI Tablet.

Den samlede gennemsnitlige udførelse af opgaverne for hver af de fire apparaterne er forholdsvis ens for tre af disse fire. Den hurtigste gennemsnitlige opgaveudførelse sker på amPHI Tablet på 21 sekunder, hvilket er 12,5 % hurtigere end Ajour Tablet med en gennemsnits udførselstid på 24 sekunder, og 16 % hurtigere end amPHI Pocket med en gennemsnitlige udførselstid på 25 sekunder. Ajour Pocket har en væsentlig langsommere udførselstid end de andre med 38 sekunder, og er således 34,2 % langsommere end amPHI Pocket.

Ved at se nærmere på opgaverne er der enkelte der tidsmæssigt skiller sig ud fra de andre. Disse er opgave 4, 7, 8 og 10, hvor tiden for Ajour Pocket ligger betragteligt højere end for de resterende andre apparater. Det er endvidere specielt interessant at betragte Ajour Pocket i forhold til amPHI Pocket. Resultaterne af Figur 47 viser, som allerede nævnt, at Ajour Pocket ved 4 af opgaverne har massive udsving i forhold til de resterende apparater og herunder amPHI Pocket. Det er dog interessant, at Ajour Pocket i 9 af de resterende 10 opgaver udføres tidsmæssigt hurtigere end amPHI Pocket. Ved at betragte de opgaver, hvor der er store udsving, kan vi se en forklaring på, hvorfor disse udsving forekommer.

Opgave 4. Du tjekker den tilskadekomnes pupiller. Angiv at de begge er normale.

Apparat	Testperson	Problem nr.	Type	Gennemsnitstid (sek.)	Forsinkelse ift. gennemsnit
Ajour Tablet	2	5	Navigation	19	+21
Ajour Pocket	1	5	Navigation	56	+54
Ajour Pocket	2	13	Forveksling	56	+1
Ajour Pocket	6	5	Navigation	56	+58
amPHI Tablet	1	5	Navigation	25	+22
amPHI Tablet	2	5	Navigation	25	+34
amPHI Pocket	1	5	Navigation	25	+8
amPHI Pocket	3	16	System-problem	25	+5

Tabel 9 - Forsinkelse for problemer i opgave 4

I denne opgave kan vi, ved at betragte logfilerne for eksperimentet, se at testpersonerne overvejende oplever en bestemt type fejl. I Tabel 9 kan vi se, at brugbarhedsproblem nr. 5, som er et navigations-problem, er gennemgående på alle fire apparater, og skyld i 97 % af forsinkelserne i forhold til gennemsnittet. Som det kan ses, har især Ajour Pocket store forsinkelser i forbindelse med navigationsproblemer, med 54 og 58 sekunder mod 8 sekunder i amPHI Pocket. Tablet PC'erne bliver omtrent lige meget forsinket af de problemer der opleves på disse. Andre typer problemer udgør ikke nogen særlig stor forsinkelse. Det problem der skaber denne forsinkelse er problem nr. 5:

Problem 5. Tilstand ved ankomst: Problemer med at finde "Pupiller".

I Ajour, er de knapper, som er nødvendige for at fuldføre opgaven, placeret under tredje underside, i menupunktet "Tilstand ved ankomst". Dette er også den placering som bruges i amPHI Pocket, mens Tablet PC'erne har knapperne placeret umiddelbart på forsiden for "Tilstand ved ankomst". Vi ser den ekstra kompleksitet, som opstår som følge af at knapperne flyttes væk fra forsiden, som en væsentlig årsag til, at testpersonerne oplever dette problem, og forsinkes i så stærk grad.

Kørsel	An-komst	Skader	Beh. af Patient	Mål-inger
Beh. før ankomst				
Tilstand v. ankomst				
Respiration: Resp. stop Meget påvirket Let påvirket Upåvirket Cirkulation: Hjertestop Meget påvirket Let påvirket Upåvirket Blødning: Obs. indre Betyd. ydre Mindre ydre Ingen				
Næste...				

Kørsel	An-komst	Skader	Beh. af Patient	Mål-inger
Beh. før ankomst				
Tilstand v. ankomst				
Rytme: Vent. flimmer Vent. Takykardi A Rytmier Sinus Rytme A Systoli Øvrige symptomer: Lammelser Krampes Opkast				
Næste...				

Kørsel	An-komst	Skader	Beh. af Patient	Mål-inger
Beh. før ankomst				
Tilstand v. ankomst				
Venstre pupil: Normal Lille Stor Lysstiv Højre pupil: Normal Lille Stor Lysstiv				
Næste...				

Figur 48 - Ajour Pocket, indgang til "Pupiller"

Tilstand ved ankomst				
Vis Oversigt				
1	2	3		
Respiration: Resp. stop Meget påvirket Let påvirket U-påvirket Cirkulation: Hjerte-stop Meget påvirket Let påvirket U-påvirket Blødning: Obs. indre Betyd. ydre Mindre ydre Ingen				

Tilstand ved ankomst				
Vis Oversigt				
1	2	3		
Pupil Højre: Lysstiv Stor Lille Normal Pupil Venstre: Lysstiv Stor Lille Normal Øvrige symptomer: Lam-melser Krampes Opkast				

Figur 49 - amPHI Pocket, indgang til "Pupiller"

Som det kan ses på Figur 48 og Figur 49, er der med den hierarkiske navigation op til fire navigations-spring ind til "pupiller" på Ajour Pocket, mod to på amPHI Pocket. På Tablet PC'erne er "pupiller" altid kun ét tastetryk væk.

Opgave 7. Brug traumeskemaet til at score patienten –
angiv at han er bevidstløs, og at han har et lukket brud i benet.

Apparat	Testperson	Problem nr.	Type	Gennemsnitstid (sek.)	Forsinkelse ift. gennemsnit
Ajour Pocket	1	19	Navigation	47	+67

Tabel 10 - Forsinkelse for problemer i opgave 7

I Tabel 10 kan vi se, at der kun er fundet et enkelt problem i opgave 7. Dette er igen et navigationsproblem, som har været skyld i en betragtelig forsinkelse. Problemet opstår i lighed med problem nr. 5, fordi testpersonen ikke kan finde et menupunkt, om end dette kun ligger 1-2 spring fra ethvert andet punkt. Vi antager, at grunden til at testpersonerne generelt løste denne opgave langsommere i Ajour end i amPHI var, at testpersonerne var vant til at finde ”Traumeskema” som det sidste menupunkt, mens det i Ajour var placeret som et underpunkt til ”Skader”, i midten af navigationsbjælken.

Opgave 8. Da du ankommer til stedet fortæller datteren
at hendes far før har haft problemer med hjertet. Angiv dette i systemet.

Apparat	Testperson	Problem nr.	Type	Gennemsnitstid (sek.)	Forsinkelse ift. gennemsnit
Ajour Tablet	3	7	Navigation	40	-16
Ajour Tablet	5	16	System-problem	40	+65
Ajour Pocket	1	7	Navigation	127	+74
Ajour Pocket	2	7	Navigation	127	+36
Ajour Pocket	4	7	Navigation	127	+68
Ajour Pocket	6	7	Navigation	127	+57
amPHI Pocket	2	7	Navigation	22	+79
amPHI Pocket	6	7	Navigation	22	-12

Tabel 11 - Forsinkelse for problemer i opgave 8

Som i de foregående tabeller, kan det i Tabel 11 ses, at langt de fleste problemer er navigationsproblemer. Igen kan det ses, at Pocket PC’erne har store forsinkelser som følge af navigationsproblemer. Især kan det ses, at fire personer har store forsinkelser i brugen af Ajour Pocket, mens kun en enkelt har det ved amPHI Pocket. Testperson 6 oplever endda dette problem, men gennemfører alligevel opgaven på 10 sekunder. Problem nr. 7 opstår, når brugerne ikke kan finde et sted at angive ”Hjerte/kredsløb” som ”Tidligere Sygdom”. I amPHI er ”Tidligere Sygdom” et menupunkt i sig selv, hvilket er en direkte udledning af den originale ambulance-journal, mens det i Ajour er placeret som en underside til ”Patient”. Denne afvigelse fra testpersonernes mentale model af systemet skabte således store forsinkelser i deres brug.

