

Indholdsfortegnelse

1	INDLEDNING	4
1.1	Styresystemer	4
1.2	Arbejdsrutiner	5
1.3	Problem.....	6
1.4	Behandling af problem	7
1.5	Afgrænsning	7
1.6	Rapportens bagmænd	7
1.7	Kapitelloversigt	7
2	OPGAVEN	9
2.1	Formål med systemudviklingsprojektet.....	9
2.2	Systemdefinition.....	9
2.3	BATOFF-kriterier	9
2.4	Omgivelser – rige billeder	10
2.5	Klasser.....	11
2.6	Adfærdsmønstre	12
2.7	Struktur klassediagram	14
2.8	Hændelser	15
3	ANVENDELSESOMRÅDE	17
3.1	Aktørtabel.....	17
3.2	Brugsmønsterdiagram.....	17
3.3	Brug.....	18
3.3.1	Brugsmønstre	18

3.4	Funktioner	25
3.5	Dialogform	26
3.6	Navigeringsdiagram	26
4	DESIGN	29
4.1	Formål	29
4.2	Kvalitetsmål	29
4.3	Teknisk platform	31
4.3.1	Udstyr.....	31
4.3.2	Basisprogrammel	31
4.3.3	Systemgrænseflade	31
4.3.4	Designsprog	31
4.4	Arkitektur	31
4.4.1	Komponentarkitektur	31
4.4.2	Standarder	32
4.5	Komponenter	33
4.5.1	Struktur.....	33
4.5.2	Sekvensdiagrammer.....	34
4.5.3	Klasser.....	35
4.5.4	Nyt brugsmønster identificeret: Slet Bruger	37
4.5.5	Udvidet klassesdiagram.....	38
5	INDFØRING AF METODEELEMENTER I NYKREDIT HHV. INROPA A/S	39
5.1	Virksomhedsbeskrivelse	40
5.2	BATOFF-kriterier	41
5.2.1	Nykredit.....	41
5.2.2	Inropa	41
5.3	Hændelsestabel	41
5.3.1	Nykredit.....	42
5.3.2	Inropa	42
5.4	Klassesdiagrammer	42
5.4.1	Nykredit.....	42
5.4.2	Inropa	43
5.5	Brug af arkitekturmønstre	43
5.5.1	Nykredit.....	43
5.5.2	Inropa	43
5.6	Opsummering af fordele/ulemper	44
6	GENERELLE KVALITATIVE OPLEVELSER	45
7	KONKLUSION OG PERSPEKTIVERING	47

7.1	Konklusion.....	47
7.2	Perspektivering	47
	LITTERATURLISTE	48
	APPENDIX.....	49

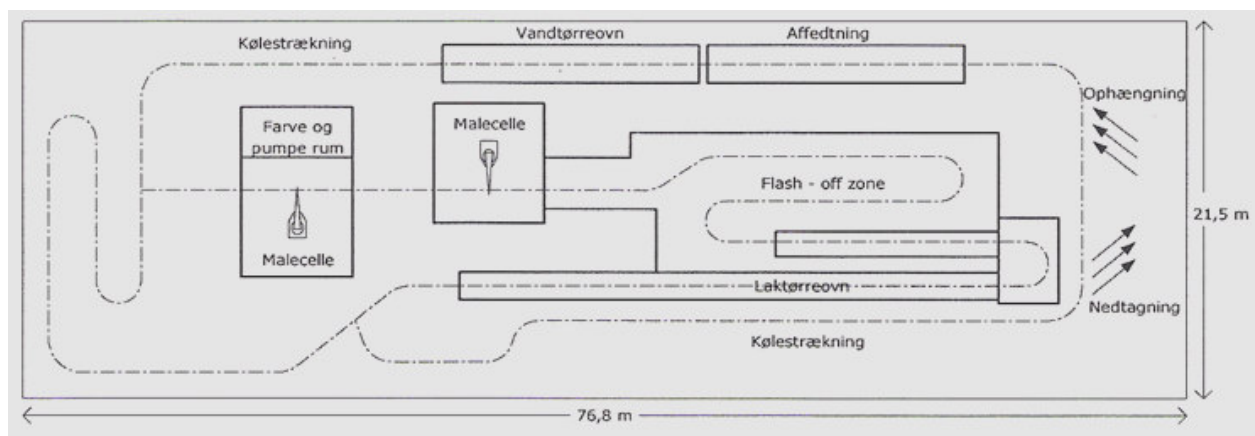
1 INDLEDNING

Dette projekt har overordnet til opgave, at udarbejde analyse og design dokument på casen ”Programmering af malerobot” ved brug af Objektorienteret analyse og design metoden beskrevet i [MAT et al, 2001].

Behovet og formålet med programmering af malerobot kan generelt beskrives ved, at de nuværende robot programmer mangler fleksibilitet, brugervenlighed og endelig er det økonomisk spørgsmål om de små/mellemstore virksomheder kan og vil investere i den nuværende produkter.

Overordnet er formålet således at udvikle et generelt værktøj til effektiv programmering af maleanlæg, hvor der tages udgangspunkt i casen: ”Crisplant Århus” [AAUProjekt, 2000].

Crisplant Århus er en produktions virksomhed der producere anlæg til sortering af pakker der hovedsagelig bruges i postcentre og lufthavne. Komponenterne der indgår i sorteringsanlæggene vådlakeres af hensyn til korrosion eller af kosmetiske årsager. Crisplant anvender malerobotter til udførelse af denne maleopgave. Figur 1.1 viser skematisk grundplan af maleafdelingen.



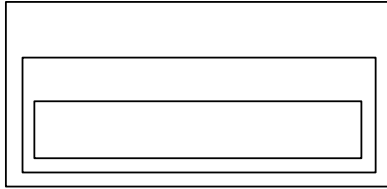
Figur 1.1

Maleafdelingen er opbygget omkring et maleanlæg der overordnet består af et conveyer system med troljefremføring. I Ophængning hænges emner på troljer. I malecelle foretages der manuel samt robotmaling.

1.1 Styresystemer

Maleafdelingen har sine egne styresystemer Malesys og Mita/Robtalk. Disse systemer er koblet sammen med hele virksomhedens styresystem MPS (Material Planning System). MPS styrer hvornår malingen af en serie af emner skal påbegyndes. MPS indeholder informationer om materialeflow og allokering af produktionsudstyr. Til hver ordre er der et jobkort med stregkode som følger ordren rundt i virksomheden. Når operatøren er færdig med at ophænge malemnet aflæses stregkoden fra jobkortet i Malesys. Emnet bliver derefter tilskrevet en bestemt trolje og herefter kender Malesys emnets nøjagtige placering i maleanlægget. Når troljen med emnet

