

dEkspSys F2004

Projektrapport

Gruppe 3:

Jeppe Warberg Larsen - 20012571

Lars Vadstrup Hansen - 19993737

Bent Bisballe Jensen - 20001467

20. maj 2004

Indhold

1	Indledning	3
I	Analyse	4
2	Opgaven	5
2.1	Formål	5
2.2	Systemdefinitionen	5
2.3	Omgivelser	6
2.3.1	Problemområde	6
2.3.2	Anvendelsesområde	7
3	Problemområdet	8
3.1	Klynger	8
3.2	Struktur	9
3.3	Klasser	9
3.4	Hændelser	13
3.5	Anvendelsesområdet	13
3.5.1	Brug	13
3.5.2	Aktører	13
3.5.3	Funktioner	22
3.6	Brugergrænsefladen	23
3.6.1	Dialogform	23
3.6.2	Oversigt	29
3.6.3	Principper	29
4	Anbefalinger	32
4.1	IT-Systemets nytte og realiserbarhed	32
II	Reflektion	33
5	Oversigt	34

6	Brugerinddragelse	36
6.1	Første iteration	36
6.2	Anden iteration	37
6.3	Tredje iteration	40
6.4	Fjerde iteration	41
6.5	Femte iteration	43
7	Tilrettelæggelse af projektforbøb	48
8	Perspektiv	50
9	Erfaringer	52
10	Referencer	54

Kapitel 1

Indledning

Denne rapport er et produkt af vores arbejde i kurset Eksperimentel Systemudvikling foråret 2004, Datalogisk Institut, Århus Universitet. Den første del af rapporten indeholder et analysedokument der beskriver en analyse af et system til en mindre virksomhed som sælger slagtøjsartikler. Virksomheden hedder MusicAid og har 2 ansatte. Systemet skal gøre det muligt at styre lager, sælge varer på internettet og i den fysiske butik samt holde styr på indkøb. Anden del af rapporten er vores refleksioner over det samlede forløb, herunder brugerinddragelse, oversigt over projekt, planlægning af projektforløb, perspektivering der munder ud i nogle konkrete erfaringer og anbefalinger til eventuelle fremtidige projekter.

Del I
Analyse

Kapitel 2

Opgaven

2.1 Formål

Music Aid er en mindre virksomhed med to fuldtidsansatte, der sælger slagtøjsartikler til musikere samt institutioner over hele landet. De har et uoverskueligt lager, hvor det kan tage uforholdsmæssigt lang tid at finde en lille vare. Grunden til at de har alle de små stumper til f.eks. trommesæt fra 1969, er at de er efterspurgt blandt kendere og pga. deres alder svære at finde reservedele til. De har et regnskabs- og lagersystem, MS¹ Navision, som de ikke får brugt i den daglige drift fordi de synes systemet er for komplekst at sætte sig ind i.

På baggrund af ovenstående problemer i driften har de et ønske om at få udviklet et system, som kan understøtte dem i at overvinde disse, samt at effektivisere salgsprocessen og dermed mindske risikoen for fejl.

2.2 Systemdefinitionen

Herunder har vi skrevet BATOFF kriterierne for vores system.

¹Microsoft

Betingelser	Brugerne af systemet skal have almindeligt kendskab til arbejde med computere, dvs. være fortrolige med grundlæggende brug af IT. Derudover skal de have indgående kendskab til termer og praksis omkring salg af slagtøj.
Anvendelsesområde	Butikken har et specialiseret varesortiment. Systemet skal assistere ekspedienterne i deres daglige arbejde ved at gøre det lettere, hurtigere og mere effektivt at arbejde med varelager, fakturering og tilbudsgivning. Der er brug for automatisering i forbindelse med indkøb og salg, således at alle tal let kan samles i regnskabet. Herudover er der en række opgavelister, som ekspedienterne selv vedligeholder, der også kunne gøres tilgængelige i elektronisk form.
Teknologi	Standard PC, mindre server, internt netværk, samt internetforbindelse.
Objekter	Kunde, Vare, Varekatalog, Ordre, Lager, Kreditor, Indkøb, Profil, Opgave, Arbejdsplan, Forretning, Kundekartotek, Ordreliste, Kreditorliste, Indkøbsliste.
Funktionalitet	Støtte til drift af fysisk butik og online butik.
Filosofi	Gennem automatisering og systematisering vil vi minimere de menneskelige fejl og effektivisere de daglige procedurer i butikken.

Ovenstående BATOFF kriterier giver anledning til følgende systemdefinition:

Systemet skal assistere ekspedienterne i en mindre virksomhed, der sælger slagtøjsartikler. Salgsområdet er dels webbutikken og dels den fysiske butik. Der skal være en nem og overskuelig måde at holde styr på det omfangsrige lager (10.000 varenumre). Systemet skal mindske fejl i den daglige drift og minde ekspedienterne om opgaver de ikke har løst. Anvendelse af strekkoder bliver ikke taget med i betragtning. Systemet betragter ikke butikkens øvrige bogholderi, dvs. udgifter og indtægter som ikke har noget med slagtøjsvarer at gøre (eksempelvis udgifter til el).

2.3 Omgivelser

2.3.1 Problemområde

Det primære fænomen i vores problemområde er det omfangsrige lager. Sekundært kommer indkøb og salg af varer og herunder lagring af kreditor- og

kundeoplysninger. Alt dette munder ud i en oplagt mulighed for online salg.

2.3.2 Anvendelsesområde

Der arbejder to ekspedienter i Music Aid, som tager sig af indkøb og salg af varer, reparation af defekte artikler samt regnskab. Overblikket over lageret ligger hos begge ekspedienter, men der findes ikke et formelt system til styring og optælling. Når der skal bestilles varer fra de enkelte producenter og distributører, kan de slå priser og varenumre op i de kataloger de har til rådighed.

En gang imellem får de nye kataloger, som skal erstatte de forrige. Disse kataloger udgør sammen med deres nuværende lagerbeholdning basis for hele deres salgsapparat. Det skal forstås således at de potentielt kan sælge alle de varer som er repræsenteret i kataloget.

Når en kunde henvender sig for at lave et større, samlet køb, laver ekspedienten gerne et tilbud til kunden. Hvis kunden accépterer tilbudet, kan ekspedienten benytte tilbudet til at lave en faktura til kunden.

Reparationer udføres næsten som et almindeligt salg idet timelønnen til en reparation betragtes som en vare. Der laves dog også en udførlig beskrivelse af hvad der er udført under reparationen, for at dokumentere den forbrugte tid overfor kunden.

De gemmer alle fakturaer for indkøb og salg, for at kunne udarbejde moms- og årsregnskab til de relevante myndigheder.

Kapitel 3

Problemområdet

3.1 Klynger

Der er 4 klynger i vores klassediagram: Salg, Køb, Lager og Arbejde. De er markeret ved de grå felter på klassediagrammet.

Salgsklyngen

Salg repræsenterer alle indgående transaktioner i forretningen, dvs. kundekartoteket og ordresystemet. Denne klynge har kendskab til lager klyngen, med henblik på ordreoprettelse.

Købsklyngen

Køb repræsenterer alle udgående transaktioner i forretningen, dvs. kreditor-kartoteket, samt bestillingssystemet, som vi har valgt at kalde *indkøb*. Køb har naturligvis, ligesom salg, kendskab til lager.

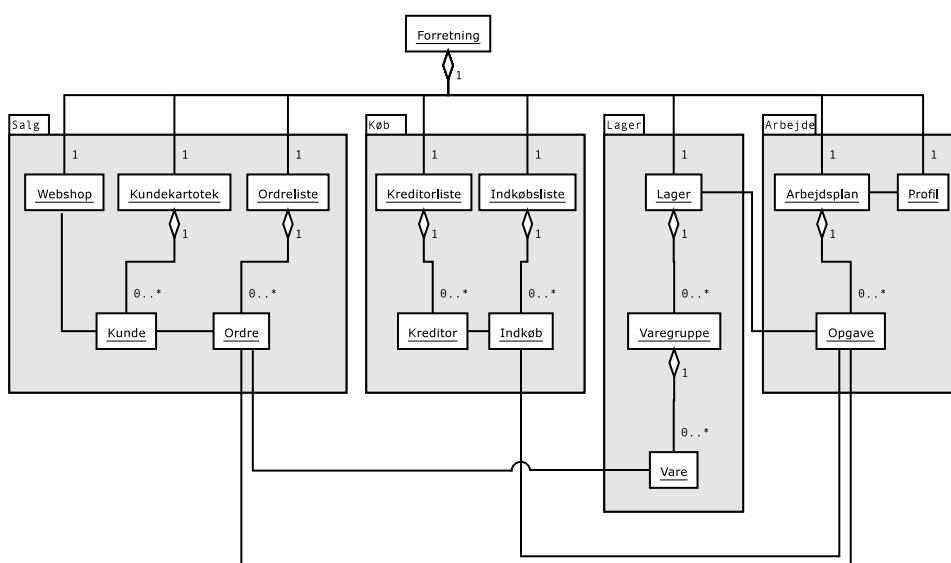
Lagerklyngen

Lager holder styr på eksisterende varer, hvad enten de findes fysisk i butikken (hvis der rent faktisk er indkøbt nogen, som endnu ikke er solgt), eller de blot eksisterer listet i et varekatalog over varer, som forretningen kan bestille hjem, og derfor potentielt har mulighed for at sælge til en kunde. Lageret indeholder derudover en række informationer omkring en given vare, på samme måde som den tykke mappe med varekataloger.

Arbejds-klyngen

Arbejde indeholder informationer om igangværende processer i forretningen, samt kommende opgaver.

Der er her en repræsentation af de ansatte, samt en liste med opgaver. Det



Figur 3.1: Klassediagram

gælder både opgaver som skal udføres og de der er blevet udført. Disse opgaver udgør tilsammen alt hvad der foregår i forretningens daglige arbejde, eksempelvis en bestilling, en betaling, et salg eller lign.

3.2 Struktur

Den overordnede objektstruktur (som ses på figur 3.1) er en træstruktur (forbundet med aggregeringer), idet forretningsklassen agerer rodknude med de seks *samlingsknuder* som undertræer, (kundekartotek, ordreliste, kreditorliste, etc...) som igen har dataknuder/dataklasser under sig.

Kendskabsrelationerne mellem klasserne beskriver hvordan klasserne er koblet sammen på tværs af klyngerne, eksempelvis *Ordre* som knytter en *Kunde* og en *Vare* sammen, og ved oprettelsen nødvendiggør et stykke arbejde som vil blive registreret som en *Opgave*.

3.3 Klasser

Forretning

Forretningsklassen repræsenterer helheden. Den er den fysiske ramme omkring alt hvad der foregår i forbindelse med handel. Den har kendskab til alle transaktioner, på alle niveauer, og kan på den måde skabe et overblik med henblik på statistik eller regnskab.

I den fysiske verden manifesteres forretningsklassen ved chefens skrivebord, med bunker af papirer, indeholdende informationer og rapporter over hvad der *er* foregået, *skal* foregå og *kan* foregå.

Webshop

Webshopklassen repræsenterer den hjemmeside hvorpå online kunder har mulighed for bestilling/køb af varer, som findes på lageret og er valgt til at figurere her (En del varer egner sig ikke til salg over nettet).

Kundekartotek

Kundekartoteksklassen er stedet hvor eksisterende kunder kan findes. Der er en række metoder stillet til rådighed til at søge efter dem, eksempelvis ved gennembladrning eller en alfabetisk sortering, afhængig af repræsentationen. I den eksisterende forretning er kundekartoteket repræsenteret ved hhv. en minimalistisk database (MS Navision), samt et hav af gule post-it notes.

Kunde

Kundeklassen er den faktiske repræsentation af kunden, i form af navn, adresse, telefonnummer, etc.

Nogle kunder handler hyppigt i forretningen og skal derfor være nemme at finde frem til, hvorimod andre kunder kun handler een gang, og ikke udleverer personoplysninger i processen.

Eksempler på de to typer kunder:

Et eksempel på den første type er en skole, som jævnligt køber nye dele til musiklokalets trommesæt. Her er en lang række informationer inkl. en eventuel rabat eller betalingsordning som skal huskes. Denne type kunde findes i den eksisterende forretning i den simple database.

Et eksempel på den anden type kunde er den mere supermarkedsagtige kunde, som kommer ind i forretningen, finder sin(e) vare(r), går hen til kassen, betaler og forlader forretningen igen. Denne type kunde kræver ingen informationer overhovedet og kan systemmæssigt betragtes som en *John Doe*¹ kunde.

Ordreliste

Ordrelisteklassen er en samling ordrer. Man har her mulighed for at danne sig et overblik over hvilke ordrer som er lavet hvornår.