Opgave 9. Patienten har taget medicin (nitroglycerin) i form af piller,
men de har ikke hjulpet. Angiv dette i systemet.

Apparat	Testperson	Problem nr.	Type	Gennemsnitstid (sek.)	Forsinkelse ift. gennemsnit
Ajour Tablet	2	6	Navigation	48	+34
Ajour Pocket	1	6	Navigation	68	+59
Ajour Pocket	2	12	Forvirring	68	+48
Ajour Pocket	6	6	Navigation	68	+40
amPHI Tablet	1	6,3	Navigation, Forveksling	93	+7
amPHI Tablet	2	6,4	Navigation, Forvirring	93	+15
amPHI Tablet	5	23	Forvirring	93	-16
amPHI Tablet	6	4,14	Forvirring, System-problem	93	+141
amPHI Pocket	2	6	Navigation	53	+67
amPHI Pocket	3	16	System-problem	53	-30
amPHI Pocket	4	26,16	Forveksling, System-problem	53	-18
amPHI Pocket	5	6	Navigation	53	+60
amPHI Pocket	6	6	Navigation	53	-34

Tabel 12 - Forsinkelse for problemer i opgave 9

I Tabel 12 kan det ses, at det i opgave 9 igen er et navigationsproblem der er dominerende. Endvidere kan det ses at Ajour Tablet's gennemsnitstid er på 48 sekunder og således er væsentlig hurtigere end gennemsnitstiden for amPHI Tablet på 93 sekunder. Både for Ajour og amPHI gælder det dog at de oplevede problemer ikke har nogen markant indflydelse på den samlede gennemsnitstid, dog med undtagelse af testperson 6 som ved amPHI oplever en forsinkelse på 141 sekunder. Ved at betragte log filen for testperson 6, kan vi dog se at den store forsinkelse hovedsageligt skyldes "Forvirring" problemet⁵.

Opgave 10. Du observerer, at patienten har vejtrækningsproblemer.

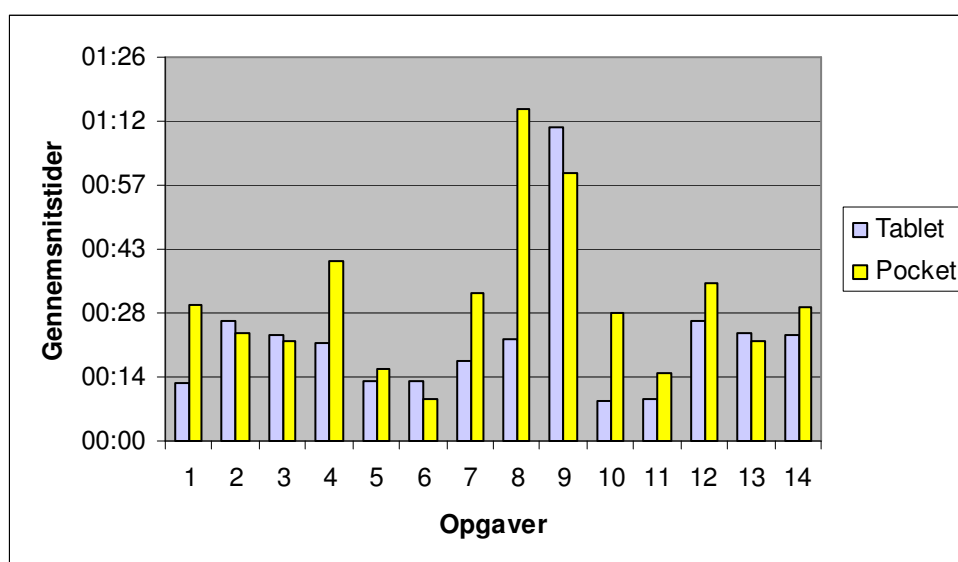
Angiv i systemet, at respiration og cirkulation er meget påvirkede.

Apparat	Testperson	Problem nr.	Type	Gennemsnitstid (sek.)	Forsinkelse ift. gennemsnit
Ajour Pocket	2	12	Forvirring	44	+51

Tabel 13 - Forsinkelse for problemer i opgave 10

⁵ Se Bilag F, Apparat C, Testperon 6, Del 2, opgave 2.

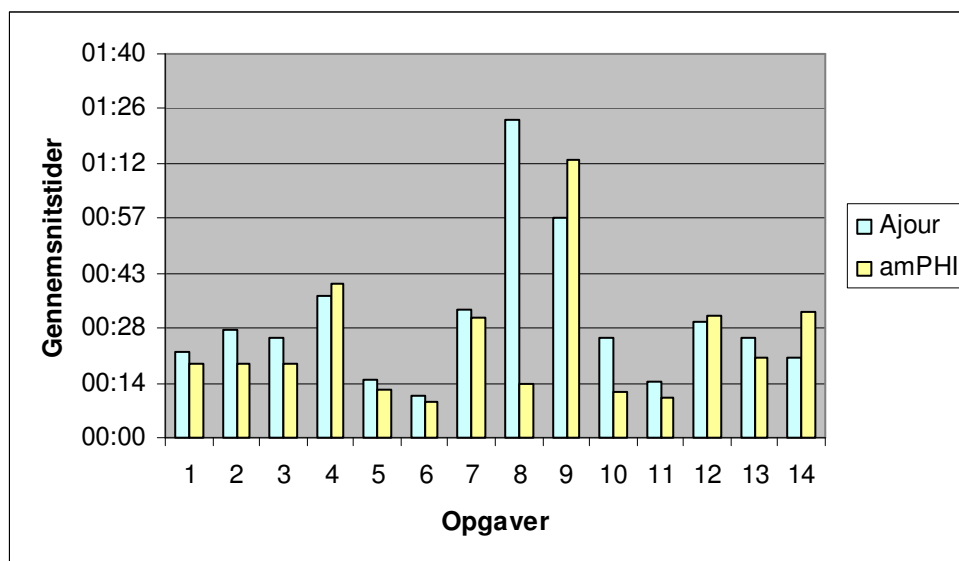
På Tabel 13 kan vi se at den observerede afvigelse tilsyneladende kun skyldes et enkelt problem. Dette problem omhandler dog forvirring omkring hvilket menupunkt der er aktivt, og kan dårligt forklare det observerede tidsforbrug. Vi tilskriver det derfor generelle problemer med at finde frem til de nødvendige knapper, uden at vi dog har klassificeret disse problemer som brugbarhedsproblemer.



Figur 50 - Gennemsnitstider for opgaver i forhold til platform

Figur 50 viser gennemsnitstiderne for opgaveudførelsen i forhold til typen af platform, og der viser sig her en klar tendens. Opgaver på Tablet PC'erne udføres enten stort set tilsvarende hurtigt eller betragteligt hurtigere end opgaveudførelsen på Pocket PC'erne. En enkelt undtagelse er der dog at finde, hvilket understøttes af tidligere resultater i dette afsnit, og dette er opgave 9. I dette tilfælde er opgaveudførelsen synligt hurtigere på Pocket PC'erne end den er på Tablet PC'erne, hvilket tallene også understøtter da opgave 9 på Tablet PC'erne udføres 14,3 % langsommere end på Pocket PC'erne. Den gennemsnitlige udførelses tid for Tablet PC'erne er på 22 sekunder, hvilket er 29 % hurtigere end den tilsvarende opgaveudførelse på Pocket PC'erne med 31 sekunder.

Figur 51 viser gennemsnitstiderne for opgaveudførelsen på henholdsvis Ajour og amPHI. Generelt set er opgaveudførelsen rent tidsmæssigt forholdsvis ens de to kombinationer imellem. Således lægger de to kombinationer sig tæt op af hinanden ved 11 af de 14 opgaver.