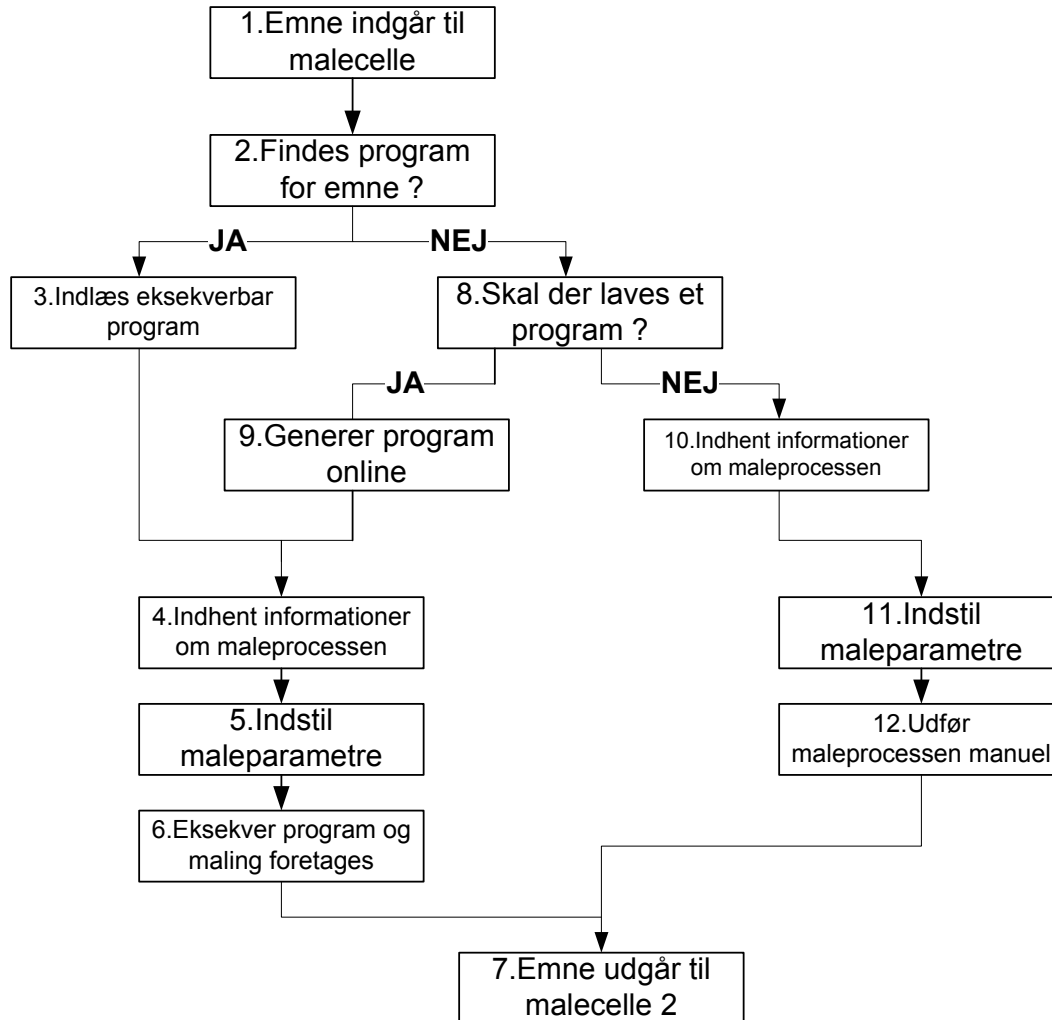
ankommer til malecellen, overføres der informationer fra Malesys til Mita om farve, sprøjtepipistol, dyse mm. Robtalk indeholder programmer til styring af malerobotten. Det er bevægelser, tænd/sluk af sprøjtepipistol, valg af sprøjtebane. Disse parametre lagres i robotprogrammer via metoden Teaching som nævnes i det følgende. Figur 1.2 viser opbygningen af de enkelte styresystemer i Crisplant.



Figur 1.2 Styresystemer

1.2 Arbejdsrutiner

Ifølge Crisplant er der 3 (male)arbejdsrutiner der kan gennemføres afhængig af komponenten/stk. størrelsen og programmerings fleksibiliteten. Mange af komponenterne er kundetilpassede og der male ofte i små serier på 5-100 stk. Hver programmering af robotten tager fra 10 til 120 minutter og under programmeringen står robotten standby, dvs. der er kun mulighed for on-line programmeing. Ved specialkomponenter er der ofte ikke rentabelt at benytte robotten da tidsforbruget til programmering overstiger programmeringstiden. 1.malearbejdsrutine er når emnet (komponenten) er placeret i malecellen og klar til at blive malet. Hvis der eksisterer et maleprogram udvælges det af maler og eksekvering starter. 2.malearbejdsrutine køres når der er tale om et nyt emne som ikke er robotmalet før. Maleren laver i dette trin et nyt program via on-line programmering af robotten. Denne programmering foregår ved at maler 1 gemmer robotten banekurver ud fra maler 2 fysiske placering af robotten male m. Når alle banekurver er gennemløbet kan robotten startes. 3.malearbejdsproces udføres når det vurderes at rutine 1 og 2 ikke er hensigtsmæssig. Figur 1.3 viser de 3 nævnte malearbejdsrutiner. Ifølge Crisplant er maleanlæggets belægningsgrad på ca. 65% og i kr. er det 4.000,00 pr. time når anlægget står stille. Crisplant har derfor efterspurgt en løsning, som kan øge udnyttelsen af robotterne på deres maleanlæg.



Figur 1.3 Malerarbejde rutiner

1.3 Problem

Forfatterne bag denne rapport har kendskab til systemer og udviklingsprocesser bag disse, i Nykredit og Inropa, som ikke i alle henseende kan betegnes som succesfulde. Der kan nævnes mangelfuld dokumentation, store vedligeholdelses-omkostninger og brugere der ikke fået indfriet deres forventninger til systemet.

Der er derfor i Nykredit løbende en styret proces i gang (Inropa har en mere uformel proces) der søger at optimere udviklingsprocessen med specifikke tiltag rettet mod brugen af værktøjer og processer.

Ved at følge kurset i SAD (System Analyse og Design) har forfatterne fået det indtryk at metoden [MAT et al, 2001] muligvis kan hjælpe med at løse nogle af de ovennævnte problemstillinger.

Dette søger vi at belyse, ved at benytte metoden på en aktuel case i Inropa A/S. Endvidere foretages en vurdering af fordele/ulemper ved indførelse af metoden, dog begrænset til enkelte elementer fra metode, i hhv. Nykredit og Inropa.

1.4 Behandling af problem

Vi vil følge de anvisninger som metoder angiver [MAT et al, 2001] på et aktuel systemudviklingsbehov i Inropa. Nykredit har også flere mulige analyse og design kandidater, men vi ser Inropa casen som et godt mix mellem kompleksitet og mulighed for at foretage en fornuftig afgrænsning.

Der foretages evaluering af produktet, i dette tilfælde analyse og design dokument og processen der giver det omtalte produkt. Der foretages endvidere en kort opsamling med henblik på en vurdering af brugen af OOAD-metoden.

1.5 Afgrænsning

For at afgrænse nærværende projekt, er fokus lagt på analyse og design faserne og der foretages ikke en egentlig kodning, test og afprøvning (implementering) af det specificerede system.

Projektet beskæftiger sig ikke med alle aspekter af OOAD i [MAT et al, 2001] , men der udvælges specifikt de område som vi finder nødvendige og formålstjenlige til det aktuelle behov. De valgte områder begrundes under de enkelte afsnit.

De nævnte virksomheder Nykredit og Inropa er på mange måder vidt forskellige (f.eks. organisation, historie, produkter og konkurrencesituation) og der kunne skrives meget om, hvad der kendetegner en virksomhed af denne specifikke type og foretages sammenligning mellem dem. Dette har vi valgt ikke at gøre.

Under analyse vil vi ikke beskrive klynger og anbefalinger, da disse afsnit ikke er relevante i denne sammenhæng. Under design vil vi ikke beskrive proces arkitektur og anbefalinger, da vi ikke ser, at det er relevant for os i denne sammenhæng.

1.6 Rapportens bagmænd

Til daglig arbejder projektdeltagerne bag dette EVU-projekt i hhv. Nykredits Koncernudvikling og –IT afdeling og Inropa A/S (Intelligent Robot Painting). Nykredit Koncernudvikling er en stor intern IT-udviklingsafdeling med knapt 300 ansatte. Størsteparten af udviklingsprojekterne udvikler specialsoftware til virksomhedens egne behov, og projekterne beskæftiger sig oftere med videreudvikling frem for nyudvikling. I Inropa A/S er der ansat 3 personer, hvoraf 2 personer forestår udvikling af Visual Basic applikationer til malerobotter.

I afsnit 5.1.1 gives der en kort beskrivelse af de 2 omtalte virksomheder.

Projektdeltagerne har ingen eller kun lidt kendskab til brug af objektorienteret analyse og design. Med den stigende brug af Web applikationer i Nykredit og udviklingspotentialer hos Inropa, er der i begge virksomheder håb om effektiv udvikling ved brug af metoden der nævnes i [MAT et al, 2001] .

1.7 Kapiteloversigt

Kapitel 1 indeholder beskrivelse af virksomheden og problemområdet.

Kapitel 2 indeholder analyse af Problemområde

Kapitel 3 indeholder analyse af Anvendelsesområde

Kapitel 4 indeholder Designdokumentet

Kapitel 5 indeholder vurdering ved indførelse af OOAD elementer i Nykredit hhv. Inropa

Kapitel 6 indeholder generelle kvalitative oplevelser ifm. udarbejdelse af denne rapport

Kapitel 7 indeholder konklusion/perspektivering

2 OPGAVEN

2.1 Formål med systemudviklingsprojektet

Som nævnt under indledning kan der skabes en mere effektiv maleproces ved at kunne programmere malerobotten online. Der skal udvikles et simpelt pc-værktøj der kan simulere maleprocessen og dens elementer (fysiske ting i maleafdelingen såsom ophængning og robotens bevægelser). Robotprogrammet skal kunne gemmes og genbruges når behovet opstår. Til Robotprogrammet kan der tilknyttes print af ophængning (her tænkes der på en mappestruktur som windows stifinder) og andre elementer så dokumentation er på plads og viden kan bruges ved næste maleopgave.