Ordrelisten er i dag repræsenteret ved en bunke post-it notes sat på spyd (igangværende ordrer), samt en mappe med håndskrevne kvitteringer (afsluttede ordrer).

¹En John Doe er en unavngiven person

Ordre

Ordreklassen repræsenterer en igangværende eller afsluttet handel.

Den knytter en kunde sammen med en mængde varer, samt eventuelle rabatter eller betalingsaftaler.

En ordre kan være i een af tre tilstande: *Tilbud*, som blot er en samling varer, med en tilknyttet pris (og eventuel rabat), *Ordre*, som er når kunden har sagt ja tak til et tilbud og sidst *faktura*, når kunden har modtaget varerne i ordren og betalt dem.

Kreditorliste

Kreditorlisteklassen er stedet hvor eksisterende kreditorer kan findes. Der er en række metoder stillet til rådighed til at søge efter dem, eksempelvis ved gennembladrning eller en alfabetisk sortering, afhængig af repræsentationen. I den eksisterende forretning er kreditorkartoteket repræsenteret ved mappen med varekataloger, hvor hvert varekatalog viser hvilke varer der er tilknyttet hvilken distributør.

Kreditor

Kreditorklassen er den faktiske repræsentation af en kreditor. Denne har ligesom kunden en række attributter som oplyser om lokalisering og kontaktmuligheder. Derudover har en kreditor en række varenumre tilknyttet, som er den mængde varer, som den givne kreditor har mulighed for at importere. En kreditor er således en importør, distributør eller producent.

En kreditor udsender med mellemrum et varekatalog med de varer, som vedkommende udbyder, samt et opdateret priskatalog.

Indkøbsliste

Indkøbslisteklassen er en samling af indkøb. Der er her mulighed for at skabe sig et overblik over hvilke indkøb som har været foretaget hvornår, samt den samlede udgift inden for et givent tidsmæssigt interval (med henblik på regnskab og kasseopgørelse).

Indkøb

Indkøbsklassen repræsenterer en igangværende eller afsluttet handel med en kreditor. Den knytter en kreditor sammen med en mængde varer, samt eventuelle rabatter eller betalingsaftaler.

Et indkøb kan ligesom en ordre være i tre tilstande, som dog ikke håndteres af forretningen, førend tilstanden *faktura* er nået (Man må antage at der findes en lignende struktur hos distributøren).

Lager

Lagerklassen repræsenterer den overordnede lagerstruktur. Man kan her overskue hvor mange varer man har hjemme af en given slags, samt søge efter varer, både fysisk (lokaliseringmæssigt) og informationsmæssigt. Lagerklassen udgør i dag to dele, nemlig hylderne og kasserne i forretningen, sammen med varekatalogets informationer.

Varegruppe

Varegruppeklassen er en lagerabstraktion som er nødvendig for nemmere at kunne finde en given vare. Derudover knytter den en række varer sammen, som har mange fællestræk. Et eksempel på en varegruppe kan være *storetrommeskind*. En varegruppe kan være mere eller mindre specifik, der kan således være en varegruppe som hedder *trommeskind*, som så at sige indeholder *storetrommeskind*, en varegruppe kan således specialiseres, indtil man når den faktiske vare.

En varegruppe repræsenteres i dag via varens fysiske placering i butikken (den står et sted hvor der er flere varer af samme slags), samt i varekatalogerne, idet en distributørs katalog og prisliste som regel kun indeholder een type varer.

Vare

Vareklassen indeholder en mængde informationer om en given vare, den kan betragtes som det nederste blad i varegruppetræet. En vare har således en lang række fællesattributter med andre varer, hvilket knytter dem logisk sammen. Udover alle disse fællesattributter har en vare een eller flere attributter, som gør den unik. En vare har desuden en række attributter, som er nødvendige for at kunne handle med den (Et antal, en indkøbspris, en pris o. lign.). En vare repræsenteres i dag i kraft af dens fysiske tilstedeværelse eller figurerende på en linie i et varekatalog (evt. med billede).

Arbejdsplan

Arbejdsplanklassen er en overordnet abstraktion af opgaver. Arbejdsplanen er sorteret efter prioritet og indeholder opgaver som skal løses, eller er blevet løst, af en medarbejder.

Opgave

Opgaveklassen er en repræsentation af en opgave som skal løses. En opgave vil typisk være knyttet til en ansat i forretningen, som i objektmodellen er repræsenteret vha. en profil. Opgaven (ligesom opgavelisten) findes i dag kun

i hovederne på de ansatte, samt i en lille udstrækning i form af post-it notes, med memoer om opgaver som skal udføres.

Profil

Profilklassen er en abstraktion over den ansattes hukommelse og arbejdsområde. En profil indeholder således en række kompetenceområder, som igen knytter profilen sammen med en række opgaver, som kræver disse kompetenceområder for at blive løst. Profilklassen er nødvendig idet, strukturen kræver en repræsentation af (aktøren) den ansatte i form af attributter.

3.4 Hændelser

Den følgende liste viser de overordnede hændelser.

Alle samlingsknuderne er undladt, idet de ikke indgår direkte i hændelserne.

Hændelse	Kunde	Ordre	Kreditor	Indkøb	Lager	Opgave	Profil
Vare salg	✓	✓			✓		✓
Vare forespørgsel	✓				✓		✓
Vareforsendelse	✓	✓			✓	✓	✓
Varebestilling			✓	✓		✓	✓
Varemodtagelse			✓	✓	✓	✓	✓
Tilbudsgivning	✓	✓			✓		✓
Registrer kunde	✓						✓
Registrer kreditor			✓				✓
Ansættelse af medarbejder							✓
Opsigelse af medarbejder							✓

3.5 Anvendelsesområdet

3.5.1 Brug

Vi har i alt tre aktører, nemlig ekspedienter, webkunder og et ikke nærmere bestemt regnskabsprogram f.eks. MS Navision. Vi vil herunder kort gøre rede for de enkelte aktører, men vægten vil ligge på ekspedientens rolle.

Vi har nedenstående brugsmønstre fastlagt udfra de møder, vi har haft med vores kunde.

3.5.2 Aktører

Vores første aktør er ekspedienten, der har den helt centrale rolle i vores system. Ekspedienterne kan, som det beskrives i nedenstående aktørtabel, have forskellige roller og rettigheder. Vi har ikke fået fastlagt disse roller og rettigheder nærmere i vores analyse, men det er klart for os, at der skal være

nogle begrænsninger for enkelte ekspedienter, i modsætning til en ekspedient med administrative rettigheder til alle områder i vores system. Det kunne f.eks. være Lars fra MusicAid, der i en periode skulle bruge en afløser, som han ikke ønskede kunne komme til at kigge i det samlede regnskab eller lignende. Derfor ville Lars give denne afløser de nødvendige rettigheder til at ekspedere og fakturere kunder, men ikke give vedkommende rettigheder til at lave en samlet kasseopgørelse eller eksport til en ekstern fil.

Ekspedient

Formål: En person, der udfører salg, tilbudsgivning, fakturering, statistik samt lageropdatering.

Karakteristik: Der er en varierende grad af erfaring med brug af IT blandt ekspedienterne.

Eksempler:

Lars (ekspedient i Music Aid) der har stor erfaring med brug af IT og stor forståelse for IT koncepter generelt.

Ad hoc arbejdskraft der skal træde til ved sygdom, ferie o.lign., kan ikke forventes at have samme erfaring som de fastansatte.

Vores anden aktør er webkunden, der via selvbetjening kan bestille varer og kontakte butikken vedrørende tidligere eller fremtidige bestillinger. Der stilles store krav til webbutikken, som det fremgår af nedenstående aktørtabel.

Webkunde

Formål: En person, der selv, via webshoppens registrerer sig som kunde og foretager en bestilling samt evt. betaling af varer.

Karakteristik: Webkunderne har også en varierende grad af erfaring med IT, hvorfor det er vigtigt at webshoppens har et interface, hvor brugbarheden er sat i højsædet.

Eksempler:

Webkunde A har aldrig handlet på nettet før, men fundet ud af at det er en nem og smart måde at få fat i specielle reservedele til sit trommesæt fra 1969, der ellers er svære at finde. Webkunde A læser grundigt alle vejledninger der står på de enkelte sider i webbutikken.

Webkunde B er en fast kunde på webbutikken der flere gange om måneden bestiller nye trommestikker og trommeskind, samt kigger efter attraktive tilbud og nye varer. Webkunde B har stor erfaring både med brug af online butikker og surfing på internetsider generelt, og springer derfor hurtigt til de relevante steder på siderne uden minutøst at læse alle instruktioner.

Vores tredje aktør er et regnskabssystem, som skal kunne få data fra vores butikssystem. Vi har ikke fået fastlagt nærmere hvilket regnskabsprogram der konkret kunne være tale om, men indtil videre kunne man forestille sig at butikken ville beholde deres udgave af MS Navision, som deres revisor også benytter sig af.

Regnskabssystem

Formål: En system, der som minimum, skal kunne modtage en ordreliste med beløbene for hver enkelt ordre, en indkøbsliste med beløbene for hver enkelt vareindkøb samt andre indtægter og udgifter der måtte være registreret i vores system. Derpå skal det kunne beregne indtægter samt ind- og udgående moms.

Karakteristik: Det specifikke format et regnskabssystem skal bruge, er ikke kendt på nuværende tidspunkt, men eftersom alle data i vores system ligger i en relationel database, er det ikke uoverkommeligt at producere et passende format til det pågældende regnskabssystem.

Eksempler:

Et eksempel på et regnskabssystem kunne være MS Navision, som butikken allerede har en licens til.

Brugsmønstre

Nedenstående brugsmønstre er lavet i samarbejde med Lars fra MusicAid. Vi anvendte flere forskellige metoder bl.a. interviews, observationer og mockups. Således har vi både benyttet nogle af vejledningerne fra [Mathiasen et al], men også anvendt yderligere teknikker til bestemmelse af brugsmønstre.

Indkøb

Brugsmønster: Indkøb udføres af ekspedienten. Der er flere måder at tilgå et indkøb på. Der er første gang, hvor selve indkøbet oprettes. Her kan ekspedienten tilføje de varer der skal bestilles hjem. Det er ikke sikkert at en pris er tilgængelig på bestillingstidspunktet. Derfor kan ekspedienten vente med at indføre denne oplysning til senere. Varerne der skal bestilles, kan ekspedienten finde via systemet, eller selv oprette en ny vare der skal bestilles hjem. På et senere tidspunkt, når varerne kommer hjem på lager, kan ekspedienten kontrollere, om prisen, antallet og typen af varer passer med det indkøb som blev oprettet i sin tid - hvis ikke, har ekspedienten mulighed for at tilrette indkøbet eller evt. sætte een eller flere varer i restordre. Hvis kreditoren der skal bestilles varer hos ikke er oprettet, kan ekspedienten oprette en kreditor i systemet.

Objekter: Lager, Kreditor, Indkøb

Funktioner: Opret indkøb, rediger indkøb, fjern indkøb, opret kreditor, opret vare

Opret kreditor

Brugsmønster: Her kan ekspedienten oprette kreditorer, der indkøbes varer hos.

Objekter: Kreditor

Funktioner: Opret kreditor

Redigér kreditor

Brugsmønster: Her kan ekspedienten rette kreditorer, som f.eks. har ændret adresse eller andre oplysninger ekspedienten registrerer.

Objekter: Kreditor

Funktioner: Opdater kreditor

Slet kreditor

Brugsmønster: Her kan ekspedienten slette kreditorer, som ved en fejl er blevet oprettet, eller hos hvilken der aldrig er købt varer. Er der allerede handlet hos kreditoren, kan denne ikke slettes af hensyn til regnskab og konsistens i databasen.

Objekter: Kreditor

Funktioner: Slet kreditor

Disse mønstre sikrer, at man har et overblik over sine kreditorer (leverandører) og at man kan differentiere de forskellige indkøb. Desuden er det her man angiver det antal der er bestilt - man kan ikke redigere antal af varer direkte på den enkelte vare. Således skal sikre at der er god dokumentation af dels indkøb og dels krediteringer på defekte og returnerede varer.

Nyt produkt

Brugsmønster: Vores aktør (ekspedienten) tager følgende skridt for at oprette et nyt produkt: Først oprettes eller vælges varekategori samt evt. tilføjes attributter. Nu udfyldes alle de attributter der tilhører varer i den pågældende varekategori.

Objekter: Vare, Varegruppe, Lager

Funktioner: Opret vare, opret varegruppe

Oprettelse af nyt produkt fra andet lignende produkt

Brugsmønster: For at komme lidt hurtigere gennem oprettelsesprocessen kan vores ekspedient vælge at kopiere et eksisterende produkt for derefter at lave passende tilretninger. Først vælges et lignende produkt (evt. i samme kategori - dette er ikke noget krav). Dernæst kopieres produktet og rettes til, så det har de korrekte attributter.