Figur 51 - Gennemsnitstider for opgaver i forhold til vores system

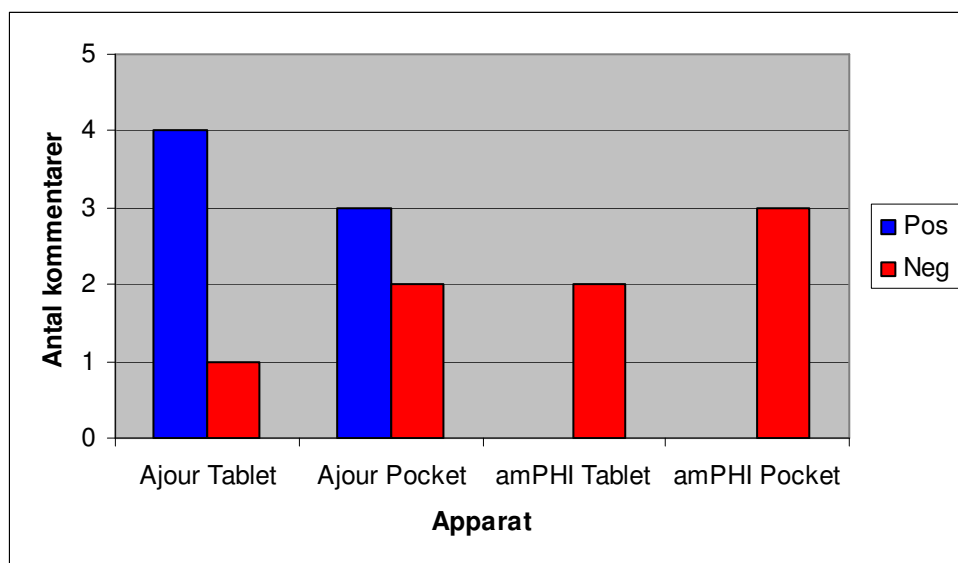
Opgaverne 8 og 9 skiller sig dog markant ud fra de andre, da udførelsen af opgave 9 på Ajour og amPHI er betragteligt langsommere end ved de resterende opgaver. Opgave 9 udføres dog 20,6 % hurtigere på Ajour end på amPHI. På tilsvarende vis skiller opgave 8 sig ud, dog er det kun Ajour, som her har et massivt udsving. I Ajour bruger testpersonerne i gennemsnit 1 minut og 23 sekunder på opgaven, hvor de i amPHI kun bruger 14 sekunder. Således tager Ajour hele 5,93 gange længere tid end amPHI. Den samlede gennemsnitlige udførselstid for Ajour er på 31 sekunder/opgave, hvilket er 22,6 % mere end amPHI's udførselstid på 24 sekunder/opgave. Havde Ajour ikke haft det massive udsving ved opgave 8 og haft en udførselstid svarende til amPHI's, ville testpersonerne kun have brugt 7,7 % længere tid med Ajour end med amPHI.

Tilfredshed

Kommentarer omkring systemerne blev registreret på baggrund af videogennemgangen af de seks testpersoners eksperimentforløb, samt i de efterfølgende interviews. I dette afsnit ønsker vi at belyse hvilken slags kommentarer, der blev ytret i forhold til hvilke apparater. Vi har således valgt at kategorisere den enkelte kommentar som enten værende positiv eller negativ i forhold til apparaterne. Vi har valgt ikke at tillægge disse resultater nogen særlig betydning, da testpersonerne for det første udmærket var klar over at Ajour var udviklet af os, samtidig med at de havde et tidligere forhold til amPHI, som givetvis også påvirkede deres holdning til dette.

Figur 52 illustrerer antallet af positive og negative kommentarer i forhold til apparat. I forhold til platform gives der ved Ajour Tablet 4 positive og 1 negativ kommentar, hvor der ved amPHI Tablet 2 negative kommentar gives og ingen positive. På tilsvarende vis gives der ved amPHI Pocket flere negative kommentarer end ved Ajour Pocket, og hvor der ved Ajour Pocket gives 3 positive kommentarer gives der ingen ved amPHI Pocket. Hvis man sætter fokus på de positive kommentarer⁶, kan man se at de hovedsageligt består i, at der ved Ajour apparaterne er gået nye veje i forhold til brugergrænsefladens opbygning. Dette kommer blandt andet til udtryk i kommentar 6 og 10, der omhandler at der ved Ajour, er benyttet en anderledes måde at overskue målinger på, samt at falckredderne ikke længere skal angive ”myndighed” før det bliver muligt at trykke på ”Morfin I.V.”.

På tilsvarende vis har kommentar 13, som er klassificeret som negativ kommentar under amPHI Tablet, en antydning af at måden, hvorpå ”Morfin I.V.” angives på, er besværlig og ”bøvlet”.



Figur 52 - Positive og negative kommentarer i forhold til apparat

⁶ Se kommentarer i bilag G

I forhold til typen af platform gives der flere positive kommentarer til Tablet PC'erne end til Pocket PC'erne. Endvidere gives der flere negative kommentarer til Pocket PC'erne end til Tablet PC'erne. Overordnet set er der en markant forskel i antallet af kommentarer i forhold til amPHI og Ajour. Der blev under eksperimenterne givet 7 positive kommentarer til Ajour, hvor der blev registreret 0 positive til amPHI. I forhold til negative kommentarer er forskellen mindre, da der ved Ajour blev angivet 3 negative kommentarer, hvor der blev angivet 5 negative kommentarer til amPHI.

7.3.2 Delkonklusion

Vi kan se, at færre af de stillede opgaver løses, i de dele af eksperimentet der involverer Ajour. Dermed siger resultaterne, at Ajour Pocket er mindst formålstjenstlig, med 10,5 % ikke-gennemførte opgaver, efterfulgt af Ajour Tablet og amPHI Pocket, begge med 9,3 % ikke gennemførte opgaver, og endelig amPHI Tablet, hvor kun 4,7 % ikke gennemføre. Dette betyder, at Ajour generelt er mindre formålstjenstligt, mens den forholdsvis høje gennemførselsprocent på amPHI Tablet betyder, at Tablet PC'erne i gennemsnit er mere formålstjenstlige end Pocket PC'erne. Der var dog ikke nogen signifikant forskel på apparaterne.

Vi kan se at udførelsen af opgaver på Ajour Pocket tidsmæssigt tager længere tid at udføre end amPHI Pocket. Ved fire af opgaverne tager det markant længere tid for testpersonerne at udføre opgaverne på Ajour end på amPHI, hvor den dog ved 9 af de resterende 10 opgaver udføres tidsmæssigt hurtigere end amPHI Pocket. Vi kan således konkludere, at hvor Ajour Pocket PC ved en stor del af opgaverne tidsmæssigt er hurtigere end amPHI Pocket, er der dog opgaver, hvor testpersonerne har store problemer med at gennemføre opgaverne tilsvarende hurtigt som de kan udføre dem på amPHI Pocket. Vi kan endvidere se at 11 ud af de 14 opgaver udføres opgaverne på Ajour tidsmæssigt tilsvarende til amPHI. Ved de resterende tre opgaver er der dog store tidsmæssige udsving, hvorved vi kan konkludere, at opgaverne på amPHI udføres hurtigere end på Ajour, dog stadig uden at der er en signifikant forskel i tiderne.

Vi har ved nærmere analyse af brugbarhedsproblemerne bag de observerede forsinkelser, set at navigationsproblemer i høj grad er årsagen til disse. Denne analyse viste klart, at specielt Ajour Pocket havde navigationsproblemer, som sinkede testpersonerne. Vi så også, at testpersonerne i nogle af de opgaver hvor der opstod brugbarhedsproblemer, generelt var langsomme til at finde de nødvendige menupunkter, hvilket underbyggede ideen om, navigationsproblemer var skyld i de

fleste forsinkelser. Endelig så vi at navigationsproblemer primært forsinkede brugerne på Pocket PC'erne.

Ud fra disse ting kan vi konkludere, at de to systemer ikke havde nogen signifikante forskelle i målingerne af deres brugbarhed. Vi kan også konkludere, at vi i udviklingen af navigationen på Ajour har afvejet fra brugernes mentale model af, hvorledes systemet bør fungere, og derved har skabt brugbarhedsproblemer.