2.2 Systemdefinition

Et PC baseret programmeringsværktøj til maleanlæg, som effektivt kan anvendes til at lave robotprogrammer, uden at standse malelinien. Dette skal øge opptiden for maleanlægget væsentligt, når manuel maling er ude af betragtning.

Værktøjet skal være let at bruge, samt hurtigt at sætte sig ind i, for en bruger som er trænet i at arbejde med malerobotter.

Det skal også være muligt at importere programmer fra robotter, så de kan editeres i værktøjets grafiske programmeringssprog, og skrives ud igen med eventuelle ændringer, uden eventuelle kommentarer og ikke understøttet robotprogramkode går tabt.

Udover en række 3D-punkter med tilhørende orientering, indeholder et robotprogram, også en lang række parametre, som værktøjet skal understøtte. Det skal være muligt for brugeren, at organisere disse parametre i et ønsket antal parameteret, som kan anvendes på robotprogrammet, for let og hurtig skift til en anden konfiguration af male specifik setup af hvorledes malingen påføres emnet. F.eks øget fanebrede eller mere tryk på malingen.

Værktøjet skal også understøtte et antal events, som f.eks. tænd og sluk for malingen. Antal parametre og eventsignaler, variere fra robot til robot. Det er derfor nødvendigt med en dynamisk struktur, som indeholder en model af og en specifikation for de kendte robottyper. Antallet af kendte robottyper skal være dynamisk.

Værktøjet skal også kunne håndtere maleanlæg med flere ens eller forskellige robotter, fordelt over en eller flere maleceller. Til mange robotter findes der flere forskellige typer malepistoler, som alle understøtter et antal varierende parametre fra robotens maleudstyr. Værktøjet skal også indeholde en model af- og en specifikation for de kendte malepistoler, og på en overskuelig måde, sammenkæde dem med den robot de tilhøre.

Brugeren skal have mulighed for at tilføje robotter og pistoler til systemet, samt anvende forskellige pistoler på robotterne i robotprogrammerne. Programmerne organiseres i 'PaintJobs' det svare til en ophængning, af et givet antal emner på en trolje, som skal males. Disse jobs kan udprintes og gives til en ophænger som ved at følge henvisninger kan initiere en ny malerobot session.

2.3 BATOFF-kriterier

Betingelser:

Ny udviklingsmetode benyttes og brugen skal tilrettes personale i maleafdeling som ikke nødvendigvis har viden om pc og brugen af de programmer som kan afvikles herfra. Det økonomiske niveau til udvikling af programmer er lav-budget og i dette tilfælde en arbejdsindsats i størrelsesorden 100 md, svarende til kr. 500.000,00.

Anvendelsesområde:

Maler, værkfører i maleafdeling

Teknologi:

Pc/server windows C++, C# software udviklet ud fra analyse og design dokumenter udarbejdet ved brug af objektorienteret analyse/design metode

Objekter:

Malerobot, ophængning, nedtagning, maleanlæg og celle, trolje, malepersonale

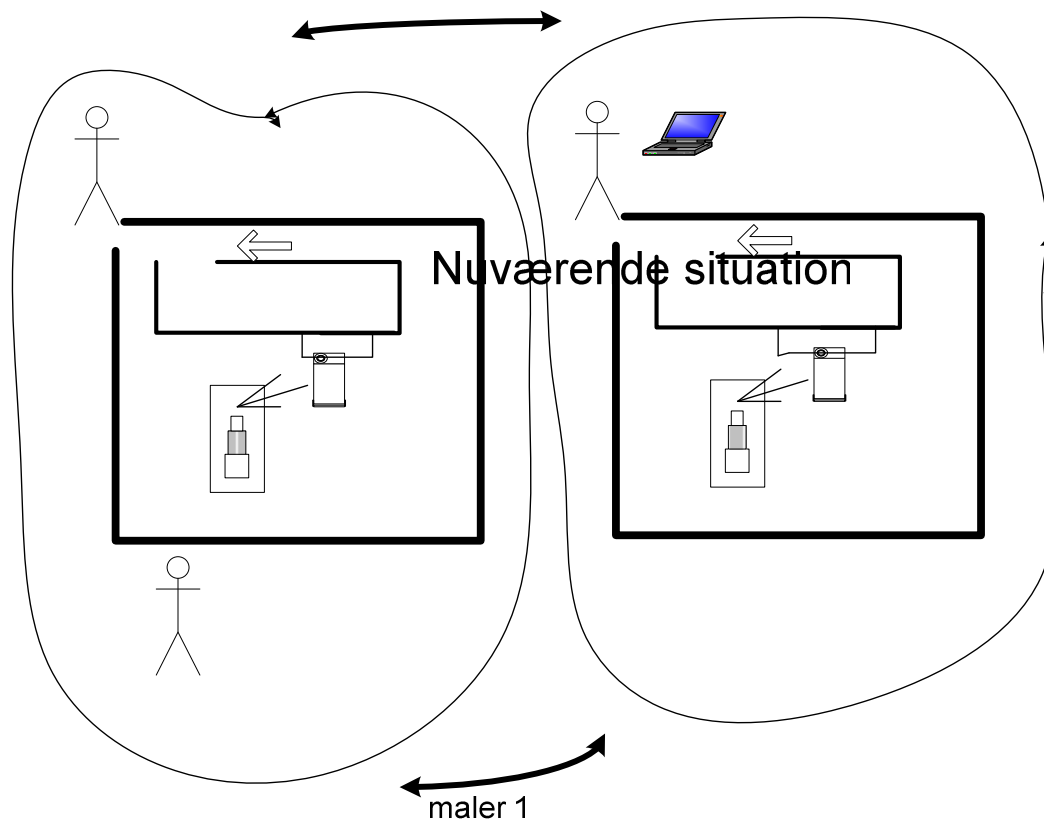
Funktioner:

Opret Bruger, Login, Opret Malejob, Programmer Malejob, Eksporter robotprogram, Print Malejob, Ophængning

Filosofi:

Udvikling af software til pc, der gør det muligt at on-line programmere malerobot og opnå større uafhængighed mellem malearbejdsopgaver og større fleksibilitet i planlægning af samme arbejdsopgaver.

2.4 Omgivelser – rige billeder



Figur 2.1 Maleprocessen nu og i fremtiden.

Figur 2.1 illustrerer at malepersonalet (maler 1 og 2) i de den nuværende arbejdsproces udfører manuelle arbejdsopgaver for, at malerobotten kan foretage en maleopgave. I den ønskede fremtidige arbejdsituation benytter maler 1 en bærbar PC, hvor der er installeret program, der kan hjælpe maler 1 med at afklare/beslutte om den påtænkte maleproces skal foretages manuel dvs. uden brug af malerobot eller via eksisterede maleprogram (hentet frem via søge facilitet) eller der skal indkodes et nyt maleprogram der kan gemmes på Pc'en.

2.5 Klasser

Malerobot

I det følgende opremses identificerede klasser med en ja/nej markering om klassen vil blive behandlet yderligere i analyse og design fasen.