Objekter: Vare, Varegruppe, Lager

Funktioner: Opret vare, opret varegruppe

Oprettelse af ny mappe

Brugsmønster: Her skal ekspedienten først indtaste mappenavn, hvorefter der oprettes relevante attributter der er specielle for varer i denne mappe.

Objekter: Varegruppe

Funktioner: Opret varegruppe

Redigering af mappe

Brugsmønster: Her kan ekspedienten redigere mappenavn samt oprette og slette attributter der er specielle for varer i denne mappe. Man kan ikke fjerne globale attributter i en undermappe - så skal man slå op i mappetræet og finde den mappe der "ejer" attributterne. Der er også nogle helt faste attributter, som pris og produktkode der er systemattributter og derfor ikke kan fjernes af vores ekspedient (hvilket i realiteten nok heller aldrig kunne være ønskeligt).

Objekter: Varegruppe

Funktioner: Redigér varegruppe

Slette mappe

Brugsmønster: Her kan ekspedienten slette en mappe, hvorved alle varer/varegrupper under denne mappe samtidig fjernes.

Objekter: Varegruppe

Funktioner: slet varegruppe

Der er mulighed for at oprette tilbud, som ikke figurerer som salg før de har ændret status til faktura. Fra salgsmodulet skal man ligeledes kunne udskrive en ordrebekræftelse, som ikke er andet end et dokument der ligner en faktura, på nær at overskriften er ordrebekræftelse. Det er ikke altid muligt eller nødvendigt at notere en kundes navn på en faktura, som f.eks. ved kontant salg i butikken. Derfor skal der være mulighed for at udelade kundenavn ved fakturaoprettelse.

Ved fakturaer skelnes mellem kontant faktura og faktura med betalingsfrist.

Herunder præsenteres et par konkrete eksempler på mere detaljerede brugsmønstre.

Oprettelse af faktura / tilbud:

Brugsmønster: Her skal ekspedienten tage een eller flere varer fra kataloget og føje til fakturaen. Dernæst vælger han om det skal være et tilbud eller en faktura. Enten oprettes eller findes kunden blandt eksisterende kunder. Ekspedienten har også mulighed for at give rabatter, enten pr. vare eller på det samlede tilbud. Her skal man enten give et tal for rabatten eller en procentsats. Ligemeget hvad man gør, skal systemet selv regne den anden form ud. Dvs. hvis ekspedienten indtaster en procentsats skal systemet regne rabatten ud i kroner og ører og vice versa. Til sidst kan ekspedienten vælge betalingsform - kontant eller kredit x dage.

Objekter: Ordre, Ordreliste, Kunde, Kundeliste, Lager

Funktioner: Opret ordre (faktura/tilbud), opret kunde

Reparationer

Brugsmønster: Her opretter vores ekspedient en speciel faktura med mulighed for ekstra meget tekst til beskrivelse af hvad der faktisk er foretaget under reparationen. Timelønnen trækker ekspedienten ind på fakturaen - det er oprettet som et varenummer. Hver måned kunne forretningen f.eks. indkøbe 162,5 timer á 150 kr. af hver enkelt medarbejder der arbejder med reparation. Kommentarerne skal løbende kunne skrives til, efterhånden som arbejdet skrider frem.

Objekter: Ordre, Ordreliste, Kunde, Kundeliste, Lager

Funktioner: Opret ordre (faktura/tilbud), opret kunde

Opret kunde

Brugsmønster: Her kan ekspedienten oprette en kunde, med de relevante oplysninger.

Objekter: Kunde, Kundeliste

Funktioner: Opret kunde

Redigér kunde

Brugsmønster: Her kan ekspedienten redigere en kunde, med de relevante oplysninger.

Objekter: Kunde, Kundeliste

Funktioner: Rediger kunde

Slet kunde

Brugsmønster: Her kan ekspedienten slette en kunde, såfremt kunden ikke har handlet hos butikken og dermed har tilknyttede ordrer.

Objekter: Kunde, Kundeliste

Funktioner: Slet kunde

Disse mønstre sikrer, at vi kan opretholde et kunderegister over kunder, der ofte handler i butikken.

Opret opgave

Brugsmønster: For at huske alle de daglige gøremål i butikken, kan det være hensigtsmæssigt for ekspedienten at registrere de opgaver denne ikke umiddelbart kan tage sig af således de ikke forsømmes. Derfor skal ekspedienten kunne oprette opgaver (specificeret i tekst) og sætte en påmindelsesdato på dem.

Objekter: Opgave, Arbejdsplan, Profil

Funktioner: Opret opgave

Disse mønstre skal assistere ekspedienterne i deres daglige arbejde f.eks. med at holde lageret fyldt og bestille varer hjem. De skal også fungere som en slags huskeliste, hvor ekspedienten kan oprette opgaver til sig selv eller en anden ekspedient. Systemet skal så overvåge de enkelte opgavers udløbsfrist og give ekspedienten besked, når tiden nærmer sig. Systemet kan også selv oprette opgaver til ekspedienten, hvis der for eksempel ikke er flere varer på lager af en bestemt type, så vil systemet selv give ekspedienten besked herom og oprette en opgave der skal sikre at ekspedienten får taget stilling til om der skal bestilles hjem af den pågældende vare.

Opret profil

Brugsmønster: Her kan en ekspedient oprette profiler for nye ansatte, således de også kan tilgå systemet. Ekspedienten skal her tage stilling til de rettigheder den pågældende skal have i forhold til de opgaver vedkommende skal løse.

Objekter: Profil

Funktioner: Opret profil

Redigér profil

Brugsmønster: Her kan en ekspedient redigere profiler for ansatte.

Objekter: Profil

Funktioner: Redigér profil

Slet profil

Brugsmønster: Hvis en ansat stopper i virksomheden kan det være hensigtsmæssigt at slette dennes konto, således systemet ikke er åbent for alle tidligere ansatte.

Objekter: Profil

Funktioner: Slet profil

Disse mønstre skal sikre at vi dels kan administrere brugere af systemet og holde styr på hvilken bruger der har adgang til hvad, samt hvad den enkelte bruger af systemet foretager sig i systemet - f.eks. foretager et salg eller indkøber varer og opdaterer lagerstatus.

Følgende brugsmønster er kun et af mange eksempler på statistikker der kan trækkes fra systemet, men det har været et væsentligt punkt for vores kunde, derfor er det taget med.

Kasseopgørelse ved fyraften

Brugsmønster: Her kan man f.eks. se statistikker over dagens indtægter, udgifter samt den nuværende lagerbeholdning.

Objekter: Ordreliste, ordrer, indkøb, lager

Funktioner: Statistik

3.5.3 Funktioner

Nedenstående funktioner er udarbejdet på baggrund af ovenstående liste af brugsmønstre.

Komplet funktionsliste

Beskrivelse	Objekt	Kompleksitet	Type
Opret,	Indkøb	Medium	Opdatering
Rediger	Indkøb	Medium	Opdatering
Fjern bestilling hos leverandør	Indkøb	Medium	Opdatering
Opret	Kreditor	Simpel	Opdatering
Rediger	Kreditor	Simpel	Opdatering
Fjern kreditor	Kreditor	Simpel	Opdatering
Opret,	Salg	Medium	Opdatering
Rediger	Salg	Medium	Opdatering
Fjern tilbud/faktura	Salg	Medium	Opdatering
Opret	Kunde	Simpel	Opdatering
Rediger	Kunde	Simpel	Opdatering
Fjern kunde	Kunde	Simpel	Opdatering
Tilføj	Vare	Medium	Opdatering
Rediger	Vare	Medium	Opdatering
Fjern vare	Vare	Medium	Opdatering
Tilføj	Mappe	Medium	Opdatering
Rediger	Mappe	Medium	Opdatering
Fjern folder	Mappe	Medium	Opdatering
Tilføj	Opgave	Kompleks	Opdatering
Fjern opgave	Opgave	Kompleks	Opdatering
List opgaver evt. filtreret efter forskellige kriterier valgt af ekspedienten	Arbejdsplan	Medium	Aflæsning / Beregning
Opret	Profil	Simpel	Opdatering
Rediger	Profil	Simpel	Opdatering
Fjern	Profil	Simpel	Opdatering
Log på, Log af		Medium	Aflæsning / Beregning
Skriv hændelse for bruger X på objekt Y	Log	Medium	Opdatering
Statistik	Indkøb/salg/Lager	Medium	Aflæsning

Specifikation af funktioner

Den komplekse funktion “Tilføj/fjern opgave”, skal knytte en opgave til et bestemt objekt, og dermed muliggøre et direkte link til den pågældende opgave der skal udføres. Dvs. ekspedienten let skal kunne finde det sted i systemet, hvor den pågældende opgave skal findes. Næste skridt er så, at systemet automatisk vil detektere, hvornår opgaven er udført. Derfor skal de enkelte objekter der bliver tilknyttet en opgave selv sige til opgavemodulet, hvornår opgaven er udført. Herefter skal opgaven automatisk fjernes fra brugerens inbox. Bemærk det er ikke alle opgaver der kan knyttes til et objekt og dermed heller ikke alle opgaver, hvor systemet automatisk kan detektere om den pågældende opgave er udført. Hvis det ikke er muligt, er det op til ekspedienterne at sætte status til “udført” på sådanne opgaver.

3.6 Brugergænsefladen**3.6.1 Dialogform**

Brugergænsefladen i vores system er inspireret programmer til filhåndtering og e-mails. Vi har således valgt dialogmønstre som er velkendte fra moderne

Name	Størrelse	Farve	Producent	Udsalgspris	Indkøbspris	Antal på lager
Pearl 16"-blå	16"	Blå	Pearl	1200,00	1000,00	1
Pearl 16"-hvid	16"	Hvid	Pearl	1200,00	800,00	0
Sonor 14"-stål	14"	Stål	Sonor	2300,00	1500,00	3
Sonor 16"-blå	16"	Blå	Sonor	2400,00	1800,00	1
DW - black pearl	16"	Sort	DW	6000,00	3000,00	1
Ludvig 16"-67 model	16"	Stål	Ludvig	3000,00	2000,00	2

Figur 3.2: Vores mock-up

operativsystemer; menuvalg, skemaudfyldelse og direkte manipulation. Følgende billede er fra vores mockup, skabt vha. billedmanipulation.

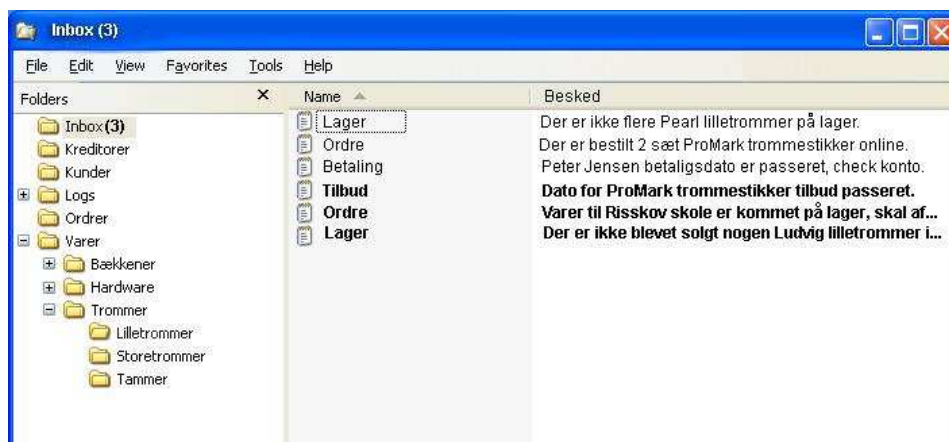
Navigationen i programmet tager udgangspunkt i en mappestruktur, som vist på figur 3.2. Skærbilledet er delt i to, hvor vinduet til venstre viser en række mapper. Ved klik på en mappe udfoldes eventuelle undermapper, og mappens indhold vises i højre vindue. Dette indhold vil for alle mappers vedkommende vises som lister af "filer". Menubaren øverst vil dynamisk ændres til at fremvise de menupunkter der er mulige for den pågældende mappe.

Der er seks mapper:

Inbox

Vi har valgt en email-metaphor for systemets kommunikation til brugeren. Således afspejler den klyngen "Arbejde". I inboxen ligger beskeder om hændelser og tilstande i systemet, som brugeren skal bekendtgøres med og tage stilling til. På figur 3.3 ses et eksemplificeret indhold i vores mock-up. En besked er dels kendetegnet ved en titel som angiver hvilken type information der er tale om, eksempelvis ordre, og dels en kort tekst der fortæller hvilke objekter der er involveret og hvorledes. Ved åbning af beskeden skal der via linkning kunne springes til et relevant sted i systemet, eksempelvis en bestemt varegruppe.