8 Konklusion, begrænsninger og videre arbejde

I dette kapitel præsenterer vi vores svar på problemformuleringens tre delspørgsmål, for således at kunne svare på dens hovedspørgsmål. Dernæst betragter vi de begrænsninger der har været igennem projektet, for til sidst at komme med vores overvejelser omkring eventuelt videre arbejde indenfor de emner vi har berørt.

8.1 Konklusion

I dette afsnit svarer vi på de spørgsmål, som vi har rejst i vores problemformulering. Vi vil først svare på de tre delspørgsmål, ud fra de erfaringer, som vi har gjort os igennem projektet, samt de resultater som vi har fået ud af eksperimentet med Ajour og amPHI. Ved hjælp af disse konklusioner vil vi dernæst svare på problemformuleringens hovedspørgsmål.

Delspørgsmål 1

- *Hvad eksisterer i litteraturen omkring det at lave brugergrænseflader, som skaber og fastholder usability på multiplatform systemer?*

I vores litteraturstudie har vi set, at begrebet ”MUI” er defineret som ”et aktivt system, som giver adgang til information og ydelser fra forskellige platforme”. Vi har set, at et sådant multiplatform system skal kunne bruges til at løse sammenhængende opgaver på en række forskellige apparater, samt at det er væsentligt at disse apparater har samme ”look-and-feel”. Desuden har vi set, at mens den almindelige PC-platform kan kaldes relativt ubegrænset, er der restriktioner man skal være opmærksom på, på små apparater.

Vi har valgt at lægge fokus på metoden ”Design by graceful degradation”. Denne tillader, at man bruger en række transformationsregler til at overføre en brugergrænseflade fra et ”ubegrænset” til et mere begrænset interface. Endvidere har vi i denne forbindelse set, at disse regler kan prioriteres, således at der skabes mindst mulig diskontinuitet. Endelig har litteraturstudiet vist os, at begrebet ”videnskontinuitet” kan bruges til at sikre, at de forskellige apparater i et multiplatform system alle svarer til brugerens mentale model af det arbejde der skal udføres.

Vi kan konkludere, at der findes en del arbejder, som omhandler det problematiske i at lave multiplatform systemer, og brugergrænseflader dertil. Vi kan desuden konkludere, at der findes litteratur omkring det at overføre et interface fra et apparat til en andet, og om hvilken betydning kontinuitet i brugergrænsefladerne, apparaterne imellem, har for brugerens mentale model af systemet.

Delspørgsmål 2

- *Hvorledes kan vi udvikle en brugergrænseflade til et multiplatform system?*
 - *Hvilken betydning har valget af udviklingsmetode?*

I udviklingen af Ajour har vi set, at Contextual Design som metode til undersøgelse af problemområdet, har været fordelagtig idet, at den har fastholdt et fokus på, hvilke opgaver systemet skal løse, samt hvornår og hvordan brugerne har brug for systemet. Contextual Design har desuden betydet, at vi har haft en meget klar fornemmelse af, hvilke problemer der ville være i at udvikle Ajour, qua vores ufuldstændige forståelse af falckreddernes arbejde. Dette har betydet, at huller i vores viden er blevet udfyldt inden disse nåede at blive hæmmende for udviklingen. Contextual Design har med sine kreative værktøjer, sikret at alle design-ideer kom frem og blev diskuteret, men samtidig har metoden krævet at der blev lavet en del arbejde, som i grupper med flere udviklere ville sikre, at implicit viden blev delt, men som i vores tilfælde blot virkede som spild af tid.

Vi har set, at det har en væsentlig betydning for de brugergrænseflader der bliver udviklet, i hvilken rækkefølge disse laves. Vi har oplevet, at det er nødvendigt, at lave iterative bevægelser mellem platformene, hvis der skal være konsistens i disses opførsel. Graceful Degradation transformationsreglerne sikrer at objekter i en brugergrænseflade overføres på den bedst mulige måde til et andet apparat, men vi har set det som en nødvendighed, at refakturere tilbage fra Pocket PC til Tablet PC, da Pocket PC'en har begrænsninger, der ellers ville betyde nedsat kontinuitet i brugergrænsefladen, på tværs af platformene.

Vi kan konkludere, at vores valg af metodisk tilgang til både undersøgelse af problemområdet, og til udvikling af brugergrænsefladen, har givet væsentlige input til det endelige design af vores multiplatform system, Ajour. Vi kan desuden konkludere, at Contextual Design ikke bør bruges uden omtanke, da dele af metoden havde omkostninger i form af tidsforbrug, som ikke stod mål med resultaterne. Således betragter vi store dele af Contextual Design som havende et fokus på

vidensdeling, hvilket ikke er praktisk i et team på to mand. Omvendt kan vi i forbindelse med Contextual Design konkludere, at de kreativitets-fokuserede dele af metoden kan være meget givende, selv for et meget lille hold af udviklere, på trods af at det kan være svært at få det fulde udbytte af aktiviteterne hvis man er mindre end fire personer til aktiviteter som bygning af affinity, hvor et aktivt samspil mellem udviklerne er vigtigt.

Delspørgsmål 3

- ❑ *Er der problemer i forhold til brugbarhed, som adskiller den ene platform i et multiplatform system fra den anden?*

Hvilke typer problemer er typiske for henholdsvis. Tablet PC'er og Pocket PC'er, og hvad er grunden til at disse problemer opstår, på den specifikke type af platform?

I resultaterne af det eksperiment vi har foretaget med Ajour og amPHI, har vi set, at der ikke er nogen signifikant forskel i antallet af brugbarhedsproblemer, ligesom der ikke er nogen signifikant forskel i det totale antal gange testpersonerne har oplevet problemer.

Vi har set, at Pocket PC'erne typisk har mange navigationsproblemer, som forsinker brugerne meget. Navigationsproblemer opstår også i en vis grad på Tablet PC'erne, men forsinker ikke brugerne i samme grad. Vi har desuden set, at Tablet PC'erne har mange systemproblemer i forhold til Pocket PC'erne, om end vi ikke har kunnet finde nogen god forklaring på dette fænomen. Endelig har vi set, at arbejdsbelastningen er størst i brugen af Pocket PC'erne, og at Tablet PC'erne er mest formålstjenstlige, dog uden at disse resultater er signifikant forskellige.

Vi kan konkludere, at der ikke har været nogen målbart signifikant forskel på de to platforme. Vi kan dog også konkludere, at det særligt er navigationsproblemer på Pocket PC'erne, der forsinker brugen af disse meget.

Hvilke forskelle og ligheder er der mellem vores system og amPHI, og hvorfor opstår disse?

I vores resultater har vi set, at der er omtrent lige mange brugbarhedsproblemer i Ajour og amPHI, og at disse problemer typemæssigt fordeler sig jævnt på de to systemer, med undtagelse af navigationsproblemer, som overvejende er at finde i Ajour, og forvirringsproblemer, som amPHI har fire gange så mange af, som Ajour.

Vi har desuden set, at amPHI Pocket har højere arbejdsbelastning end Ajour Pocket, mens Ajour Tablet har højere arbejdsbelastning end amPHI Tablet, uden at disse resultater dog er signifikante. I resultaterne kan vi se, at Ajour er mindre formålstjenstlig og tidsmæssigt langsommere end amPHI, hvor navigationsproblemerne især forsinker brugen af Ajour Pocket.

Ud fra disse resultater kan vi konkludere, at der ikke er nogen målbart signifikant forskel på Ajour og amPHI. Vi kan dog sige, at den lave effektivitet på Ajour Pocket skyldes svære navigationsproblemer.

Er der en sammenhæng mellem de fundne problemer, og de emner der behandles i litteraturen?

Ved at betragte resultaterne af vores eksperiment kan vi se, at der er dele af vores resultater, som hænger direkte sammen med de emner der behandles i litteraturen, mens der er dele af resultaterne, som ikke behandles i litteraturen. Det er tydeligt for os, at vi ved at bryde med brugerens mentale model, skaber en mængde problemer i brugen af Ajour, fordi brugerne har en forventning om hvorledes systemet bør være udformet, som ikke bliver indfriet.