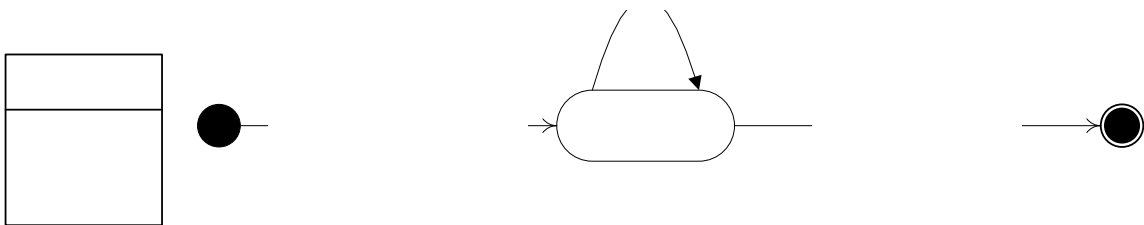
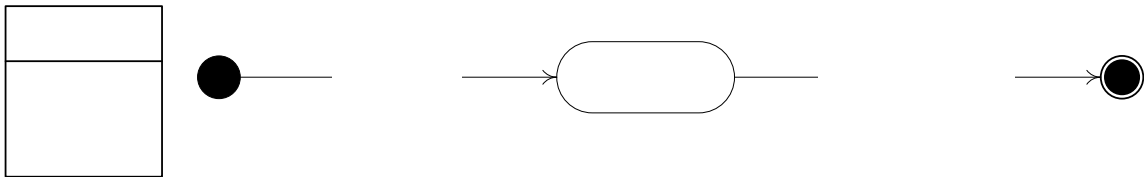
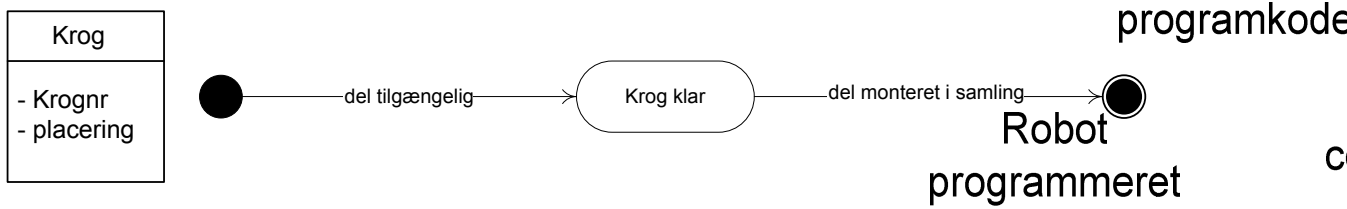
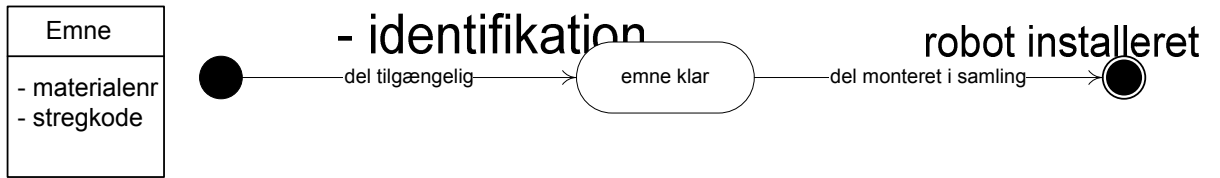
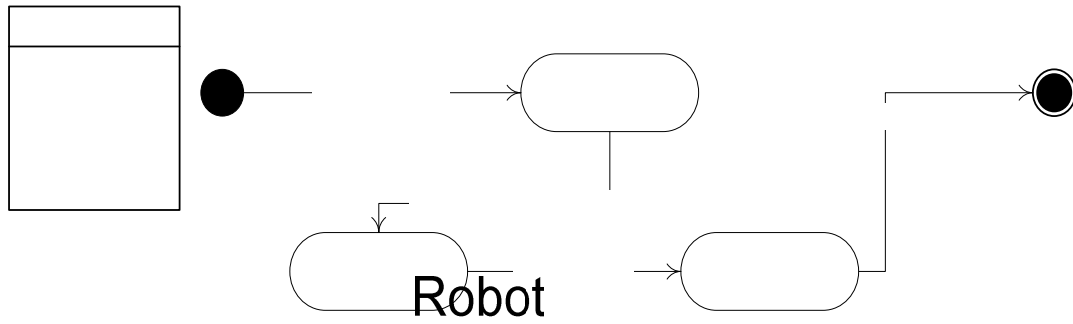
Klasse	Beskrivelse/vurdering Hvorfor er denne klasse med i afklaringen?	Ja / nej
Maler	Funktionen maler, manuel maler eller operatør ?, Vi vil registrere oplysninger om maler under klassen bruger (se nedenfor)	Nej

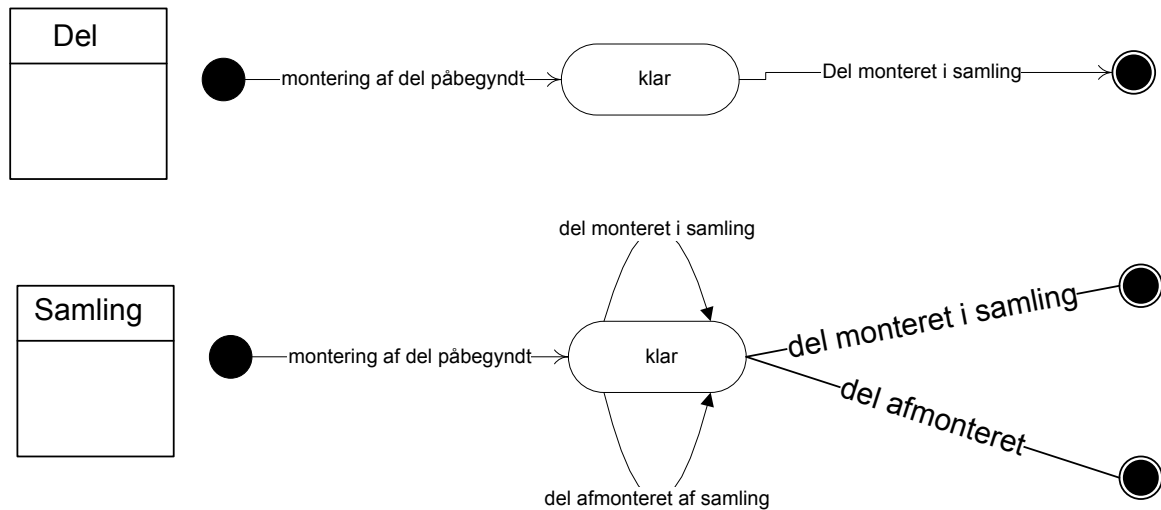
Robot	Forskellige robotter der kan betjenes eller programmeres. En robot kan entydigt bestemmes via dens art og type.	Ja
Robottype	En robot af mærket ABB f12...dette er ikke en klasse	Nej
Bruger	Bruger er fællesbetegnelse for maler, operatør, værkfører	Ja
Maleanlæg	Består af malecelle og ydre anlæg. Der er behov for kunne tilknytte en bruger til en specifik robot i et maleanlæg.	Ja
Malejob	En entydig beskrivelse af et malejob, dvs. med emne, trolje, maleanlæg	Ja
Celle (malecelle)	Består af robot, pistol og ophæng, trolje. De enkelte klasser og deres indbydes afhængigheder skal kunne styres.	Ja
Malepistol (+dyse)	Forskellige type af malepistol til forskellige ”arbejdsopgaver”. En malepistol kan entydig bestemmes ud fra nummer, fabrikat.	Ja
Arbejdsseddel	Beskrivelse af hvad hvilket emne der skal males og hvordan. Arbejdsseddel er et grundlæggende begreb i virksomheden og muligheden for at have arbejdsseddel gemt elektronisk kan være en hjælp. Vi har valgt at afgrænse os fra denne del.	Nej
Arbejdsopgave	Hvad er det operatøren foretager? Denne klasse er med under klassen arbejdsseddel. Ifølge BATOFF er det ikke et planlægningsværktøj der skal udvikles og derfor fravælges denne.	Nej
Emne	Det materiale der skal males og defineret på arbejdsseddel. Et emne er entydig bestemt. Et emne skal kunne genkendes, således at det kan søges ud og der kan findes et maleprogram til det.	Ja
Fixtur	Beskriver hvordan emnet fastgøres til troljen. Er afgørende for at kunne starte et malejob	Ja
Trolje	Betegner det materiale som emnet skal hænges op på via kroge. Troljens udformning bestemmer hvad der kan hænges op på den og dermed hvilke emner der kan robotmales.	Ja
Ophængning	Det område hvor emnet hænges op før maleprocessen begynder. En ophængning er et samlet betegnelse og er måske blot et sted i maleafdelingen? Det er en delmængde af trolje og derfor udelades den.	Nej
Krog	Fra en trolje er der kroge som emnet kan hænges/placeres på. Det er vigtigt at den rigtige krog er placeret det rigtige sted på troljen. Hvis dette ikke er tilfældet kan der opstå fejl under maleprocessen og processen skal muligvis tages om.	Ja
Nedtagning	Det område hvor emnet tages ned efter maleprocessen er afsluttet. Det er en delmængde af trolje og derfor udelades den. Der er ikke behov for at registrere oplysninger om hvornår maleprocessen er udført.	Nej
Program	Indeholder instruktioner i form af robotbevægelser	Ja
Bevægelse	Robotbevægelser er instruktioner på formen: flyt robot arm a cm til position x,y,z.	Ja
Malefane	Beskriver tykkelse, tryk, puls af malepistol	Nej
Konfiguration	Entydig beskrivelser af paintset med parametre som farve og varighed	Ja
Begivenhed	Angiver styring af konfigurationen	Nej

2.6 Adfærdsmønstre

En beskrivelse af mulige hændelser for alle objekter i en klasse.