Inbox skal have diverse e-mail funktionalteter som markering af læst/ulæst, arkivér m.m.



Figur 3.3: Forslag til meningen med Inbox.

Kreditorer

Denne mappe svarer til objektet “Kreditorliste”. Her ses en liste over virksomheder som butikken foretager indkøb hos, på samme måde som en adressebog.

Kunder

Listen over registrerede kunder, som objektet “Kundekartotek”. Ligeledes som en adressebog.

Logs

Systemet skal løbende bogføre alle hændelser med konsekvenser. Som det ses i objektdiagrammet, har vi endnu ikke taget stilling til grænsefladedesign af et sådant log-system.

Ordre

Mappen Ordre, som afspejler “Ordreliste”, skal vise en liste over alle aktuelle ordre i systemet. En ordre er kendetegnet ved en række attributter.

Varer

I vores prototype har vi koncentreret os om varelageret, som er essentielt i vores system, jvf. klassediagram og brugsmønstre.

Ved klik på Varer udfoldes mappetræet med varegrupperne. De forskellige varetyper har tilknyttet tekniske data, som er hensigtsmæssige at angive

Navn	Varegruppe	Pris	Indk.Pris	Antal	ProdVare...	Grossist	Producent
Starclassi...	Storetrom...	0	0	0	SCP-BD-...		Tama
Paiste Po...	China	398	169	0	712001p...		Paiste
HHX Evol...	Crash	2100	1250	0	sc367803		Sabian
Rockstar ...	Storetrom...	0	0	0	RSC-BD-...		Tama
Deep Forest	Storetrom...	0	0	0	MPX-BD-...		Mapex
Rockstar ...	Storetrom...	0	0	0	RSCEFX-B...		Tama
Paiste	Effekt	250	104	0	702001p...		Paiste
Zildjian	Crash	1600	750	0	z04cr18		Zildjian
Zildjian A...	Crash	2590	1394	0	1820517		Zildjian
V Series	Storetrom...	0	0	0	MPX-BD-...		Mapex
Swingstar ...	Storetrom...	0	0	0	SSC-BD-BLS		Tama
Saturn Pro	Storetrom...	0	0	0	MPX-BD-...		Mapex
Rockstar	Storetrom...	0	0	0	RS-BD-BRG		Tama
Pro M Seri...	Storetrom...	0	0	0	MPX-BD-...		Mapex
Starclassi...	Storetrom...	0	0	0	SCM-BD-...		Tama
Meinl Clas...	Ride	1795	890	0	me0003ri...		Meinl
Remo	Skind	155	47	0	rem-011...		Slagtojen...
Sabian	Crash	1498	520	0	sc234502		Sabian
M Series	Storetrom	0	0	0	MPX-BD-		Mapex

Figur 3.4: Prototypen - Der skal ikke skelnes mellem mappe/fil-ikoner

som attributter, frem for blot at indgå i den tekstuelle beskrivelse af produktet. Et eksempel er et trommeskind hvis diameter angives i tommer. 5 af attributterne er statiske og globale og findes i alle varegrupper. Det drejer sig om Navn, Varegruppe, Pris, Indkøbspris og Antal.

Figur 3.4 er et screenshot af prototypen. Den markerede mappe er her den generelle Varer, som har fået 3 ekstra attributter. Visningen er rekursiv, dvs. alle varer i undermapperne vises, og det kun med den aktuelle mappes attributter, som har mening for alle underliggende varer. Derfor angives Varegruppe, som henviser til den meste specifikke undermappe. For eksempel er den øverste vare på figuren “Storetrommer”, som, lidt indforstået, hører ind under “Trommer”.

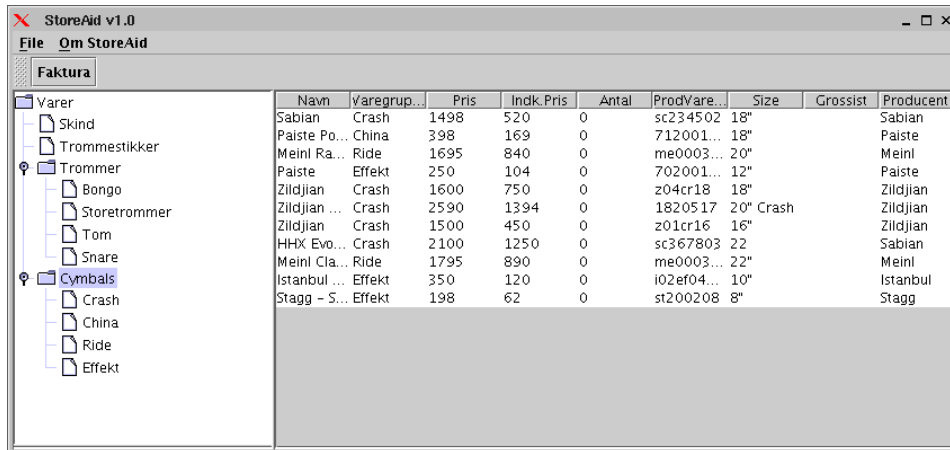
Attributternes navne vises øverst i vinduet, og kan således bruges som en nøgle til hurtig sortering af en varegruppe, på samme måde som man i et filsystem kan vælge at vise mapeindhold sorteret alfabetisk eller efter filstørrelse m.m.

På figur 3.5 er der klikket på Cymbals (bækkener). Denne undergruppe er udvidet med attributen “Size”.

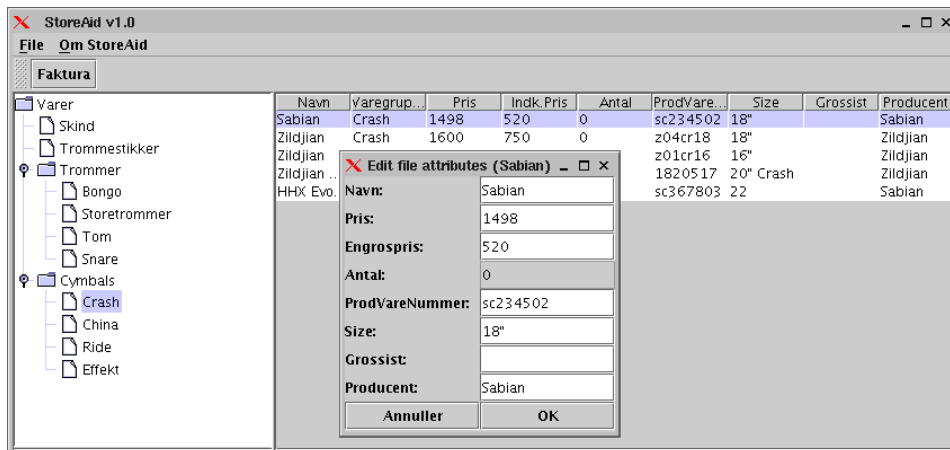
På figur 3.6 er der højreklikket på en bestemt vare og valgt “Redigér vare”. En dialogboks popper op, og de forskellige attributværdier kan ændres ved direkte indtastning. Dog ikke “Antal” som kun kan ændres ved salg/indkøb.

En væsentlig overvejelse omkring brugerfladen er muligheden for manipulation af indholdet i Varer. Det færdige system kan leveres med en varekategorisering der afspejler det nuværende sortiment, men det skal være muligt og nemt at ændre dette. På figur 3.7 er der højreklikket på Trommer og valgt “Redigér mappe”. Under “Alle attributter” vises nedarvede + egne attributter. Ved at indtaste og klikke på “Tilføj” udvider man attributmængden i forhold til den ovenstående mappe. Dette påvirker således også

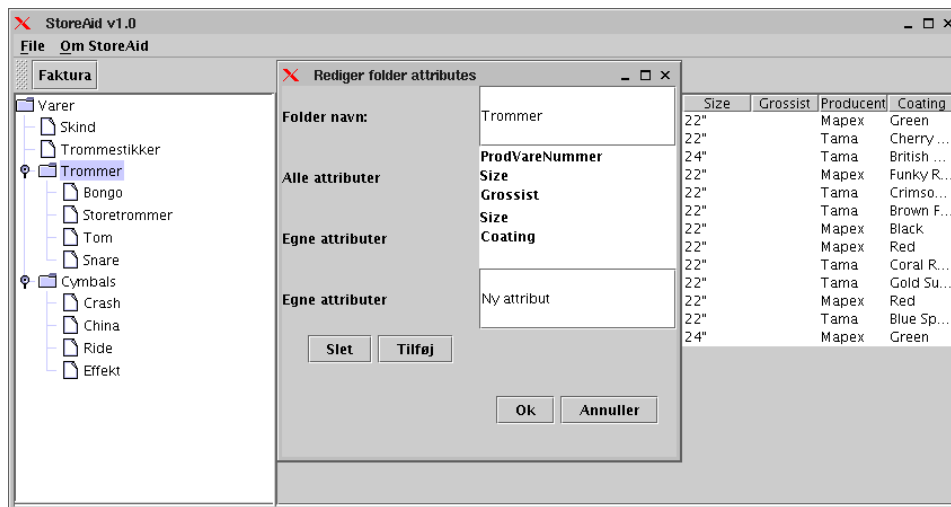
3.6. BRUGERGRÆNSEFLADEN KAPITEL 3. PROBLEMOMRÅDET



Figur 3.5: En undergruppe i Varer er markeret.



Figur 3.6: Redigering af en vare.



Figur 3.7: Redigering af en varemappe.

mappens undermapper. Man kan slette en attribut ved at markere den og klikke "Slet". Dette bør gøres med omtanke. Ændringerne træder i kraft, når der klikkes "Ok".

En ny undermappe skabes ved at højreklikke på en mappe og vælge "Ny mappe". Dialogboksen er den samme som for "Redigér mappe".

Tilføjelser

Brugerfladen skal have en lang række tilføjelser i forhold til det der er beskrevet ovenfor:

- En tilbudsmappe i roden af træet, som man så kunne trække varer over i, for at sætte dem på tilbud.
- Der skal være et søgefelt i toolbaren, som virker på den måde, at man kan indskrænke de viste elementer i filvinduet, udfra en tekstsøgning.
- Der skal overvejes en måde at bestille nye varer hjem på, evt. gennem kreditor, evt. direkte på varen.
- Der skal være bruger-login med password, når klienten opstartes. Når man først er logget på, skal man have mulighed for at skifte bruger, uden nogen som helst form for besvær.
- Der skal være en toolbar, som afspejler de mulige funktioner i den givne kontekst.
- Inbox'en skal angive om den enkelte opgave er udført, og i så fald skal beskeden kunne slettes.

Efter MusicAid blev præsenteret for prototypen, blev følgende bemærket:

- Popup dialogbokse opleves hurtigt som irriterende. Det foreslås at de vises i et fast vindue under fileview'et.
- Der ønskes alternativer til højreklikket. På toolbaren kan der sættes knapper for de hyppigt benyttede handlinger.
- Fleksibel omstrukturering af varer. Opret hurtigt en ny vare, læg den på et generelt niveau, og placér den specifikt på et senere tidspunkt. Markér en eller flere varer og drag-and-drop dem til en anden mappe. Flyt hele mapper.
- Fed skrift i de attributkolonner som kunden finder vigtigst.
- Ændring af vareattributter ved multipel markering. Eksempel: ny leverandør af varemærket Zildjian -> alle Zildjian-produkter markeres og redigeres samtidig.
- Skjulning af data, først og fremmest indkøbspris, ved visning af skærm i en kundesituation.

3.6.2 Oversigt

Navigationen i et filsystem er dynamisk, og derfor er der ikke medtaget et navigationsdiagram med skærm-til-skærm billeder. Når der ikke aktive dialogbokse på skærmen, kan man som udgangspunkt navigere i mapperne vilkårligt.

3.6.3 Principper

[van Welie and Trætteberg, 2000] har beskrevet en lang række interaktionsmønstre som kendetegner gode brugergrænseflader. Den teori har vi brugt som en slags checkliste for at vores GUI overholder normer for hvad der er

“brugbart” “og intuitivt”. Vores system er designet ud fra brugerens ønsker og opfattelse af, hvad en god og brugbar GUI er, men implicit har vi brugt flere mønstre:

Container Navigation: Opdeling af skærmen for overskuelig navigering i grupperede data. Vores system er som sagt inspireret af mail/news-programmer, hvor markering af en mappe medfører visning i en anden splitskærm, hvor markering af en “fil” medfører visning i en tredje.

Grid Layout: På figur 3.6 er dialogboksen arrangeret med minimalt antal kolonner og rækker for større overskuelighed.

Shield: En bruger kan (utilsigtet) udføre handlinger med store konsekvenser. Dertil skal en dialogboks bede brugeren bekræfte at handlingen virkelig ønskes udført. I vores system drejer det sig bl.a. om redigering i varelageret.