I udviklingen har vi ikke fundet en sammenhæng mellem navigation på Pocket PC'er, og den litteratur, som har ligget bag mange af de design-beslutninger, som vi har taget i forbindelse med at overføre Ajour fra Tablet PC'en til Pocket PC'en. Vi kan dog se, at den radikale ændring af navigationsformen, som vi har indført på Pocket PC'en, kan ses som en form for transformation af et grafisk element, hvorved vi regrupperer eller deler interaktører, og dermed skaber et højt diskontinuitetsniveau.

Vi kan konkludere, at de problemer vi har oplevet i Ajour, omkring navigation og brugernes forståelse af systemet, i vid udstrækning er dækket af [Flo04] og [anonymous]. Vi kan også konkludere, at begrebet navigation, som direkte har stor betydning for brugbarheden af Ajour Pocket, kun implicit behandles i den litteratur vi har undersøgt, da navigationens elementer indgår i de almene Graceful Degradation regler, mens selve begrebet navigation ikke behandles eksplicit.

Hovedspørgsmål

- *Hvilke problemer er der i at skabe og fastholde brugbarhed i multiplatform systemer?*

Igennem dette projekt har vi set, at der er lavet forskning i at overføre et design fra en platform til en anden. Denne forskning fokuserer meget på at sikre, at designet er kontinuert, således at brugeren kan bruge sin viden om systemet fra en platform, til at bruge det på en anden. Litteraturen har vist os, at brugerens mentale model af systemet har stor betydning for brugen af systemet.

Brugen af kontinuitetsbegrebet har været medvirkende til at sikre at Ajour lever op til brugerens mentale model, og har dermed også sikret at der ikke opstod brugbarhedsproblemer i overførslen fra

Ajour Tablet til Ajour Pocket. I de tilfælde hvor vi har afvejet fra brugerens mentale model, har vi til gengæld set, at der er opstået problemer, som især har betydet store nedsættelser af effektiviteten i brugen. Contextual Design har ligeledes haft betydning for, at brugernes mentale model blev tænkt ind i designet af Ajour, i kraft af den grundige undersøgelse af problemområdet.

I udviklingen af Ajour, har vi fremstillet et system, som ikke signifikant adskiller sig fra Judex' amPHI i antallet af brugbarhedsproblemer, arbejdsbelastning, effektivitet eller formålstjenstlighed, og vi har oplevet, at det i høj grad er de samme problemer der går igen på disse multiplatform systemer. Navigationen fungerer bedst på en stor skærm, hvor brugeren har overblik over indholdet, og idet at der skiftes til et mindre display, opstår der problemer med at navigere i systemet.

Vi kan konkludere, at det at skabe et brugbart multiplatform system, i første omgang ikke adskiller sig fra det at skabe et brugbart enkelplatform system. Hvis brugerens mentale model af systemets funktioner følges, og almene "commonsense" retningslinier for brugergrænseflade-design følges, vil systemet som udgangspunkt have en rimelig brugbarhed. Det der er specielt for multiplatform systemer er, at der kan opstå problemer i overgangen mellem systemets platforme. Således kan vi konkludere, at det for at sikre brugbarhed, når et system flyttes fra en platform til en anden, er vigtigt at kontinuiteten i systemets opførsel fastholdes, ligesom vores eksperiment har vist os, at hvis systemets opførsel ændres, fra apparat til apparat, vil bruddet med brugerens mentale model af hvordan systemet opfører sig betyde, at brugbarheden nedsættes.

8.2 Begrænsninger

Dette projekt ligger under for en del begrænsninger. Når vi igennem projektet har talt om multiplatform systemer, har vi reelt kun arbejdet med de to typer platform, som vi har arbejdet med i udviklingen af Ajour, Tablet PC'er og Pocket PC'er, hvor den litteratur, der findes på området, generelt forsøger at favne bredere, og således indfange samtlige mulige platforme, der kan være indeholdt i et multiplatform system. Dette betyder naturligvis, at de resultater vi har opnået igennem vores eksperiment, kun gælder for systemer, som er en kombination af Tablet PC, og Pocket PC. Den case vi har arbejdet med, er desuden forholdsvis speciel. Selve de arbejdsopgaver der løses ved hjælp af de systemer vi har testet, er meget fokuserede på registrere en patients tilstand, hvorfor det er naturligt, at det meste arbejde i at bruge systemerne er, at finde en gruppe knapper, og vælge den i gruppen, som repræsenterer patientens tilstand.

I vores brugbarhedseksperiment valgte vi, at benytte falckreddere, som før havde brugt et system, som var en forgænger til den version af amPHI, som vi brugte. Dette betød, at de havde en del erfaringer med, hvorledes et sådant system fungerede, og dermed også havde en helt klar forventning til, hvorledes de skulle løse de opgaver vi stillede dem. Dette forhold har også betydet, at vi ikke kan vide, hvor vidt testpersonernes tidligere erfaringer med amPHI har haft betydning for resultaterne af vores eksperiment.

Vi har fravalgt at drage konklusioner på baggrund af de data vi har indsamlet omkring tilfredshed. Grunden til dette har været en stærk usikkerhed på, om disse data var valide, når testpersonerne for det første vidste at Ajour var et system, som vi havde udviklet, og desuden havde forudindtagede meninger om amPHI, de benyttede en ældre version af til daglig.

8.3 Videre arbejde

Der er adskillige ting, som vi mener kunne være interessante, i forhold til at arbejde videre med dette projekt. Ajour indeholder brugbarhedsproblemer, som det ville være enkelt at rette en stor mængde af, så systemet ville fremstå klart bedre i en direkte sammenligning med amPHI.

Alternativt har vi set, at nogle af de designvalg, som vi har foretaget os, klart kunne forbedre amPHI, hvis de blev implementeret deri.

Vi har lavet et stykke arbejde omkring alternativer (appendix A), som kunne gøre et system som Ajour bedre. Vi mener, at hvis vi implementerede og aftestede vores begrænsede prototype, imod den fulde Pocket PC, og mod amPHI's Pocket PC, ville vi kunne se, at mange af de effektivitetsproblemer, som skyldes navigationsproblemer, ville forsvinde, fordi brugerne ville få et langt bedre overblik over indholdet. Vi kunne desuden arbejde videre med det koncept for ”mode-skift” i brugergrænsefladen, som vi kiggede på i appendix C. Dette ville være et tiltag, som igennem context-awareness, ville skabe yderligere forbedringer i forhold til effektivitet, idet at brugerne kun ville blive præsenteret for de dele af systemet, som de havde brug for i den givne situation.

Vi har set et problem i vores eksperiment, i form af et validitetsspørgsmål. I denne forbindelse kunne det være interessant at afteste Ajour og amPHI på flere falckreddere, som ikke havde nogen forudsætninger for at have forudindtagede meninger om nogen af systemerne. Således kunne vi be- eller afvise resultaterne af vores eksperimenter, og eventuelt samtidig se om der var tegn på nogen form for indlæringseffekt i form af en bestemt mental model for systemet, som ville have indflydelse på resultaterne af det første eksperiment.

9 Appendix

9.1 Appendix A – Pocket PC Limited

Den tredje og sidste prototype blev udviklet som en del af den almindelige Pocket PC. Denne begrænsede udgave opstod på baggrund af den forståelse af måden hvorpå et mobilt system reelt kan bruges af falckreddere, som vi skabte os igennem CD processen. Kombineret med direkte ønsker fra falckreddere, var dette en klar indikation af, at det var fornuftigt at lave en sådan version.

Designbeslutninger

Prototypen er grundlæggende en undertype af den almindelige Pocket PC. Redderne udtalte til os, at de reelt kun ville have brug for at kunne registrere målinger af puls, blodtryk og deslige, mens de var udenfor ambulancen. Vi valgte ud fra dette, at lave en prototype, hvori der kun var medtaget de dele af registreringen, hvor det er vigtigt, at registreringen sker samtidigt med observationen, og de dele der omhandler opsamling af patienten – ”patient”, ”tilstand ved ankomst” og ”behandling før ankomst”.