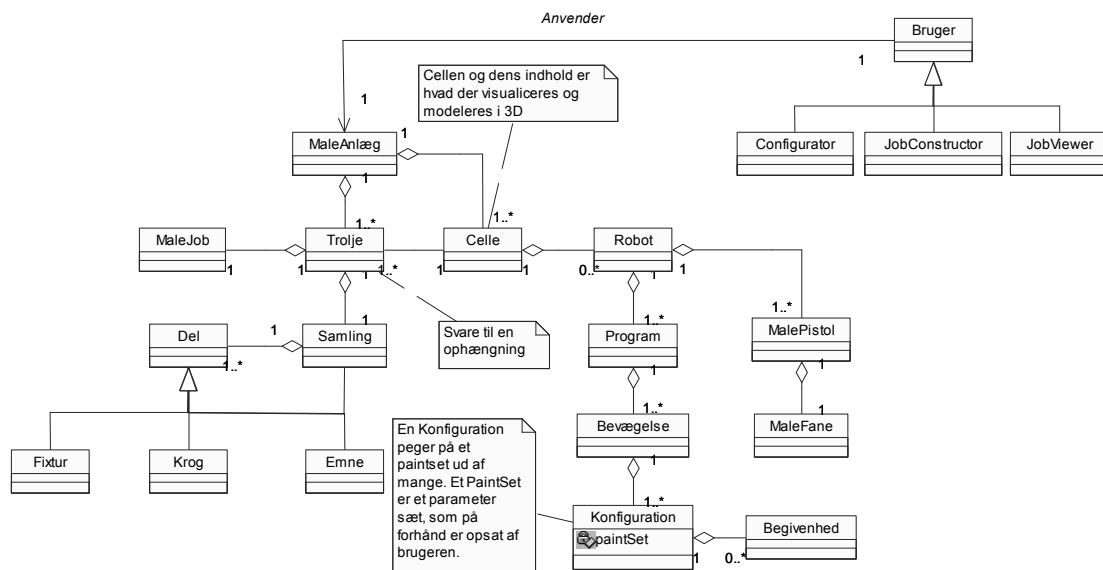
I det følgende beskrives tilstandsdiagrammer for udvalgte klasser. De følgende tilstandsdiagrammer er valgt fordi de er grundlæggende og dermed danner fundamentet.





2.7 Struktur klassediagram

Klassen Bruger er en generalisering af Configurator, jobconstructor og jobviewer. Et objekt af klassen Bruger anvender Maleanlæg, hvor der er associeret en trolje. På en trolje er der i problemområdet identificeret objekterne Fixtur, krog og emne. Opbygningen og sammenhængen mellem disse objekter er udarbejdet i samlingsmønstret efter inspiration fra [MAT et al, 2001].



Figur 2.2 Struktur klassediagram for problemområdet.

2.8 Hændelser

I det følgende nævnes de hændelser som opfylder de nævnte krav:

Ligger hændelsen indenfor systemdefinitionen? Er hændelsen øjeblikkelig og atomer? Kan hændelsen identificeres når den indtræffer?

Hændelsestabel:

Hændelse/Klasser	Robot	Program	Malepistol	Celle	Maleanlæg	Bruger	Trolje	Fixtur	Emne	Krog	Konfiguration	Del	Samling	Malefane
Ophængning				+	+	*	+	+	+	+		*	*	
Configuration af maleanlæg	+		+	+	+	*	+				+			+
Importer programmer	+	+				*					+			
Editer i grafisk brugergrænseflade		*				*		*	*	*	*	*	*	
Ændre parameter i paintset						*					*			
Oprette/slette paintset						*					*			
Tilknyt paintset til punkt i robotprogram		+				*					+			
Tilføj/fjern robot	+			+		*								
Tilføj/fjern malepistol	+		+			*								+
Tilføj/fjern malecelle				+	+	*								
Ophæng fixtur,kroge, emner på trolje						*		+	+	+		*	*	
Opret, slet, åben, gem malejob		+			+	*		+	+	+	+			
Start/stop simulering af malepistols bevægelse			+			*	+							
kopier, klip, flyt, slet fixtur, kroge, emner og robotprogram	+	+				*		+	+	+				
Nedtagning					+	*	+	+	+	+				

+ : hændelse kan forekomme 0 eller 1 gang. *: 0 eller flere gange.

Figur 2.3 Hændelsestabel

3 ANVENDELSESOMRÅDE

3.1 Aktørtabel

I figur 3.1 nedenfor vises aktørtabel som angiver interaktionen mellem den enkelte aktør og brugsmønstre. Brugsmønstrene bliver normalt udvalgt og godkendt i samarbejde med brugeren. Her har vi valgt et sæt af brugsmønstre, så vi overordnet har en hel arbejdsgang.

Formål med Jobkonstruktør: fuld legalisering til systemet, opretter nye robotprogrammer og er ansvarlig for opbygning og programmets struktur.

Formål med Jobviewer: Læse adgang til systemet og henter eksisterende robotprogrammer frem til eksekvering.

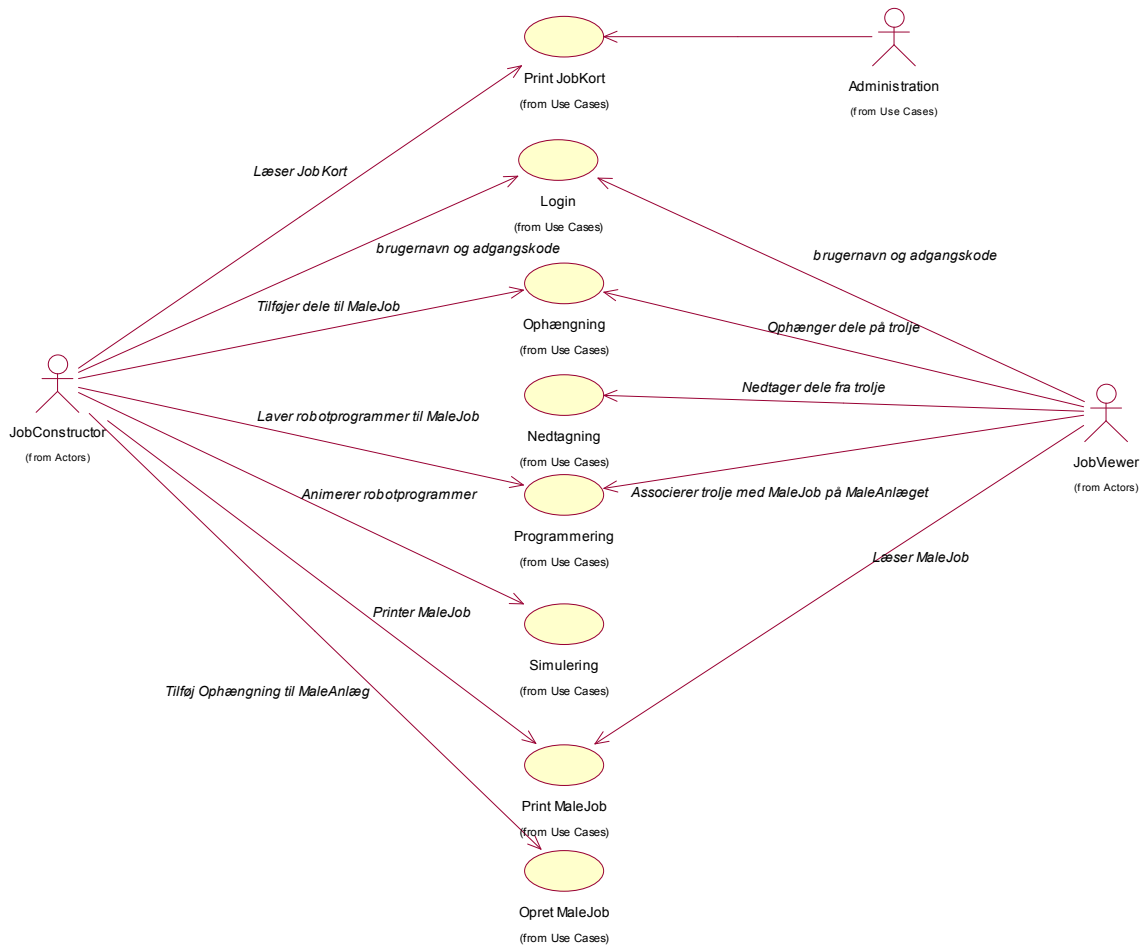
Formål med Adminstrator: Ingen adgang til systemet. Aflevere oplysninger til arbejdsedel (jobkonstruktør).