Preferences: En bruger skal nemt kunne angive sine præferencer for systemets funktioner og virkemåde. Vores system har en kompleksitet og dynamik, der gør det umuligt at fastlægge alle facetter af GUI'en i udviklingsfasen. Derfor skal flere ting holdes redigerbare i en præference-menu. F.eks. skal man kunne vælge rækkefølge af kolonner, skjul kolonner, kolonner med fed skrift m.m. Inbox-funktionaliteten er ikke beskrevet tilbunds gående, men det er sikkert, at brugeren skal have mange muligheder for opsætning af denne.

Navigating between Spaces: Når pladsen ikke tillader, at alt kan vises på skærmen samtidigt, må objekterne grupperes i logiske enheder som man let kan navigere imellem. Under fileviewet vil vi både have redigering af fildata, salgsfaktura og indkøbsfaktura og vi løser dette med en topstillet, vandret navigation i form af faneblade. Mappetræet er derimod et venstre-stillet, lodret navigationsområde.

Continuous Filter: Hurtigt at finde et element i en sorteret mængde. Brug et filter der kun viser de ønskede elementer. Under udfyldning af faktura skal der hurtigt vælges kunde og varer fra de respektive mapper. Det er en funktion vi ikke har analyseret færdig, men der skal laves en form for indskrænkende søgning. F.eks. at man i kundemappen indtaster de første 2-4 cifre af telefonnumret, og øjeblikkeligt vises de matchende indgange.

[Shneiderman, 1997] opremser en række principper og fordele ved WYSIWYG tilgang, med udgangspunkt i tekstbehandling, og vi har overvejet hvorvidt det lader sig overføre til vores system. Handlinger skal være reversible, dvs. der skal tilbydes en undo-funktionalitet som nemt bringer systemet tilbage til den tilstand den var før den fejlagtige handling blev udført. En ægte og ubegrænset undo-funktion vil være svær at implementere her, og der er mange brugsmønstre som ikke kan annulleres. Til gengæld benytter vi principper om

hurtig respons og intuitiv navigering med cursor og piletaster. Vi har osse en værktøjslinie med ikoner for hyppigt benyttede handlinger, og de kan i et vist omfang vedhæftes grafiske symboler. [van Welie and Trætteberg, 2000] taler også for et sådant Command Area, men påpeger, at der er et trade-off mellem skærmforbrug og udbudet af knapper. Ved nogle knapper giver det bedst mening at påsætte tekst istedet for at opfinde symboler og piktogrammer. Brugeren bliver jo ekspert i systemet og kender dets terminologi.

[Shneiderman, 1997] diskuterer også fordele og ulemper ved direkte manipulation. Vores system har få elementer af dette. Der er ikke mange handlinger som vil give mening at udføre i et visuelt sprog. Blandt de steder det er muligt, er ved omstrukturering af varekatalog og flytning af varer til faktura. Her kan drag-and-drop benyttes.

Kapitel 4

Anbefalinger

4.1 IT-Systemets nytte og realiserbarhed

Systemet her er kun nyttigt for MusicAid såfremt det overgår de eksisterende værktøjer, og er i stand til at integrere de mange funktionaliteter. Hvis systemet ikke indfrier de høje krav til fleksibilitet og hurtighed, vil man i mange arbejdssituationer gøre tingene på “den gamle måde” og derved svigte hele konceptet i elektronisk lagerstyring.

MusicAid har ambitioner om internet-handel, og dertil anser vi elektronisk lagerstyring for en nødvendighed. Ideen er netop, at systemet rummer både butik- og internethandel, så man undgår parallelle systemer.

Et fuldt udviklet system vil desuden kunne lette arbejdet med regnskab og årlig revision.

At systemet er realiserbart, må antages ud fra at mange lignende detailforretninger anvender IT til lagerstyring og e-handel, heriblandt nogle af MusicAid's konkurrenter.

Hardware-aspektet i systemet bør ikke give anledning til hindringer, idet vi antager, at det kan køre på en standard PC.

De økonomiske rammer for en realisering af projektet har vi ikke overvejet. Med systemets mange funktioner og kompleksitet, må man forvente at der ligger mange timers arbejde i programmeringen. Til gengæld kan vi håbe på små udgifter til software, licenser o.l. da vi i høj grad kan benytte ikke-kommercielle værktøjer.

Del II

Reflektion

Kapitel 5

Oversigt

Dato	Type	Beskrivelse
26. Januar	Start	Gruppe 3 dannet.
6. Februar	Forberedelse	Første iteration planlægges (Med DrumCity).
9. Februar	Mail	MusicAid siger officielt ja til at være med i projektet.
11. Februar	Møde	Første møde med MusicAid, hvad ønsker kunden?
17. Februar	Forberedelse	Andet møde med MusicAid planlægges.
18. Februar	Møde	Projektet tager form, det bearbejdede materiale fremlægges.
24. Februar	Forberedelse	Tredie møde med MusicAid planlægges.
27. Februar	Møde	Fremlæggelse af mockup.
2. Marts	Rapport	Første delaflevering afleveret.
8. Marts		Ny objektmodel introduceret.
	Kodning	Prototype udarbejdes
20. April	Forberedelse	Fjerde møde med MusicAid planlægges.
21. April	Møde	Fremlæggelse af prototypen.
27. April	Rapport	Anden delaflevering afleveret.
6. Maj	Forberedelse	Femte møde med MusicAid planlægges.
10. Maj	Møde	Møde ang. faktura system.

Vores projekt har gennemgående været præget af iterationsmønstre. Hver iteration består af 4 faser (jvf. [Bødker et al. 2000]), forberedelse, fokusering, fordybelse og fornyelse. Disse fire faser benytter vi i praksis ved flg.: Til forberedelsen forbereder vi. Til mødet fokuserer vi. Efter mødet (umiddelbart efter, således ikke indsat som event) fordyber vi, samt en smule fornyelse og så igen i forberedelsen til næste møde, samt i den mellemliggende periode sker den egentlige fornyelse.

Alle møderne har på stringent vis været planlagt op ad en specifik interview teknik, på en sådan måde, at vi kunne sikre os at vi ikke påduttede *informanten*¹ viden eller idéer som denne ikke selv ville have fået.

Som dokumentation af møderne har vi benyttet almindelig notering, bortset fra det sidste møde, hvor noteringen var suppleret af de tegninger, som informanten tegnede. Vi har flere gange benyttet to *referenter* med hver sit koncentrationsområde, for på denne måde at få dokumenteret interviewet så bredt som muligt.

Alle møderne er afslutningsvis rundet af med en kort uformel snak med informanten, med fokus på forløbet, udbyttet samt en overordnet resumering af de umiddelbare resultater. Dette blev gjort for at undgå eventuelle oplagte misforståelser i fortolkninger af informantens handlinger, samt for at sikre sig, at informanten følte at interviewet var forløbet på fornuftig vis.

¹Herefter anvender vi kunde, informant og bruger i samme betydning.

Kapitel 6

Brugerinddragelse

6.1 Første iteration

Dagsorden

På dette møde skulle vi først og fremmest begynde at klarlægge kundens behov, samt skabe en fælles referenceramme.

Metode

Vi planlagde vores første møde som et ustruktureret/kvalitativt “in situ” interview [Bødker et al. 2000], hvor formålet var at komme til en større forståelse af kundens ønsker og behov. Det var vigtigt for os at være meget åbne og lytte til kunden, frem for selv at komme med forslag og ideer. Med os til mødet havde vi nogle meget åbne spørgsmål [Online dagbog], for på den måde at få brugeren til at snakke så åbent, som muligt og dermed komme nærmere en forståelse af hvordan han og hans kollega arbejder i butikken i dag.

Resumé

En del af interviewet foregik som en tur rundt i butikken, hvor informanten viste os de forskellige områder og de arbejdsredskaber der blev anvendt de pågældende steder. Han viste os bl.a. deres regnskabssystem MS Navision, MS Outlook, varekataloget (mapper med brochurer og flyers fra leverandører), lageret (til dels ordnede skuffer med de ca. 10.000 forskellige varestumper) og bogholderiet (mapper med fakturaer for indkøb og salg).

Refleksion

Det viste sig at være en god måde at få skabt en fælles referenceramme til det næste møde, idet vi nu havde konkrete udtalelser, som vi kunne bruge til at finde ud af hvordan vi skulle planlægge vores næste møde, samt få spurgt ind

til relevante områder der var blevet bragt op på dette første møde. Dette var også helt i tråd med det, som [Bødker et al. 2000] siger om at det ustrukturerede “in situ” interview ud fra en interviewguide kan lede til en større forståelse af kundens arbejdspraksis. Fordelen ved at lave det “in situ” er:

“..., at interviewet foregår på den pågældendes arbejdssted, hvor informanten derfor kan referere til og vise arbejdsredskaber så som kartoteker, papirer, systemer på computer m.m. under interviewet.” [Bødker et al. 2000]

Hvilket i vores tilfælde var meget nyttigt i det vi hurtigere kunne kommunikere omkring det vi fysisk kunne se i arbejdsminjøet. Det var et godt møde for både os og kunden i det vi fik en masse information, som vi kunne bearbejde inden næste møde og kunden fik et indblik i hvem vi var og hvad projektet i grove træk ville gå ud på. Dog skal det nævnes at vi ikke fik redegjort, som vi ellers havde planlagt, for det samlede projektføreløb i forbindelse med dEkspSys kurset. Vi fik etableret vores første forbillede [Mathiassen et al., 2001], nemlig MS Navision, men typisk blev systemet brugt som et negativt eksempel på hvordan de ikke ønskede systemet skulle fungere, i stedet for at være et direkte forbillede. Der var dog mange elementer, som kunden ønskede i det nye system, men som blot skulle fungere radikalt anderledes. Vi fik også ryddet den første misforståelse af vejen, da vi i forbindelse med de indledende øvelser omkring at få selve mødet i stand havde forstået at MusicAid også beskæftigede sig med udlejning af musikudstyr. Dette var ikke tilfældet. Mødet blev dokumenteret ved vores notater og siden ført ind i dagbogen. Afslutningsvis aftalte vi hvornår næste møde skulle finde sted.

6.2 Anden iteration

Dagsorden

Fra det første møde havde vi fået en mængde information omkring deres arbejde samt de redskaber de benyttede. Vi valgte derfor at tage udgangspunkt i nogle få områder og få klarlagt problemstillingerne omkring dem først.

Metode

For det første var vi interesserede i at kende endnu mere til deres arbejdsgang, hvilket vi ville gøre ved hjælp af dels observation [Bødker et al. 2000] og dels ved endnu et interview denne gang lidt mere struktureret end sidst, dog stadig med meget åbne rammer for svar [Bødker et al. 2000], hvor det var samtalen der var bærende og ikke struktureret omkring spørgsmål og svar. For det andet ville vi gerne have kunden til at sætte flere ord på de ønsker, han måtte have til det nye system. Til mødet havde vi derfor medtaget

tre udskrifter fra et system til lagerstyring, ordrehåndtering og kundekartotek [osCommerce]. Meningen med disse udskrifter var at de skulle agere exploratory prototype [Floyd, 1984] under interviewet, således vi sammen med kunden kunne udforske disse, og inspirere kunden til at kritisere og komme med forbedringer. Derfor var det vigtigt ikke bare at smide disse tre udskrifter på bordet og spørge kunden om han synes om dem, men i stedet grundigt forklare kunden, at dette var ikke noget færdigt udkast, men snarere et oplæg til diskussion og åbent for kritik og forbedringer. Floyd skriver om prototyper:

“..., a practical demonstration of possible system functions serves as a catalyst to elicit good ideas and to promote a creative co-operation between all parties involved.” [Floyd, 1984]

Netop derfor var det vigtigt for os at udskrifterne ikke stod alene, men at der fulgte en grundig og fyldestgørende forklaring med på hvorfor disse udskrifter blev præsenteret. Efter endt gennemgang af ovenstående interview ønskede vi at lave en mere uformelt ustruktureret interview omkring hvordan de i fremtiden ønskede at arbejde. Dette interview skulle også inspireres af prototypen, som vi gerne vil have kunden til evt. at tegne forbedringer direkte på.

Resumé

Til den første del bad vi kunden udføre et par af de arbejdsopgaver, som han omtalte på sidste møde. Først fandt han en vare på lageret, som han så ville lave en faktura på. Den kunne han lave enten i MS Excel eller i en håndskrevet udgave. Han fandt forholdsvis hurtigt varen på lageret til trods for at det ikke formelt er helt systematiseret, men når han dagligt bruger lageret er det ikke nogen større opgave for ham. Oprettelsen af en faktura, som var den anden opgave vi stillede ham, gennemførte han ikke, men sagde blot at han havde de to ovennævnte muligheder. Den tredje opgave vi bad ham udføre var at bestemme en pris på en vare der ikke var mærket, for at se hvordan han arbejdede med det samlede katalog over alle producenter / leverandørernes varer. Her gennemførte han forholdsvis hurtigt opgaven og kunne se prisen på den pågældende vare.