Kørsel	An-komst	Skader	Beh. af Patient	Mål-inger	Patient	Tilstand v. ankomst	Beh. før ankomst	Målinger	GCS
Målinger		GCS							
Øjne:		Verbalt:		Motorisk:	Øjne:		Verbalt:		Motorisk:
Spontan		Orienteret		På opfordring	Spontan		Orienteret		På opfordring
På tiltale		Desorienteret		Lokalisere smerte	På tiltale		Desorienteret		Lokalisere smerte
På smerte		Enkelte ord		Normal bøjerefleks	På smerte		Enkelte ord		Normal bøjerefleks
Ingen		Lyde		Abnorm bøjerefleks	Ingen		Lyde		Abnorm bøjerefleks
		Ingen		Strække-bevægelser			Ingen		Strække-bevægelser
Tilføj				Ingen			Tilføj		
Tid:		10:56	10:56		Tid:		11:16	11:16	
Øjne:		3	4		Øjne:		2	3	
Verbalt:		4	5		Verbalt:		2	3	
Motorisk:		5	5		Motorisk:		2	3	
Ialt:		12	14		Ialt:		6	9	

Figur 53 - Pocket PC fuld → limited

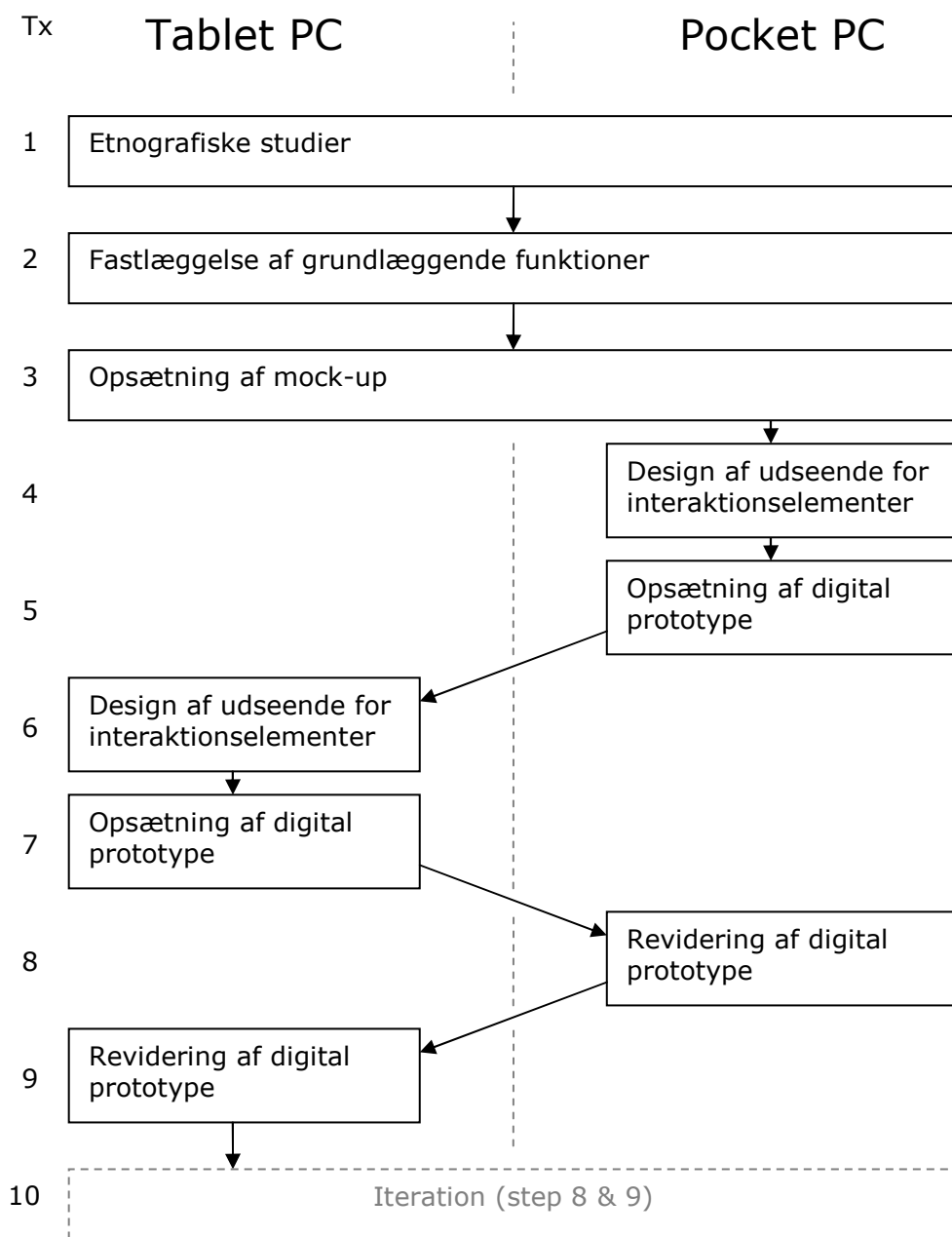
Således kunne halvdelen af menupunkterne fjernes, hvorfor der ikke var brug for den hierarkiske opbygning. Dette betød igen, at der blev en smule mere plads på skærmen, hvorfor nogle af de knapper der før var klemte sammen, kunne få mere plads. Et eksempel på dette kan ses på Figur 53.

Denne "begrænsede" version af programmet adskiller sig ikke i særlig høj grad fra den almindelige Pocket PC-version, men det var vores antagelse, at en begrænset version af programmet ville være målbart lettere at bruge, både i forhold til tidsforbrug, workload og subjektiv tilfredshed, og muligvis endda antallet af brugbarhedsproblemer.

9.2 Appendix B – Judex' udvikling af programmet

I dette afsnit præsenterer vi udviklingen af Judex' system. Systemet består grundlæggende af to separate enheder, dels et program til en Tablet PC, som er tænkt til stationær brug, og dels et program til Pocket PC, som kan benyttes mobilt på en PDA.

Som det kan ses i Figur 54, har Judex' gået igennem en springende proces startende med udvikling til platform, for derefter at koncentrere sig om Pocket PC'en, og for til sidst at fokusere på Tablet PC'en. Vi vil først forklare hvad Judex har gjort i de enkelte trin, for dernæst at diskutere deres begrundelse for at fortage de valg de har gjort. For at illustrere opdelingen i aktiviteterne under udviklingen, har vi valgt at opdele Figur 54 således, at Tablet PC'en er til venstre, og Pocket PC'ens aktiviteter er til højre. Desuden er aktiviteterne delt op efter tidsindex (Tx), således, at aktiviteter med separate kasser, men med samme tidsindex har foregået i to parallelle forløb, mens aktiviteter der deler en kasse på tværs af platformene, har foregået samlet for begge platforme.



Figur 54 - Judex' udviklingsproces

1. Etnografiske studier

Indledningsvis udførte Judex en række etnografiske studier, hvor en person fulgte en række falckreddere på deres kørsler.

2. Fastlæggelse af grundlæggende funktioner

Den grundlæggende funktionalitet blev af Judex til dels fundet ud fra de etnografiske studier, og til dels ud fra et redder kursus som samme person deltog i. Derudover havde Judex adgang til reddernes papirjournal⁷, hvor de enkelte elementer på denne blev studeret i detaljer.

Efterfølgende blev der afholdt en række brainstorming sessions, hvor flere udviklere fra Judex diskuterede de grundlæggende funktioner på plads.

3. Opsætning af mock-up

I dette trin påbegyndte Judex at opbygge selve brugergrænsefladen. Dette blev gjort ved hjælp af Access, hvor udviklerne hurtigt og nemt kunne skabe en brugergrænseflade. Der blev benyttet en tilgang tilsvarende til den som Judex benyttede ved fastlæggelsen af grundlæggende funktioner. Det vil sige en brainstorming, hvor flere udviklere ræsonnerede sig frem til en række mock-up, hvoraf en enkelt blev fortrukket frem for de andre.

4. Design af udseende for interaktionselementer (Pocket PC)

I dette skridt blev det besluttet at udseende for interaktionselementerne skulle designes specifikt mod Pocket PC'en. Denne beslutning blev taget, da Judex på daværende tidspunkt havde problemer med Tablet PC'en, og valgte derfor i stedet at koncentrere sig om Pocket PC'en.