Brugsmønstre/Aktør	Job-konstruktør	Job-viewer	Administrator
Print jobkort til brug for maleprocessen	V		V
Login til pc-program	V	V	
Opret, ret malejob	V		
Ophængning	V	V	
Nedtagning		V	
Programmering	V	V	
Simulering (afprøvning:robotbevægelser over emnet)	V		
Print malejob	V	V	

Figur 3.1 Aktørtabel for malerobotprogrammet

3.2 Brugsmønsterdiagram

Det følgende brugsmønsterdiagram viser i store træk den samme information som aktørtabellen blot er relationen mellem aktør og deres brug af systemet vist grafisk i UML notationen.



3.3 Brug

3.3.1 BRUGSMØNSTRE

I det følgende er der specificeret brugermønstre for disse

- Opret Malejob
- Login
- Opret Bruger
- Programmer Malejob
- Eksporter robotprogram
- print Malejob
- Ophængning

Brugermønster:Opret MaleJob

Aktør: JobConstructor

Formål: Konstruerer en model af en fremtidig ophængning på det fysiske anlæg.

Type: Primær og essentiel

(primær: brugsmønster benyttes meget ift. sekundær: benyttes af og til)

Essentiel: overordnet beskrivelse ift. teknisk: teknisk beskrivelse

Overblik: JobConstructor opretter nyt MaleJob i systemet, hænger emne(r) , kroge og fixtur(er) op på en trolje, i et PC baseret offline miljø, som er en model af den fysiske ophængning.

Kryds reference: Programmer, Login.

Typiske hændelser:

Aktør:

System:

1. Logger ind i systemet.	2. Validerer login.
3. Opretter nyt MaleJob.	4. Registrerer oprettelsen og beder brugeren om info vedrørende MaleJob navn.
5. Svare på 4. og bekræfter.	6. Validerer og gemmer informationerne.
	7. Henter en tom trolje frem, samt en model af malecellen, og gør klar til at brugeren kan indhente og ophænge modeller af emner, kroge og fixture på troljen.*
8. Indhenter emne, krog, eller fixtur.	9. Tilføjer model af indhentet geometri til MaleJob.*
10. Gentager 8, eller flytter/ roterer geometri model(er) på plads.	11. Moddeler flytning/rotation af geometri i MaleJob.
12. Afslutter Opretning af MaleJob	13. Gemmer MaleJob.

*Alle emner, kroge og fixture bærer selv deres eget navn, som også vil blive anvendt i MaleJobbet. Opretning og navngivning af robotprogrammer og paintsets er også muligt.

Funktioner: opret, udvælg parametre, tilknyt robottype, læs parametre, beregn geometri

Brugsmønster: Login

Aktør: JobConfigurator, JobConstructor, JobViewer.

Formål: Give en bruger adgang til systemet.

Type: Primær og essentiel

Overblik: Brugeren logger ind og får adgang til bestemte funktioner, knyttet til bruger typen.

Kryds reference: Opret nyt MaleJob, osv.

Typiske hændelser:

Aktør:

System:

1. Går til Login.	2. Beder brugeren om brugernavn og adgangskode.
3. Svare på 2. og bekræfter.	4. Validerer informationerne og giver brugeren adgang til systemet.

Funktioner:

Valider login, Gem login

Brugsmønster: Opret Bruger

Aktør: Configurator, JobConstructor

Formål: Give nye brugere adgang til systemet.

Type: Sekundær og essentiel

Overblik: Configurator eller JobConstructor opretter en ny bruger i systemet, de kan kun oprette bruger med samme eller lavere bruger niveau. Dvs. at en ikke JobConstructor kan oprette en Configurator.

Kryds reference: Login.

Typiske hændelser:

Aktør:

System:

1. Logger ind i systemet.	2. Validerer login.
3. Går til opretning af ny bruger.	4. Beder brugeren om brugernavn, adgangskode, og brugerniveau til den nye bruger.
5. Svare på 4. og bekræfter.	6. Validerer og opretter den nye bruger, samt gemmer informationerne.
7. Gentager 3. eller forlader opretning af ny bruger.	

Funktioner: Opret bruger.

Brugsmønster:Programmer MaleJob

Aktør: JobConstructor

Formål: Konstruerer en model af fremtidig robotprogrammer til de fysiske robotter.

Type: Primær og essentiel

Overblik: JobConstructor konstruerer modeller af robotprogrammer til robotterne i modellerede malecellen, som er en model af den fysiske malecelle.

Kryds reference: Opret MaleJob, Login.

Typiske hændelser:

Aktør:

System:

1. Henter MaleJob ind i systemet.	2. Viser MaleJob for brugeren.
3. Vælger robot.	4. Registrerer valg af robot.
5. Indsætter punkt i robotprogram.	6. Tilføjer punkt til robotprogram og viser det samlede program for brugeren.
7. Gentager 5, eller flytter et punkt i maleprogram.	7. Flytter punkt og viser resultatet for brugeren.
8. Gentager 3, 7 eller afslutter programmering.	9. Gemmer programmer til MaleJob.

Funktioner:

Læs Robotprogram, opret, ret, slet RobotProgram.

Brugsmønster: Eksporter Robotprogrammer

Aktør: JobConstructor

Formål: Lave de grafisk programmerede programmer om til rigtig robotprogrammer og eksporterer dem til de fysiske robotter.

Type: Primær og essentiel

Overblik: JobConstructor vælger at eksporter robotprogrammerne til de fysiske robotter. Systemet konverterer de grafiske programmer til rigtige robotprogrammer der er tilpasset den enkelte robottype, og som enten overføres automatisk til robotterne eller til en flytbar lagringsenhed, som JobConstructor tager med til de fysiske robotter, hvor han overføre programmerne manuelt.

Kryds reference:

Typiske hændelser:

Aktør:

System:

1. MaleJob færdiglavet.	2. Viser MaleJob for brugeren.
3. Vælger at eksporter robotprogrammerne til de fysiske robotter.	4. Registrerer valg, konverterer programmer til rigtige robotprogrammer, som enten overføres automatisk til robotterne eller til en flytbar lagringsenhed.

Funktioner:

Eksporter RobotProgrammer.

Brugsmønster:Print MaleJob

Aktør: JobConstructor eller JobViewer

Formål: Udskrive en oversigt over en ophængning.

Type: Primær og essentiel

Overblik: Vælger at udskrive MaleJob på printer. Systemet laver en eller flere billeder af den virtuelle ophængning, og samler info. Vedrørende antal emner og deres produktId, antal kroge der er anvendt og deres længde, fixture og tilhørende fixturId, robotId'er, samt robotprogramId'er. Hvorefter det udskrives på papir, der senere kan anvendes som anvisning til en ophængning.

Kryds reference: Ophængning

Typiske hændelser:

Aktør:

System:

1. MaleJob færdiglavet.	2. Viser MaleJob for brugeren.
3. Vælger at udskrive MaleJob på printer.	4. Registrerer valg, generer billeder, samler info. og udskriver MaleJob.

Funktioner:

Læs MaleJob navn, Læs Emne, læs Fixtur, læs krognr, læs robot navn.

Brugsmønster: Ophængning

Aktør: JobViewer

Formål: Foretage en ophængning på det fysiske anlæg.

Type: Primær og essentiel

Overblik: JobViewer foretage en ophængning på det fysiske anlæg, ved at printe et MaleJob og følge dets anvisninger.

Kryds reference: Print MaleJob.

Typiske hændelser:

Aktør:

System:

1. Logger ind i systemet.	2. Validerer login.
3. Vælger at åbne et MaleJob.	4. Registrerer valg, henter MaleJob frem til brugeren.
5. Vælger Print MaleJob.	6. Printer MaleJob.
7. Følger anvisninger angivet på MaleJob-printseddel. Dvs. finder de emner kroge og fixture angivet på papirets stykliste, og anbringer dem på en trolje, som anvist.	
8. Fortæller maleanlægget at MaleJobbet er placeret på TroljeID.*	

* På nogle maleanlæg skal jobnavnet svare til robotprogrammnavnet, det vil også sige at alle robotprogrammer (et til hver robot) skal hedde det samme på alle robotter. Dette er gældende på Crisplant. Denne restriktion skal kunne, slås til og fra i systemet.

Funktioner:

PrintMaleJob.

3.4 Funktioner

Her fokuseres der på HVAD systemet skal gøre for aktøren. De enkelte funktioner opdeles i funktionstyperne: opdatering, signalering, aflæsning og beregning. En funktion skal på den ene side være tilpas detaljeret så aktøren (brugeren) kan forholde sig til indholdet, men i visse situationer skal funktionerne kun beskrives på et mere overordnet niveau.