Den anden del af mødet, hvor vi ville have kunden til at tage udgangspunkt i udskriften og fortælle om krav og ønsker til det nye system gik også fint, dog fik vi ham ikke rigtig til at “tegne og fortælle”, han nøjedes med at pege og fortælle, hvilket nok var fordi han ikke helt tog eksempelsystemet til sig. Der kan være flere grunde til dette, men primært tror vi det handler om at det har set for færdig ud, til at han havde lyst til at modificere med egne ideer og kritik. Derfor fik vi ikke skabt en fornemmelse af ejerskab [Mogensen and Trigg, 1992] over eksemplet hos kunden.

I første omgang bad vi kunden sætte nogle ord på overordnede krav til systemet, ud fra det kendskab han havde til deres nuværende system sammen med de fremtidige ønsker til det nye system. Her understregede vi vigtigheden af ikke at lade sig begrænse, men bare komme med alle ideer selvom de ikke synes helt rimelige eller opnåelige på nuværende tidspunkt. Dette er en af faserne¹ i fremtidsværkstedet [Bødker et al. 2000], hvor man beder deltagerne om at komme med alle ideer, selvom de kan synes vanvittige og urealistiske. Meningen med dette var ikke at afholde et fremtidsværksted (som også primært bruges når der er større grupper af brugere der skal komme frem til en fælles konsensus), men snarere at tage et element derfra og bruge til at få så mange ideer på banen, som muligt for så senere at lave en udvælgelsesproces blandt dem. Vi fik vha. prototypen kunden til at sætte ord på de fremtidige scenarier omkring arbejdsopgaver system skulle kunne løse. Disse scenarier mandede ud i en række meget grove skitser for brugsmønstre [Online dagbog]. Desuden fik vi en række ustrukturerede forretningstekniske krav til systemer, herunder bl.a. at kundenummeret ikke skulle være et systemgenereret kundenummer, men snarere kundens telefonnummer, som det er i dag. Dette illustrerer det meget praktiske forhold kunden har til processen.

Refleksion

Omkring vores brug af observation, kan man sige at vi gjorde et par fejl i forhold til retningslinjerne fra [Bødker et al. 2000]. For det første var vi flere der observerede, og derfor kan vi have forstyrret mere end godt var. For det andet, observerede vi ikke konkrete “rigtige” arbejdsopgaver, men snarere fiktive opgaver udformet i fællesskab med kunden. For det tredje så burde vi nok have gennemført observationen som en “flue på væggen” observations-session, snarere end et interview med observation. Så kunne vi nemlig bedre have undersøgt say/do problemet [Bødker et al. 2000] og dermed set om det kunden siger han gør faktisk er det han gør i en arbejdssituation. I forhold til vores brug af exploratory prototype, som en form for inspiration, må vi sige at det er med blandede resultater. Som tidligere nævnt fik vi ikke brugeren til at tage prototypen til sig, men på den anden side fungerede prototypen som et udmærket udgangspunkt for kunden til at sætte ord på en række brugsscenarier.

Det samlede resultat fra dette møde blev et bedre overblik over deres nuværende arbejdspraksis, en række ustrukturerede krav til det nye system samt en række grove skitser til brugsmønstre, der bl.a. indeholdt information om hvordan kunden ønsker at foretage salg, lagerstyring samt diverse statistikker over udgifter og indtægter. Disse resultater er nedfældet i vores

¹Visionsfasen

dagbog.

6.3 Tredje iteration

Dagsorden

Vi holdt os til retningslinjerne angivet i [Mathiassen et al., 2001] og benyttede også teori fra [Bødker et al. 2000] som bruger begrebet “fornyelsesfase” om den situation hvor man har fokus på visionsudvikling. Efter at have klarlagt hovedtrækkene i problem- og anvendelsesområdet samt diskuteret brugerflademæssige tilgange, skulle vi nu præsentere et produkt som kunden kunne opleve og vurdere.

Metode

Til mødet ville vi medbringe en mock-up af en simpel udgave af systemet. Mock-ups er beskrevet af [Ehn and Kyng, 1991]. Ligesom andre tekniker, handler mock-ups om at inddrage brugeren på et tidligt stadie i analyse- og designfasen. Ved at præsentere brugeren for en attrap/papmodel uden nogen funktionalitet implementeret, kan han hurtigt få en hands-on oplevelse af systemet. Brugen af papmodellen bringer hurtigt nye ideer og problemstillinger frem i lyset, så kunde og udviklere får en større fælles forståelse af opgaven. Derved kan der spares tid og penge, da mock-ups er hurtige at lave. [Ehn and Kyng, 1991] fokuserer godt nok en del på mock-ups af større og mere fysiske systemer, men det lader sig fint anvende i vores projekt også. Vores “papmodel” vil være computerbaseret, og er til dels det [Bødker et al. 2000] kalder en horisontal prototype. En sådan model indeholder ingen funktionalitet, men kan bruges til at illustrere designvisioner som kunden på den måde kan vurdere.

Mock-up'en blev fremstillet vha. HTML og fotomanipulerede screenshots af et filsystem i Windows XP. På den måde kunne vi give brugergrænsefladen et velkendt look-and-feel, og med HTML lave en lille smule “funktionalitet” ved at et klik på et udvalgt sted fører til visning af et andet skærbillede. Modellen kan afprøves på projektgruppens hjemmeside².

På forrige møde blev der diskuteret mulige design af systemet og herunder brugergrænsefladen. Dels blev der fremhævet elementer fra e-handel systemet [osCommerce], og dels velkendte mail/news-programmer som Outlook Express. Resultatet blev som tidligere beskrevet i afsnittet om brugergrænsefladen; et system af mapper som afspejler de ringbind og stakke af papirer som forretningen idag benytter. Planen for præsentationen var at lade informanten afprøve modellen mens vi stillede små “opgaver” til brugen af den. Opgaverne var selvsagt ekstremt afgrænsede, men vi vurderede at den til-

²www.daimi.au.dk/deva/dEkspSys/

gang kunne vise, om vores design nogenlunde svarede til brugerens visioner for systemet.

Resumé

Brugeren var generelt meget tilfreds med mock-up'en og den svarede i store træk til de tanker han havde gjort sig. Afprøvningen af systemet gav anledning til nye spørgsmål og diskussioner om de forskellige objekter.

Ordre/faktura: hvad er det, hvordan oprettes ny, hvor på skærmen.

Inbox/besked: hvad skal de fortælle, hvordan kobles de til andre objekter, hvordan håndteres de.

Generelt: hvordan opstarter man / logger på, skal man højre- eller dobbeltklikke på ting, hvilke knapper og menuer er der brug for.

Refleksion

Resultatet af mødet var, at vores indledende design svarede til kundens forventninger, og at det derfor var et vi skulle arbejde videre på.

Angående modelering af mock-up, kunne man spørge, om det var nødvendigt at lave den så forholdsvis grundigt som vi har gjort. Kunne man eventuelt have nøjedes med tegninger på papir? Med brugerens IT-kendskab kunne vi sikkert godt "tale samme sprog" på baggrund af papirskitser. På den anden side gav det en hvis tilfredsstillelse at se en virkelighedstro model, og det gjorde det lettere at forstå systemet.

6.4 Fjerde iteration

Dagsorden

Fremlæggelse af prototypen. Afprøvning af prototypen skulle gerne afdække eventuelle konceptuelle problemer omkring den grafiske repræsentation af lageret.

Metode

Vi valgte at bruge interview metoden *Thinking out loud* [Bødker et al. 2000], men fandt den dog så stringent at vi vurderede at informanten ville blive forstyrret i sin proces ved hele tiden at skulle fortælle om sine handlinger i en høj detaljegrad. Vi modificerede derfor metoden, så brugeren ikke skulle fortælle i stor detaljegrad undervejs i processen, men kun overfladisk forklare hvad der blev foretaget, for først efter hver case at forklare i detaljer hvad der motiverede de udførte handlinger. For at dette kunne virke, var vi nødt til at lave nogle meget små cases, så informanten ikke ville glemme sine forudgående tanker og handlinger, inden de skulle begrundes.

Centralt i eksperimentet havde vi valgt at lave en *Experimental prototype* [Floyd, 1984] som ville være en *Full functional simulation* implementeret i et sprog og framework som gjorde den hurtigt at kode, uden at optimere metoder eller algoritmer. Resultatet blev langsomt, men egnet til at afprøve interfacet omkring lagerstyringen. Evolutionary prototyping var der ikke mulighed for, eftersom vi ikke havde planer om at benytte prototypen i en senere iteration. Prototypen blev lavet som en vertikal prototype [Floyd, 1984] centreret omkring lagerstyringen. Det blev overvejet at lave resten af systemet som en horisontal prototype for at give et indblik i det overordnede layout, men dette blev vi af tidsmæssige årsager nødt til at undlade.

Prototypen skulle testes vha. en række *use cases*, som vi designede til, dels at komme omkring al den implementerede funktionalitet og dels at afspejle de mest almindelige procedurer i forretningen. vi lavede følgende *use cases*:

Case 1

- *Scenarie* En kunde ønsker at købe et 16" Zildjan crashbækken.
- *Opgave* Find Zildjan 16" bækken.

Case 2

- *Scenarie* Du observerer at prisen er forkert.
- *Opgave* Ret prisen.

Case 3

- *Scenarie* Kunden synes det er mange penge for et crashbækken og ønsker et andet, som ikke er oprettet i databasen.
- *Opgave* Opret ny vare.

Case 4

- *Scenarie* En kunde ønsker at købe et trommeskind, som ikke engang er oprettet som varegruppe.
- *Opgave* Opret varegruppe.
- *Opgave* Opret vare.

Case 5

- *Scenarie* En anden ansat har tilføjet en vare uden pris, du kender prisen.
- *Opgave* Find varen uden pris.

Case 6

- *Scenarie* Alle varer mangler en specifik attribut.
- *Opgave* Opret global attribut.

Resumé

Under afprøvningen blev informanten først briefet omkring fremgangsmetoden, hvorefter vi begyndte afprøvningen. De enkelte Use cases blev udført een efter een, mens det blev noteret præcis hvad informanten foretog sig. Han fortalte hvad han tænkte inden han gjorde noget, dog ikke i en (for

ham) forstyrrende detaljegrad. Efter hver use case udspurgte vi nærmere til hvad hans motivation var for de handlinger han havde udført, samt hvilke forventede udfald han havde til hver handling.

Refleksion

Vi oplevede under forsøget at informanten udførte langt de fleste handlinger, som vi havde forventet det. De mest nævneværdige komplikationer opstod i kraft af uhensigtsmæssigheder i prototypens implementation og ikke i selve designet. Der blev ydermere pointeret en række ønsker omkring tilføjelser til brugergrænsefladen i form af shortcuts, toolbar o.lign. Informanten kunne se manglen på disse i kraft af en den en gang imellem besværlige eller uhensigtsmæssige måde at aktivere kommandoer på. En anden ting som informanten opdagede ved brug af prototypen, var at popup dialogerne “Redigér fil” og “Redigér folder” var generende for workflowet. Dette resulterede i, at han ønskede et popup frit arbejdsmiljø, som vi ville tage med i betragtning i den følgende iteration.

Et tilbageblik på vores afprøvningsmetode kunne afsløre at mødet var forløbet som vi havde planlagt det og nye interessante resultater var opnået.

6.5 Femte iteration

Dagsorden

Efter respons på 2. delaflevering var der ikke meget tid tilbage af selve projektet, så vi var i tvivl om hvad det ville give mening at gå i kast med. Vi valgte at holde et sidste møde med MusicAid for at analysere flere løse ender. En af de uklarheder der havde voldt os størst problemer under udarbejdelse af prototype, var fakturaen; både dens funktionalitet og placering på skærmen. Dette vedrører også struktur og søgning i mapperne.

Metode

Der ville ikke blive tid til udarbejdelse af en elektronisk mock-up eller forbedring af prototypen, så vi valgte “rige billeder” jf. [Mathiassen et al., 2001] mao. papir og blyant. Et rigt billede er en uformel tegning som beskriver tegnerens opfattelse af situationen. Målet er således af skabe forståelse og overblik, først og fremmest hos udviklerne. Generelt bør rige billeder ikke give anledning til streng fortolkning, men vi vil her bruge dem til en forholdsvis specifik GUI-beskrivelse.

Vores aktivitet var således at arbejde med eksperimentiel prototype og formålene var analyse og design. Til at udveksle ideer og viden om systemets videre design, ville vi medbringe udskrifter af let modificerede screendumps fra prototypen. På disse var der “skærmpads” til at tegne GUI-elementer og

dynamik i systemet. Udbyttet skulle altså blive en mere konkret beskrivelse af flere centrale elementer.