5. Opsætning af digital prototype (Pocket PC)

Opsætningen af den digitale prototype tog udgangspunkt i de mock-up der omhandlede Pocket PC'en, og blev implementeret i eMbedded Visual Basic.

⁷ Se bilag B.

6. Design af udseende for interaktionselementer (Tablet PC)

Da udseende for interaktionselementerne til Tablet PC'en blev designet senere end Pocket PC'en, betød dette at Tablet PC'en blev underlagt de designmæssige valg der allerede var blevet taget i forhold til Pocket PC'en. Overskrifter, knapper og så videre blev derfor designet således at disse ikke kun af udseende var tilsvarende, men også størrelsesmæssigt tilsvarende.

7. Opsætning af digital prototype (Tablet PC)

Judex besluttede i denne fase at benytte sig af de skærbilleder der var på Pocket PC'en på Tablet PC'en. I stedet for to undersider på Pocket PC'en kan begge være på en side på Tablet PC'en. Denne tilgang blev grundlæggende benyttet for alle sider på Tablet PC udgaven.

8. Revidering af digital prototype (Pocket PC)

I denne del af udviklingsfasen, blev Pocket PC'en stemt overens med de rettelser der kom, som et resultat af opsætningen af den digitale prototype på Tablet PC'en. Grundlæggende var udgangspunktet at Pocket PC'en skulle lægge til grund for Tablet PC'en, dog kunne enkelte rettelser på sidstnævnte også medføre rettelser på Pocket PC'en.

9. Revidering af digital prototype (Tablet PC)

Tilsvarende blev Tablet PC'en tilpasset i forhold til de eventuelle rettelser der måtte komme på Pocket PC'en, og som nævnt i skridt 8 havde disse rettelser ofte en større indflydelse på Tablet PC'en ind den anden vej.

10. Design af udseende for interaktionselementer (Tablet PC)

Skridt 8 og 9 er siden blevet gennemgået adskillige gange. Dette inkluderer blandt andet en usability evalueringsrapport som vi udførte for Judex forrige semester af amPHI [And05]. Desuden har der løbende været tæt kontakt mellem udvikler og redderne der benytter systemet i Nibe.

Programmer

Vi vil i dette afsnit beskrive og illustrere resultatet af Judex' løbende design- og udviklingsproces. Processen udmundede i to programmer; et program kørende på en Tablet PC, og et kørende på en Pocket PC, med samme funktionalitet som Tablet PC udgaven.

Hardwaren i de to benyttede apparater bestod af:

PPT 8800 Pocket PC:

- 64 MB RAM
- Intel XScale 300 MHz Processor
- 240x320 pixels TFT skærm
- Microsoft Windows Pocket PC 2003 Professional



Figur 55 – PPT 8800 Pocket PC

MPC-100E Tablet PC:

- 10,4 TFT LCD, Touch Screen og 800x600 (max opløsning)
- Intel XScale PXA255 400 MHz
- Windows CE .NET
- 128 MB SDRAM on board
- 32 MB Flash Rom



Figur 56 – MPC-100E Tablet PC

Resultat af udviklingen

Det samlede resultat af udviklingen blev et komplet sæt af skærm billeder⁸ til henholdsvis Tablet og Pocket PC. Vi vil i dette afsnit illustrere brugergrænsefladens opbygning gennem et par dialog

⁸ Se bilag L

billeder. Dette gøres til dels for at fremvise Judex' valg af farver samt at belyse amPHI' måde at gå i dialog med brugeren.

Behandling

Figur 57 viser "behandling" skærbilledet på Tablet PC'en. I toppen af skærmen er "behandling" markeret med en grøn farve, hvilket indikerer for brugeren for i navigationen denne befinder sig. Judex' har valgt en lyse knapper på en grøn baggrundsfarver. Desuden har de valgt at overskrifter markeret med gul og at bokse, hvori værdier kan angives er hvide. Enkelte overskrifter har dog ikke en gul overskrift men i stedet en lilla, for at symbolisere at informationen under denne er af ren og skær informativ art. Således er det ved disse ikke muligt at indtaste eller markere nogen værdier.

Figur 57 - Behandling skærbilledet på Tablet PC

På Figur 58 har brugeren valgt at trykke på knappen "Iltmaske Venturi", hvorved højre side af skærbilledet erstattes med en dialog boks. Heri skal brugeren angive mængde og godkende. Efter godkendelsen vil ilt type og mængde være synlig i boksen nede i venstre hjørne med overskriften "Værdier givet".

Behandling								Sygehus:	Redder:
Alarndata	Patientdata	Tidligere sygdomme	Førstehjælp	Tilstand v. ankomst	Skader	Behandling			
Respiration:				Iltmaske Venturi:			Oversigt		
Frie luftveje	Iltmaske Resv.	Iltmaske Venturi	Ilt næse	28 %	40 %	80 %	Glasgow Coma Score		
Sugning	Ventilation	Tungeholder	Larynx-maske				Målinger		
Værdier givet:							Medicin		
							Traume-skema		
							Noter		
				Godkend		Tilbage			

Figur 58 - Behandling skærbilledet på Tablet PC, hvor der gives ilt

Figur 59 illustrerer det tilsvarende skærbillede til Figur 57, dog i form af Pocket PC'en. Som det kan ses er skærbilledet delt op i tre undersider, men indeholder ellers de nøjagtige samme data som Tablet PC'en.

Behandling					
Vis	Oversigt	1	2	3	
Respiration:					
Frie luftveje	Iltmaske Resv.	Iltmaske Venturi	Ilt næse		
Sugning	Ventilation	Tungeholder	Larynx-maske		
Værdier givet:					
Venflon Grå					

Behandling					
Vis	Oversigt	1	2	3	
Cirkulation:					
HLR	Defibrilering	Overvågning EKG	EKG-12		
Lejret	Venflon	Drop			

Behandling					
Vis	Oversigt	1	2	3	
Anden behandling:					
Urias	Scoop-bære	Spine-board	Vacuum-splint		
Nødbrus	Halskrave	Vacuum-madras	Repositionering		
Isposer	Bandagering				

Figur 59 - Behandling skærbilledet på Pocket PC

Figur 60 viser hvordan der på Pocket PC'en gives ilt efter at brugeren har trykket på "Iltmaske Venturi". Imodsetning til Tablet PC'en står "Godkend" dog i toppen af billedet, men resultatet ved at trykke på knappen er det samme. Ilt angivelsen samt værdi vises i "Værdier givet" som kan ses på Figur 59.

Behandling		
Godkend	Tilbage	
Iltmaske Venturi:		
28 %	40 %	80 %

Figur 60 - Behandling skærbilledet på Pocket PC, hvor der gives ilt

9.3 Appendix C – Context-awareness

Ud fra den hypotese, at Falckreddere har brug for at registrere forskellige ting, alt efter hvilken type kørsel de er ude på, kom vi frem til, at en funktionalitet, der justerede brugergrænsefladen efter reddernes behov, kunne være ønskelig i vores program, da dette ville lette navigationen. Med afsæt i denne tanke besluttede vi at undersøge nogle af de journaler, som var blevet udfyldt under udrykninger.

Formålet med dette var at se, om der var registreringer der blev brugt så lidt under nogle former for kørsler, at det kunne være gavnligt, ikke at gøre disse tilgængelige i hovedmenuen.