Funktion	kompleks – medium – simpel	Funktionstype
Opret bruger	Simpel	Opdatering
Login		
- valider login	Simpel	Aflæsning
- gem login info	Simpel	Opdatering
Print Malejobkort		
- læs krognr, emne, fixtur,	Medio	Aflæsning

robotnavn		
Programmering		
- tilknyt robottype	Simpel	Opdatering
- opret, udvælg parametre	Kompleks	Opdatering
- opret, slet, robotprogram	Simpel	
- søg robotprogram	Medium	Aflæsning
Malejob simulering		
- læs parametre	Simpel	Aflæsning
- beregn geometri	Kompleks	Aflæsning, beregning
- afspil animation	Kompleks	Aflæsning, beregning
Ophængning	Kompleks	Opdatering
Eksporter robotprogram		
- indlæs konverteringsparametre	Simpel	Aflæsning
- konvertering	Kompleks	Opdatering

3.5 Dialogform

Vi har valgt menuvalg til navigering og skemaudfyldelse til indtastning af data for derved at hjælpe brugeren ift. kommandosprog.

De enkelte vinduer der nævnes nedenfor kan ses i ”naturlig” størrelse i Appendix.

Vindue Hoved: ved valg af file åbnes Overordnet menu

Vindue Overordnet menu:

Vindue Nyt malejob: Under Navn indtastes navn på det nye malejob og placering på drev udvælges under Placering. Når det er ok trykkes knap OK og malejob gemmes.

Vindue Aktiv malejob:

Vindue Bruger admin: Brugernavn, adgangskode og niveau indtastes. Ved tryk på Gem bruger eller slet Bruger foretages oprettelse eller sletning. Validering af indtastningsfelter foretages.

Vindue Login: Brugernavn og adgangskode indtastes og hvis denne er af rollen Configurator vil knappen Administrere brugere blive tilgængelig ellers er den lukket for aktivering.

Vindue Simulering: Grafikmodul hvor der er knapper til at vise robotsimulering. Med knapper kan sekvens afspilles igen.

Vindue Generere robotprogram: viser den tidsmæssige udstrækning af genereringen. Vindue lukker ned når den er færdig.

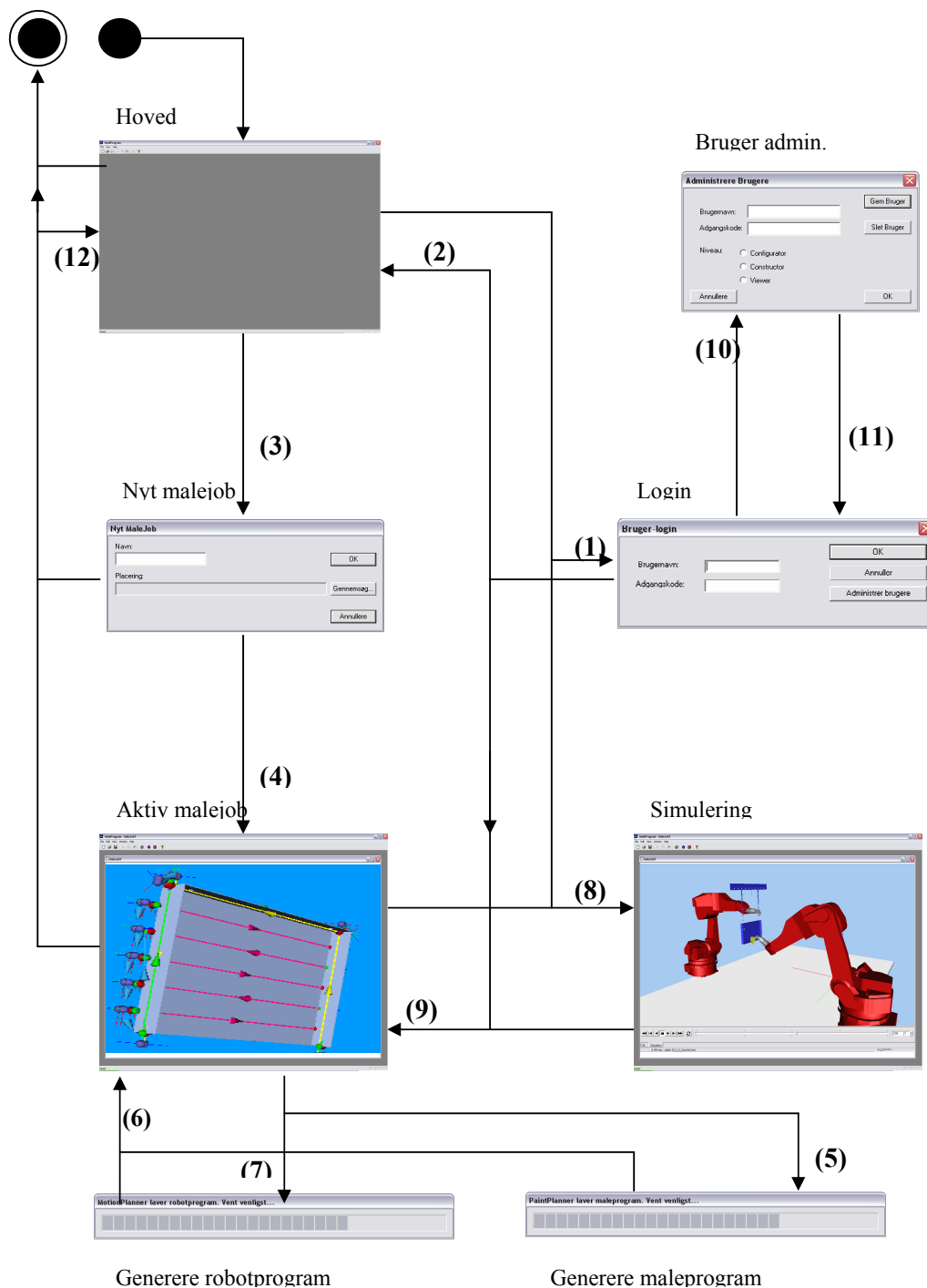
Vindue Generere maleprogram: viser den tidsmæssige udstrækning af genereringen. Vindue lukker ned når den er færdig.

3.6 Navigeringsdiagram

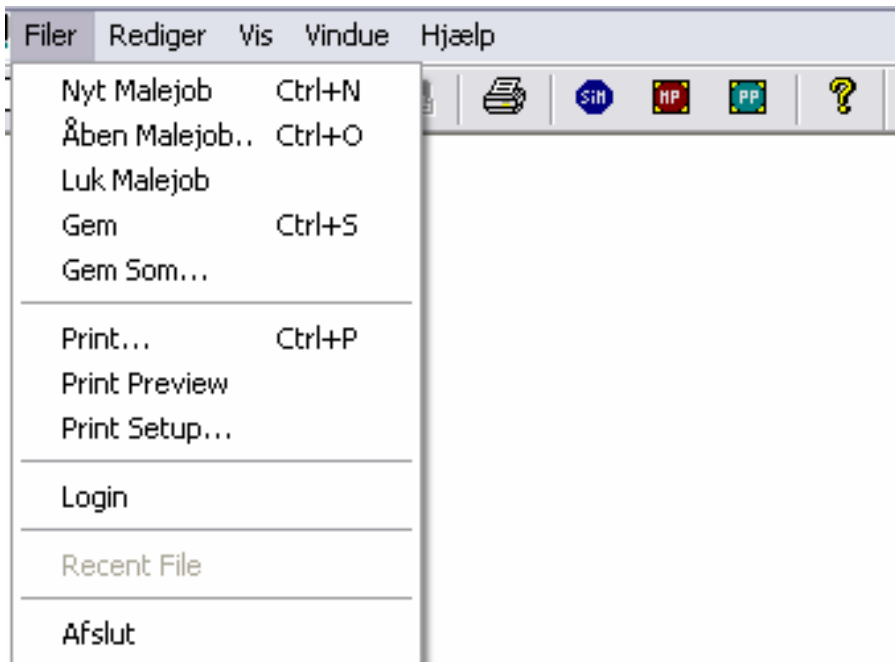
Tallene nedenfor henviser til dialogen i navigeringsdiagrammet.

- (1) Menu: Filer -> Login
- (2) Bruger Login: OK-knap, eller Annuller-knap.
- (3) Hoved: Nyt Malejob-knap, eller Menu: Nyt malejob.
- (4) Nyt Malejob: OK-knap.
- (5) Aktiv malejob: Kør PaintPlanner-knap.
- (6) Generere maleprogram: Færdig.