	Navn	Varegruppe	Pris	Indk.Pris	Antal	ProdVareNummer	Grossist	Producent
Varer	Starclassic Perf...	Storetrommer	0	0	0	SCP-BD-CBK		Tama
Skind	Paiste Power Ch...	China	398	169	0	712001p18		Paiste
Trommestikker	HHX Evolution	Crash	2100	1250	0	sc367803		Sabian
Trommer	Rockstar Custom	Storetrommer	0	0	0	RSC-BD-BRF		Tama
Cymbals	Deep Forest	Storetrommer	0	0	0	MPX-BD-100		Mapex
	Rockstar Custo...	Storetrommer	0	0	0	RSCEFX-BD-CLB		Tama
	Paiste	Effekt	250	104	0	702001p12		Paiste
	Zildjian	Crash	1600	750	0	z04cr18		Zildjian
	Zildjian A Custom	Crash	2590	1394	0	1820517		Zildjian
	V Series	Storetrommer	0	0	0	MPX-BD-105		Mapex
	Swingstar Custom	Storetrommer	0	0	0	SSC-BD-BLS		Tama
	Saturn Pro	Storetrommer	0	0	0	MPX-BD-102		Mapex
	Rockstar	Storetrommer	0	0	0	RS-BD-BRG		Tama
	Pro M Series	Storetrommer	0	0	0	MPX-BD-103		Mapex
	Starclassic Maple	Storetrommer	0	0	0	SCM-BD-CRF		Tama
	Meinl Classic	Ride	1795	890	0	me0003r122		Meinl
	Remo	Skind	155	47	0	rem-0114-ba		SlagteJcentret
	Sabian	Crash	1498	520	0	sc234502		Sabian
	M Series	Storetrommer	0	0	0	MPX-BD-104		Mapex
	Meinl Raker	Ride	1695	840	0	me0003r120		Meinl
	Zildjian	Crash	1500	450	0	z01cr16		Zildjian

Figur 6.1: Projektgruppens forslag til en løsning med faneblade.

Resumé

Gruppen lagde ud med at tegne et forslag til placering af faktura-formularen. Den indsættes som et faneblad i vinduet under fileviewet, se figur 6.1. Et andet faneblad indeholder da visning af en markeret fil, altså et element fra mappen Kunder, Varer, m.m. Kunden, var med på den ide, og vi overdrog pennen til ham for at lade ham beskrive de tanker han havde gjort sig om brug af faktura. En vare skal kunne drag-and-droppes til et felt, se figur 6.2. En alternativ måde at tilføje en vare er at klikke på knappen "Tilføj til faktura", tegnet øverst på skærmen. Herefter blev der diskuteret forskellige tilgange til at søge i de enkelte mapper.

I fakturaen skal der angives et kundenavn, i hvert fald hvis der er tale om en ordre til en kunde i kartoteket. Brugeren ønsker tre væsentlige måder at afslutte faktura-skrivningen, vist ved de tre knapper nederst.

"Tilbud" bruges når man ønsker at give et tilbud på en eller flere varer, som kunden så kan overveje. På samme måde som en ikke-afsendt kladder i et mailprogram. Tilbudet gemmes i en mappe, Tilbud, og kan hentes ind i fakturavinduet igen.

Faktura

Tilføjet til faktura

Varer	Navn	Varegruppe	Pris	Indk.Pris	Antal	ProdYareNummer	Grossist	Producent
<input type="checkbox"/> Skind	Starclassic Perf...	Storetrommer	0	0	0	SCP-BD-CBK		Tama
<input type="checkbox"/> Trommestikker	Paiste Power Ch...	China	398	169	0	712001p18		Paiste
<input type="checkbox"/> Trommer	HHK Evolution	Crash	2100	1250	0	sc367803		Sabian
<input checked="" type="checkbox"/> Cymbais	Rockstar Custom	Storetrommer	0	0	0	RSC-BD-BRF		Tama
	Deep Forest	Storetrommer	0	0	0	MPX-BD-100		Mapex
	Rockstar Custo...	Storetrommer	0	0	0	RSCFX-BD-CLB		Tama
	Paiste Effekt		250	104	0	702001p12		Paiste
	Zildjian Crash		1600	750	0	z04cr18		Zildjian
	Zildjian A Custom	Crash	2590	1394	0	1820517		Zildjian
	V Series	Storetrommer	0	0	0	MPX-BD-105		Mapex
	Swingstar Custom	Storetrommer	0	0	0	SSC-BD-BLS		Tama
	Saturn Pro	Storetrommer	0	0	0	MPX-BD-102		Mapex
	Rockstar	Storetrommer	0	0	0	RS-BD-BRG		Tama
	Pro M Series	Storetrommer	0	0	0	MPX-BD-103		Mapex
	Starclassic Maple	Storetrommer	0	0	0	SCM-BD-CRF		Tama
	Meinl Classic	Ride	1795	890	0	me0003r122		Meinl
	Remo Skind		155	47	0	rem-0114-ba		Slagøjacentret
	Sabian Crash		1498	520	0	sc234502		Sabian
	M Series	Storetrommer	0	0	0	MPX-BD-104		Mapex
	Meinl Raker	Ride	1695	840	0	me0003r120		Meinl
	Zildjian Crash		1500	450	0	z01cr16		Zildjian

dblet til visning
 dblet til visning
 Faktura
 NAVN
 dig-drop
 gem som ordre
 tilbud
 vis faktura
 Confirm

Figur 6.2: Kundens beskrivelse af faktura.

“Gem som ordre” vælges når kunden har indvilliget i at købe og ordren ikke kan effektueres i situationen, enten fordi varerne først skal findes på hylderne eller bestilles hjem. Konsekvensen er, at ordren gemmes i mappen Ordre.

“Faktura” er afslutning på handelen. Når kunden har modtaget varerne og betalt eller fået en regning, kan der udskrives en kvittering. Fakturaen gemmes i systemet.

På et tredje stykke papir, figur 6.3, præciserede brugeren hvordan han ønsker en faktura repræsenteret i systemet. Den skal lægge sig tæt op af den måde som rigtige fakturaer/kvitteringer er udformet (disse kan selvfølgelig variere en del). Varerne vises på en liste, og de kolonner der skal med er vist på figur 6.4:

På de næste kolonner 6.4 Under kolonnen Total summeres en Subtotal og efter en eventuel samlet rabat vises Total, som er den samlede pris, altid med moms. Da diskuterede vi hvordan rabat skal fungere. En rabat skal kunne angives på flere måder. Dels som at beløb i kroner, og dels procentvis. Dette skal tilmed kunne gøres pr. enkelt vare eller for hele handlen. Desuden har MusicAid kunder som får fast rabat. Kombinationer af disse muligheder kan

	Navn	Varegruppe	Pris	Indk. Pris	Antal	ProdVareNummer	Crossist	Producent
<input type="checkbox"/>	Starclassic Perf...	Storetrommer	0	0	0	SCP-BD-CBK		Tama
<input type="checkbox"/>	Paiste Power Ch...	China	398	169	0	712001p18		Paiste
<input type="checkbox"/>	HHK Evolution	Crash	2100	250	0	sc367803		Sabian
<input checked="" type="checkbox"/>	Rockstar Custom	Storetrommer	0	0	0	RSC-BD-BRF		Tama
<input checked="" type="checkbox"/>	Deep Forest	Storetrommer	0	0	0	MPX-BD-100		Mapex
<input checked="" type="checkbox"/>	Rockstar Custo...	Storetrommer	0	0	0	RSCEFX-BD-CLB		Tama
	Paiste	Effekt	250	104	0	702001p12		Paiste
	Zildjian	Crash	1600	750	0	z04cr18		Zildjian
	Zildjian A Custom	Crash	2590	1394	0	1820517		Zildjian
	V Series	Storetrommer	0	0	0	MPX-BD-105		Mapex
	Swingstar Custom	Storetrommer	0	0	0	SSC-BD-BL5		Tama
	Saturn Pro	Storetrommer	0	0	0	MPX-BD-102		Mapex
	Rockstar	Storetrommer	0	0	0	RS-BD-BRG		Tama
	Pro M Series	Storetrommer	0	0	0	MPX-BD-103		Mapex
	Starclassic Maple	Storetrommer	0	0	0	SCM-BD-CRF		Tama
	Meinl Classic	Ride	1795	890	0	me0003r122		Meinl
	Remo	Skind	155	47	0	rem-0114-ba		Slagøjacentret
	Sabian	Crash	1498	520	0	sc234502		Sabian
	M Series	Storetrommer	0	0	0	MPX-BD-104		Mapex
	Meinl Raker	Ride	1695	840	0	me0003r120		Meinl
	Zildjian	Crash	1500	450	0	z01cr16		Zildjian

Figur 6.3: Kunden præciserer beskrivelse af faktura via tegningen.

Antal	Varenummer	Navn	Pris	Rabat	Total

Figur 6.4: En fakturas udformning

blive udviklede, så rabat-funktionaliteten blev ikke slået fast i denne omgang. Dernæst talte vi løst om hvad et ordreobjekt er for en størrelse og hvordan det skal modelleres i systemet. Bl.a. skal det kunne angive en status på handlen; om den er afsendt, om der ventes på indkøb af varer, om der er betalt m.v. Det ledte videre til overvejelser om hvor meget systemet overhovedet skal kende til betaling/betalingsform, da det jo trods alt ikke er et regnskabssystem. En sidste detalje der blev snakket om, er hvad Indkøbspris refererer til og hvad nytte den har i systemet, da den jo kan ændre sig hele tiden. Det synes at give mest mening, hvis prisen henviser til seneste indkøb af varen.

Refleksion

Udbyttet af mødet var en væsentligt større klarhed over designet af faktura og de brugsmønstre der knytter sig til den. Vi fik løst nogle af de brugerflademæssige problemer vi havde med objekternes placering og størrelse på

skærmen. Set i bakspejlet var det noget som burde have været klarlagt på et tidligere tidspunkt. Igen var mødet præget af brugerens gode IT-kendskab og at han havde gjort sig tanker om systemet siden sidste møde. Ved at tegne tingene nåede vi hurtigt til en fælles forståelse af de begreber som kan være svære at udtrykke i alene ved ord.

Kapitel 7

Tilrettelæggelse af projektforløb

I vores projekt har vi taget udgangspunkt i [Mathiassen et al., 2001] og [Bødker et al. 2000]. Dette var i første omgang fordi det var de tekster vi først havde til rådighed og havde et overblik over. Ud fra disse tekster tilrettelagde vi de første møder for at få etableret et fornuftigt samarbejdsgrundlag, samt nogle fælles referencerammer. Vi havde fra starten ikke et klart overblik over alle de iterationer vi skulle igennem, men det blev hurtigt klart for os, at det heller ikke var noget problem, idet vi netop arbejdede i iterationer, hvor man efter hver endt iteration kan udpege nye indsatsområder i samråd med brugerne.

[Christensen et al., 1998] beskriver et projekt omkring et stort internationalt shippingfirma, der er meget større og på mange områder meget anderledes end vores projekt. På trods af dette er der stadig en del paralleller mellem de to udviklingsforløb.

De helt trivielle ligheder er, at vi arbejder i iterative forløb og alle aktiviteter er centreret omkring brugerne.

Et eksempel på en forskel mellem vores projekter er at vi havde valgt ikke at inddrage en undersøgelse af kommunikationen mellem de ansatte på arbejdspladsen, da de kun er to ansatte. Derfor mente vi ikke at dette var et område vi behøvede at tage med i vores betragtning af deres arbejdspraksis. I [Christensen et al., 1998] var der stor fokus på netop dette område, fordi der var så meget kommunikation mellem de mange forskellige afdelinger i det globale netværk af afdelinger.

For at få brugerne til at tænke i konkrete baner introducerede vi tidligt, både exploratory prototypes men også mockups. I modsætning til [Christensen et al., 1998] havde vi ikke en evolutionary prototype, som vi hele tiden udviklede videre på. Dette ville vi også gerne have gjort, men grundet mangel på resourcer var det ikke muligt. I [Christensen et al., 1998] arbejdede OO-udviklerne parallelt med designerne og etnograferne. Det ville have været en stor fordel for

os at benytte den samme metode idet vi, som tidligere nævnt, gjorde vigtige og store fremskridt ved introduktionen af den funktionelle prototype. Her kunne vores bruger bedre forstå de implikationer der lå i de designforslag han blev præsenteret for.