Kørsel	Ydelses- Tekst	Pårørende indberettet	Oplyst af patient	Behandling før ankomst	Tilstand ved ankomst	Skader	Traume- skema	Medicin	GCS	Observa- tioner	Behandling af patient	Andel af kørsler	Kørsler
1	Ambulance t. trafikulykke	73%	36%	73%	64%	82%	82%	18%	45%	73%	82%	2,4%	11
2	Ambulance t. trafikulykke	85%	58%	88%	88%	77%	69%	0%	77%	88%	54%	5,8%	26
1	Ambulance til ulykke	60%	30%	80%	70%	80%	40%	10%	80%	90%	80%	2,2%	10
2	Ambulance til ulykke	90%	58%	90%	90%	88%	9%	6%	53%	69%	66%	17,8%	80
1	Ambulance t. akut sygdom	86%	34%	83%	86%	8%	5%	34%	54%	83%	85%	13,1%	59
2	Ambulance t. akut sygdom	88%	54%	83%	86%	12%	4%	14%	48%	84%	65%	58,6%	263
100,0%													449

Tabel 14 - Procentsatser for registreringstype efter kørselstype

Ud af de 1071 journaler, var 54 testjournaler, som automatisk var blevet sorteret fra, mens en lang række ikke indeholdt nogen registreringsdata, eller var kørselstyper, som var blevet oprettet, men som ikke krævede nogen registrering, udover at kørslen havde fundet sted (eks. ”Ikke i arbejde (u.

beregning))). Det endelige resultat blev, at 449 datasæt indeholdt data som kunne være relevante for vores analyse.

Som det kan se i Tabel 14, var de tre typer kørsel, som vi fandt det relevant at dele op i, "Ambulance til trafikulykke", "Ambulance til ulykke" og "Ambulance til akut sygdom". Desuden blev disse typer delt op i graderne 1 og 2, hvilket er en indikator for hvor akut returkørslen er, hvor 1 er med udrykning, og 2 er uden.

I Tabel 14 kan det ses at de fire første kolonner (Pårørende Underrettet, Oplyst Af Patient/pårørende, Behandling Før Ankomst og Tilstand Ved Ankomst) alle bruges i vid udstrækning under alle kørselstyper, om end "Oplyst Af Patient/pårørende" bruges i ringere grad under mere akutte kørsler, hvilket måske kan forklares med at det i disse tilfælde er vigtigere at behandle akutte skader end at tale med de pårørende. Vi kan se at "Skader" bruges meget lidt under "Ambulance til akut sygdom", og at "Traumeskema" kun sjældent bruges under "Ambulance til ulykke, kørsel 2" og "Ambulance til akut sygdom". Den type registrering der bruges mindst er "Medicin", som kun bruges i mere en 15 % af registreringerne for to ud af de seks kørselstyper. Endelig kan vi se, at "Glasgow Coma Scale", "Observationer" og "Behandling af patient" bruges rimelig jævnt under alle kørselstyper.

Ud fra de observationer vi har gjort os ved hjælp af Tabel 14, har vi besluttet, at funktioner skal gøres tilgængelige som standard, hvis de bruges i mere end 15 % af tilfældene, men ellers kun kunne gøres tilgængelige ved at aktivere et "Full mode". Disse fravalg kan ses i Tabel 15.

Kørsel	Ydelsestekst	Fjernet funktionalitet
1	Ambu. t. trafikulykke	
2	Ambu. t. trafikulykke	Medicin
1	Ambu. til ulykke	Medicin
2	Ambu. til ulykke	Traumeskema, Medicin,
1	Ambu. t. akut sygdom	Skader, Traumeskema
2	Ambu. t. akut sygdom	Skader, Traumeskema, Medicin

Tabel 15 - Fjernet funktionalitet efter kørselstype

10 Summary

In this project, we investigate usability, in the context of multiplatform systems. With the main question “Which problems are associated with creating and retaining usability in multiplatform systems”, the projects initially studies what literature exists on the chosen subject. This study reveals, that the available literature is focused on the transition from “unlimited” to “limited” platforms, e.g. from a standard desktop computer, to a limited device, like a Pocket PC or mobile phone. In this literature, the concept of “Graceful Degradation” is discussed, it being a way to modify elements in a user interface, to fit onto a limited interface. Other topics that have been found of interest in this study are the concepts of “Continuity”, which is used to describe whether or not a given multiplatform device works in the same way on all its platforms, and “Mental Models”, which describe the way in which users understand a computer system, and expect it to function.

After this initial study, the project focuses on the development of a functioning multiplatform system. This system, named “Ajour” is developed using Contextual Design, a method in which the developers of software closely follows the potential users of the system, in what is called “interviews”. During these “interviews”, every aspect of the customers work is discussed, so that both developer and customer get an explicit understanding of the processes involved. A number of these “interviews” are then processed by the developer team in “interpretation sessions”, and the resulting pieces of information are used to form a so-called “affinity-diagram”. From this affinity-diagram, the two developers drew visions (rich pictures) of the proposed system in action, and gained concrete knowledge about which features were wanted in the system, and which questions about the behavior of the customer, had not yet been answered to satisfaction. Finally, this process resulted in mock-ups made with post-it notes, which could be implemented directly in C#.

The final part of the project was a usability-experiment, wherein Ajour and amPHI, a program made by Judex A/S, were tested by Falck rescue-workers, stationed in Nibe. Six workers each tested each of the total of four devices, Ajour Pocket, Ajour Tablet, amPHI Pocket and amPHI Pocket, in order to compare these, and thus gain knowledge about differences and similarities between amPHI and Ajour, and more importantly Tablet PC’s and Pocket PC’s.

The conclusions from this experiment are, at there were no significant differences between Ajour and amPHI, or between the Pocket PC and Tablet PC's. There were minor differences, and a large problem relating to efficiency on Ajour Pocket could be related to navigation issues. It also appeared that these navigation issues occurred as a result of breaking with the mental model, which the users had about the system. This indicated that the step from Tablet to Pocket in Ajour had failed in preserving continuity, and this indicated, that usability problems could indeed arise, due to failures to preserve continuous behavior in the user interfaces on multiplatform systems.

11 Litteraturliste

Bøger

[Hol05]

Holtzblatt K., Wendell J. B., Wood S. (2005) Rapid Contextual Design – Elsevier Inc. ISBN: 0-12-354051-8

[Mol00]

Molich, R. (2000) Brugervenlige edb-systemer.
ISBN 87-571-1647-4

[Nie02]

Nielsen J., Coordinating User Interfaces for Consistency. Neuauflage 2002 Morgan Kaufmann Publishers. ISBN: 1558608214

[Nor90]

The Design of Everyday Things (1990) Basic Books, ISBN: 0465067107

[Rub94] Jeffrey Rubin, 1994: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests: John Wiley & Sons, Inc.: ISBN 0-471-59403-2.

Artikler

[Ble03]

Blechner M, Monaco V, Knox I, Crowley RS. Using Contextual Design to Identify Potential Innovations for Problem Based Learning
Hartford Hospital, CT, USA. PMID: 14728140 [PubMed - indexed for MEDLINE]

[Den03]

Denis C., Karsenty L. Inter-usability of multi-device systems – a conceptual framework. In Seffah A., Javahery H. Multiple User Interfaces: Multi-Devices, Cross-Platform and Context-Awareness. ISBN: 0470854448

[Flo04]

Florins M., Vanderdonckt J. (2004) Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems. IAG – School of Management – Université catholique de Louvain, Place des Doyens, 1

[Kje03]

Kjeldskov J., Skov M. B. and Stage J. (2003) Usability Evaluation of IBM IPJ 2.3 Electronic Patient Record - After one Year of Use (in Danish), October 2003. Aalborg, Aalborg University, Department of Computer Science

[Kje05]

Kjeldskov J., Graham C., Pedell S., Vetere F., Howard S., Balbo S. and Davies J. (2005) Evaluating the Usability of a Mobile Guide: The influence of Location, Participants and Resources. Behaviour and Information Technology 24(1):51-65

[Sef04]

Seffah A., Forbrig P., Javahery H. (2004) Multi-devices "Multiple" user interfaces: development models and research opportunities. The journal of Systems and Software 73 (2004) 287-300

Internet

[Nie97]

<http://www.useit.com/papers/webwriting/>

[msdn]

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnw2kcli/html/W2Kcli_chapter5.asp

Rapporter

[And05]

Andersen, R. F. B., Høgh, L. (2005) Evaluering af amPHI -- kontekstens betydning for usabilityeksperimenter.

<http://www.cs.aau.dk/library/cgi-bin/detail.cgi?id=1105524142>

[Bec02]

Beck E. T., Christiansen M. K., Kolbe N. (2002) Metoder til brugbarhedstest af mobile apparater

<http://www.cs.aau.dk/library/cgi-bin/detail.cgi?id=1023094360>