Da vi skulle udvikle prototypen undersøgte vi en række forskellige udviklingsmiljøer, for at skabe os et overblik over hvilke muligheder vi havde for valg af udviklingsværktøjer, deriblandt sprog. Det var fra begyndelsen klart, at sproget skulle være objektorienteret og at vi gerne ville have et IDE¹ der tilbød hurtig udvikling af GUIen. Valget blev dog afgrænset af at det eneste sprog alle gruppens medlemmer havde kendskab til, var Java. Det lykkedes os aldrig at finde et IDE² der på fornuftig vis kunne skifte mellem tekstuel kode og GUI-editor, så vi endte med at lave det meste i Emacs. Det betød en noget længere udviklingstid end beregnet og ændringer blev besværliggjort. [Grønbæk et al., 1997] beskriver en metode kaldet *Cooperative Experimental System Development* (CESD), som i lighed med [Christensen et al., 1998] fokuserer på brugerinddragelse og iterationer, men også understreger vigtigheden af at brugerne inddrages i alle aktiviteter under hele projektforsløbet. Her er de *concerns* der er i et sådant forløb: Planlægning af forløbet (*Management*), analyse (*Analysis*), design (*Design*), realisering - både i teknisk implementation og organisationen (*Realisation*) samt fremtidige tilpasninger af systemet (*Computer Supported Work*). CESD er bygget op omkring activities og concerns.

Forskellen mellem vores projekt og begge de ovennævnte projekter er, at de kigger meget længere frem i processen end vi gør, hvorfor de også skal tage stilling til nogle ting, som vi f.eks. ikke har fået inddraget i vores projekt. Det er bl.a. den videre udvikling efter ibrugtagning i [Grønbæk et al., 1997] samt den føromtalt meget gennemarbejdede evolutionary prototype i [Christensen et al., 1998]. En anden metode vi kunne have anvendt, men valgte fra, var Extreme Programming [Rittenbruch et al. 2002], som ligefrem fremhæver kundens rolle i kooperativ udvikling mellem brugere og udviklere. Således er det kunden der primært styrer processen. Metoden fokuserer på, at man på et hvert givent tidspunkt i processen har et fungerende system, som forbedres iterativt mens *user stories* opdateres under *the planning game*. Den tilgang blev introduceret forholdsvis sent i kurset, og den syntes ikke at indeholde elementer der gav mening at bruge i vores projekt, specielt fordi der i XP skal bruges meget mere tid udvikler og bruger imellem og den havde vi simpelthen ikke.

¹Integrated Development Environment

²Vi benyttede Netbeans IDE til at konstruere hovedvinduet, og så gik vi over til Emacs.

Kapitel 8

Perspektiv

Ud fra dels [Floyd, 1987] og dels [Hirschheim and Klein, 1989] vil vi perspektivere vores projektforsøg.

I [Floyd, 1987] taler de om to forskellige perspektiver i forhold til udviklingsprocessen - nemlig de produkt- og procesorienterede perspektiver, hvor det produktorienterede perspektiv primært er rettet mod formelle, matematisk korrekte programmer i modsætning til det procesorienterede perspektiv hvor fokus primært er rettet mod programmet som et element¹ der indgår som en del i en større helhed [Floyd, 1987]. Floyd skriver også at der er sket et skifte af fokus, fra det produktorienterede perspektiv, til det mere procesorienterede perspektiv. Samtidig understreger hun at begge perspektiver er en ganske naturlig del af alle software udviklingsprojekter.

I [Hirschheim and Klein, 1989] introduceres to akser til at kategorisere fire forskellige tilgange til systemudvikling. De to akser er henholdsvis den ontologiske (konflikt / orden i yderpunkterne) og den epistemologiske (subjektivism / objektivisme i yderpunkterne). De fire kvadranter inddeler således de fire forskellige paradigmer, som er funktionalisme (objektivisme og orden), radikal strukturalisme (objektivisme og konflikt), social relativisme (subjektivism og orden), neohumanisme (subjektivism og konflikt). De fleste projekter udvikles under en kombination af de fire forskellige paradigmer, dog ofte med en af dem, som den dominerende.

I vores projekt har vi været meget omhyggelige med at lægge fokus på brugeren og dennes behov. Dette har vi gjort gennem antagelsen om at det er brugeren der sidder inde med den viden, der i sidste ende bedst lader sig realisere i et færdigt og brugbart system. Denne viden består både af deres ord og handlinger, eller rettere observation af samme. Derfor må vores tilgang i følge [Hirschheim and Klein, 1989] overvejende være subjektiv og vores sociale omgivelser er ordnede. Således er vores udviklingsperspektiv primært præget af et social relativistisk paradigme.

¹Floyd [Floyd, 1987] omtaler programmet, som et værktøj eller et arbejdsmiljø for brugerne

I forhold til antallet af ansatte er vores virksomhed så forholdsmeæssig lille at der ikke kan opstå den slags konflikter mellem ledelse og de ansatte, som [Hirschheim and Klein, 1989] beskriver. Det kan i den forbindelse bemærkes at vi ikke kunne rubricere vores informant som hverken leder eller ansat, men derimod snarere som en helt tredje type - selvstændig forretningsdrivende (både ansat *og* leder). I den terminologi som [Hirschheim and Klein, 1989] benytter har vi valgt at betragte informanten som ansat.

Under størsteparten af vores udviklingsforløb har vi haft et procesorienteret perspektiv [Floyd, 1987], hvilket underbygges dels af vores gennemgående brugerinddragelse, men også vores iterative arbejdsproces, som beskrevet tidligere. Der skete dog et skift til et produktorienteret perspektiv under udviklingen af vores prototype²; produktorienteret i den forstand, at vi havde fokus på et system der formaliserede procedurer ud fra en specifikation [Floyd, 1987]. Efter udviklingen af prototypen var færdig og vi påbegyndte den videre planlægning af projektet, skiftede vi tilbage til det procesorienterede perspektiv.

²I fjerde iteration

Kapitel 9

Erfaringer

På baggrund af vores forløb er der en række erfaringer vi har gjort os, som ville gøre at vi ville være meget bedre rustet til at lave et bedre og mere givtigt forløb, hvis vi nu stod og skulle igang med et nyt projekt.

For det første er der en række uhensigtsmæssigheder i valget af projekt, idet der er flere teknikker, som ikke rigtig passer til et projekt i vores format. Her tænker vi specielt på størrelsen af den organisation vi udvikler systemet til. Alene denne faktor gør at vi drastisk minimerer muligheden for at inddrage flere interessenter og f.eks. se på konfliktløsning mellem de forskellige grupper. Det er en erfaring vi ikke har haft mulighed for at gøre os i forbindelse med dette projekt.

En anden følge af vores projektvalg er at der heller ikke er så meget kommunikation mellem parterne, som kunne være interessant at analysere, som der tydeligvis ville være i et projekt der foregik i en bare lidt større organisation. I begyndelsen af projektet havde vi stor glæde af [Mathiassen et al., 2001], da vi netop havde brug for en meget konkret opskrift på hvordan vi skulle gribe hele situationen an. Denne del ville vi nok holde fast i, da den er en rigtig god måde at få hul på kommunikationen med kunden. I det videre forløb ville vi dog nok have brugt flere kræfter på at udvikle en evolutionary prototype, som et supplement til analysedokumentet. På den måde ville vi hurtigere kunne få alle vigtige aspekter til evaluering. Dette ville kræve et meget bedre udviklingsværktøj, noget i stil med det der blev anvendt i BETA¹, hvor modellen hele tiden afspejledes i prototypen. Således mener vi at vi kunne have opnået et resultat der i endnu højere grad afspejlede kundens behov og ønsker. Prototypen ville også udgøre et godt grundlag for at lave det færdige produkt i det der ville være meget af den relevante struktur og information i prototypen der ville afspejles i det endelige produkt.

Dog skal det nævnes, at vi trods alt har opnået gode resultater med de metoder vi har anvendt i dette projekt og at de bestemt kunne benyttes i et andet projekt i samme størrelse eller større. En af fordelene ved at vælge at arbejde

¹Hermed mener vi det udviklingsmiljø der er beskrevet i [Christensen et al., 1998]

sammen med en mindre organisation har været, at vi let kunne kommunikere med vores bruger og aftale møder. Vores bruger har haft meget let ved at gennemskue IT mæssige problemstillinger, så vi har ikke skulle “oversætte” begreber og vi har ikke skulle bruge lang tid på at sætte ham ind i konceptet ved de forskellige metoder vi har anvendt.

Fordelen er at møderne aldrig bliver træge af forståelsesmæssige komplikationer. På den anden side kan der være detaljer vi overser fordi vi regner med at vi forstår hinanden, men faktisk ikke gør det. Derfor ville det være en fordel med en lidt mere blandet brugergruppe, der havde forskellige forudsætninger og kvalifikationer.

Kapitel 10

Referencer

Vores mockup kan ses på adressen:

<http://www.aasimon.org/dEkspSys/mockup1>

mirror: <http://www.daimi.au.dk/~deva/dEkspSys/mockup1>

Yderligere information om projektet og gruppen, inkl. dagbog kan læses på:

<http://www.aasimon.org/dEkspSys>

mirror: <http://www.daimi.au.dk/~deva/dEkspSys>

Koden til prototypen, samt en kørselsvejledning kan ligeledes findes på gruppens hjemmeside. Eftersom prototypen benytter sig af en postgresql server, vil det være nødvendigt at indsætte et loginnavn og password et sted i koden. For at modtage disse, bedes en mail sendes til deva@daimi.au.dk, hvorefter disse vil blive tilsendt (vi er ikke interresserede i at offentliggøre dem i rapporten eller på hjemmesiden)

Litteratur

- [Bødker et al. 2000] Bødker, K., Kensing, F., and Simonsen, J. (2000). Professionel IT-Forundersøgelse - Grundlaget for Bæredygtige IT-Anvendelser. Samfundslitteratur, Frederiksberg, Denmark. Kapitel 8 og Afsnit 9.2, 9.4, 9.8.
- [Christensen et al., 1998] Christensen, M., Crabtree, A., Damm, C., Hansen, K., Madsen, O., Marquardsen, P., Mogensen, P., Sandvad, E., Sloth, L., and Thomsen, M. (1998). The M.A.D. experience: Multiperspective Application Development in evolutionary prototyping. In Jul, E., editor, ECOOP-98 - Object-Oriented Programming. Proceedings of the 12th European Conference, pages 13-40. Springer Verlag.
- [Ehn and Kyng, 1991] Ehn, P. and Kyng, M. (1991). Cardboard computers: Mockingit- up or hands-on the future. In Greenbaum, J. and Kyng, M., editors, Design at Work: Cooperative Design of Computer Systems, pages 169-195. Lawrence Erlbaum Associates.
- [Floyd, 1984] Floyd, C. (1984). A systematic look at prototyping. In Budde, R., Kuhlenkamp, K., Mathiassen, L., and Züllighoven, H., editors, Approaches to Prototyping, pages 1-18. Springer Verlag.
- [Floyd, 1987] Floyd, C. (1987). Outline of a paradigm change in software engineering. In Bjerknes, G., Ehn, P., and Kyng, M., editors, Computers and Democracy - a Scandinavian Challenge, pages 191-210. Aldershot, Avebury, England.
- [Grønbæk et al., 1997] Grønbæk, K., Kyng, M., and Mogensen, P. (1997). Toward a cooperative experimental system development approach. In Kyng, M. and Mathiassen, L., editors, Computers and Design in Context, pages 201-238. MIT Press.
- [Hirschheim and Klein, 1989] Hirschheim, R. and Klein, H. (1989). Four paradigms of information systems development. Communications of the ACM, 32(10):1199-1216.

- [Mathiassen et al., 2001] Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., and Stage, J. (2001). *Objektorienteret Analyse og Design*, Forlaget Marko.
- [Mogensen and Trigg, 1992] Mogensen, P. and Trigg, R. (1992). Using artefacts as triggers for participatory analysis. In Muller, M., Kuhn, S., and Meskill, J., editors, *Proceedings of the Participatory Design Conference (PDC) 1992*, pages 55-62. CPSR.
- [Online dagbog] <http://www.aasimon.org/dEkspSys>
mirror: <http://www.daimi.au.dk/~deva/dEkspSys>
- [Rittenbruch et al. 2002] Rittenbruch, M., McEwan, G., Ward, N., Mansfield, T., Bartenstein, D. (2002). Extreme Participation - Moving Extreme Programming Towards Participatory Design. In Binder, T., Gregory, J. and Wagner, I. (eds.), *Proceedings of the Participatory Design Conference (PDC'02)*, pp 29 - 41.
- [Shneiderman, 1997] Shneiderman, B. (1997). *Designing the User Interface*. Addison-Wesley, 3rd edition. Kapitel 6.
- [osCommerce] <http://www.oscommerce.com>
- [van Welie and Trætteberg, 2000] van Welie, M. and Trætteberg, H. (2000). Interaction patterns in user interfaces. In Wallingford, E., editor, *Proceedings of the 7th Pattern Languages of Programs Conference (PLoP-2000)*. Washington University Technical Report wucs-00-29.