- (7) Aktiv malejob: Kør MotionPlanner-knap.
- (8) Aktiv malejob: Vis Simulering-knap.
- (9) Simulering: Tilbage-knap, eller Login: OK-knap, eller Annuller-knap.
- (10) Login: Administrere brugere-knap
- (11) Bruger Admin.: OK-knap eller Annuller-knap.
- (12) Aktiv malejob: Lukke-knap, Menu: Filer->Luk malejob, eller Nyt malejob: Annuller-knap.



Den følgende menu viser den overordnede menu, som kan benyttes fra de øvrige menuer.
Trykknapper SiM aktivere Simulering, MP aktivere MotionPlanner og PP starter PaintPlanner.



4 DESIGN

4.1 Formål

At skabe en mere fleksibel og effektiv maleproces i en industriel maleafdeling, ved at flytte programmeringen af malerobotterne, fra den fysiske robot, til et PC baseret maleværktøj. Værktøjet skal kunne anvendes parallelt med, at malelinen er aktiv, så det ikke længere er nødvendigt at standse hele maleanlægget, for at foretage programmering af robotterne. Værktøjet skal visualisere den fysiske malelinie i 3D, hvor programmeringen af robotterne foregår via tastetryk og museklik. Det skal være et let overskueligt værktøj, som en robotoperatør hurtigt kan sætte sig ind i. Robotprogrammet skal kunne gemmes og genbruges når behovet opstår. Arbejdet i værktøjet organiseres i malejobs, som indeholder grafisk visualisering af malleprogrammer, emner fisture og ophængningskroge, samt mulighed for visning af en samlet stykliste. Et malejob kan udprintes på et A4, til brug ved ophængning af de fysiske emner. Når et malejob er færdigt kan værktøjet kompilere og eksportere de grafiske maleprogrammer til rigtige robotprogrammer, som kan eksekveres på de fysiske robotter.

4.2 Kvalitetsmål

Systemet skal være nemt at bruge. Det skal være nemt for brugeren at danne sig et overblik over systemets funktioner og forstå deres sammenhænge. Et andet vigtigt kriterie er at brugeren kan stole på systemet og hermed få større tillid til brugen af det.

Systemet skal være dynamisk; det skal være nemt at ændre og tilføje funktionalitet til systemet. F.eks. understøttelse af nye robottyper.

Data der behandles i systemet er ikke fortrolige og kriterier som data-sikkerhed må betegnes som mindre vigtige.

Den tekniske platform er fastlagt på forhånd af kunden, idet den PC der benyttes i deres daglige arbejde skal benyttes. Kriterier som Flytbart er således ikke relevant i forbindelse med udvikling af systemet mht. operativsystemet. Det er derimod vigtigt at systemet udvikles således at det let kan tilpasses for skellige maleanlæg, da stortset alle anlæg vil være forskellige fra kunde til kunde.

Genbrug af komponenter er ikke essentielt for systemet, men kan benyttes i det omfang det kan lette udviklingen af systemet.

Systemet skal integreres til en MotionPlanner, som oversætter robotprogrammer fra det generelle grafiske robotsprog, anvendt i værktøjet, til den enkelte fysiske robot. Der skal her til laves en snitflade som kan kommunikere med denne MotionPlanner, i forbindelse med export og import af robotprogrammer. Systemet skal også integreres til en Robotsimulator, som kan vise en model af den fysiske robot og simulerer robotens bevægelser ved eksekvering af et robotprogram.

At Systemet er pålideligt er mindre vigtigt, da der er tale om offline programmering. Skulle systemet svigte kan det genstartes, og arbejdet kan fortsættes fra sidste gemning af et malejob. Et nedbrud må dog ikke forudsage, at større mængder data går tabt. For at sikre det implementeres en auto-gem-funktion, som sikre at der maksimalt går få minutters arbejde tabt.

Systemet skal være fleksibelt, f.eks. kunne med små ændringer kunne tilsluttes en anden MotionPlanner.

På et område skal systemet være testbart; Et program programmeret i værktøjet, skal kunne testes før det sendes til den fysiske robot. Dette skal ske ved at anvende en robotsimulator, som anvender samme snitflade som MotionPlanneren. Det skal også kunne testes at bevægelserne i simuleringen, svare til bevægelserne på den fysiske robot.

Det er vigtigt at programmerne oversættes korrekt.

Med udgangspunkt i disse betragtninger er følgende prioritering fastlagt.

Kriterium	Meget vigtigt	Vigtigt	Mindre vigtigt	Irrelevant	Trivielt opfyldt
Brugbart	X				
Sikkert			X		
Effektivt	X				
Korrekt		X			
Pålideligt			X		
Vedligeholdbart			X		
Testbart		X			
Fleksibelt		X			
Forståeligt	X				
Genbrugbart				X	
Flytbart mellem maleanlæg		X			

Flytbart mellem operativsystemer				X	
Integrerbart		X			

4.3 Teknisk platform

4.3.1 Udstyr

Hardwaremæssigt, vil systemet bestå af PC som arbejdsstation. Der kan anbefales krav til PC'ens ydeevne, mht. Processer, grafikkort og lignende. Dog skal systemet kunne eksekveres på en hver PC, der opfylder kravene til Microsoft Windows XP 32bit Edition.

4.3.2 Basisprogrammearbejde

Microsoft Windows XP 32bit Edition.

4.3.3 Systemgrænseflade

Til export og import af robotprogrammer mellem værktøjet og robotterne, skal systemet understøtte floppy-drev, FlashDisk og/eller netværks-drev, efter behov.

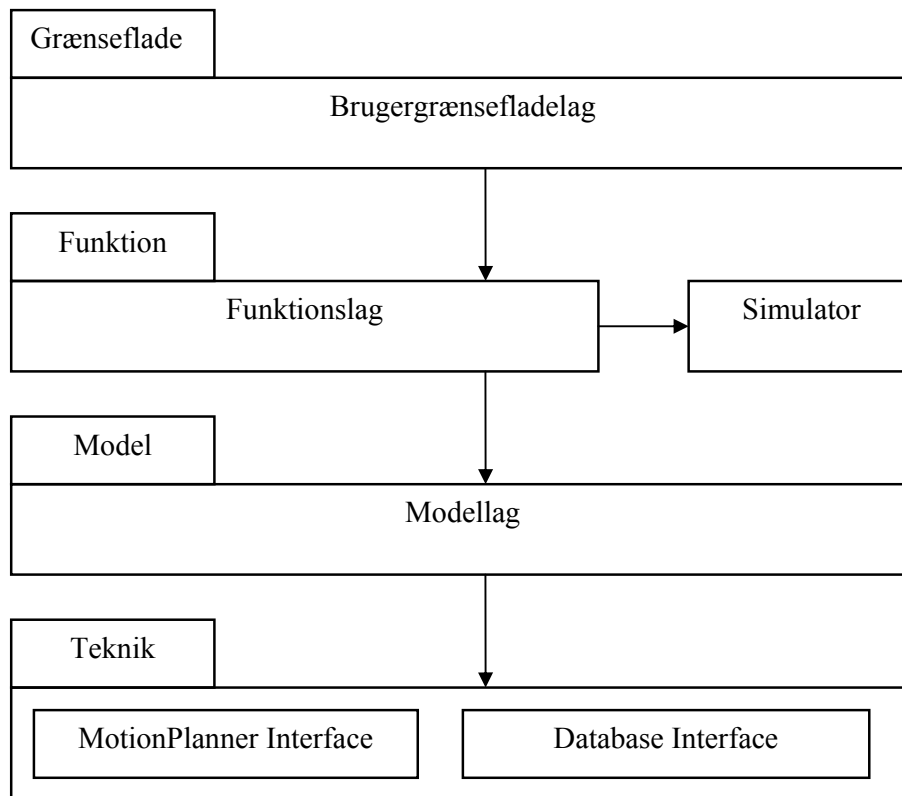
4.3.4 Designprog

Designsproget er baseret på UML standarden.

4.4 Arkitektur

4.4.1 Komponentarkitektur

Vi har valgt at benytte den lagdelte standard arkitektur for systemet.



Funktionslaget indeholder alle operationer knyttet direkte til modellen, de fungerer som et interface til vores model og sikre et enkelt brugergrænsefladelag med en minimeret kobling til problemområdet.

For at bevare strukturen har vi valgt en Lukket-Streng arkitektur mellem lagene Brugergrænseflade-Funktioner, således at operationer i et lag kun kan bruge operationer i det underliggende lag. Derved holdes brugergrænsefladen skarpt adskilt fra modellen.

Vi har valgt at ligge et tekniks lag under modellen, som skal interface til MotionPlanneren. Derved er det ikke nødvendigt at ændre i model laget ved en udskiftning- af eller understøttelse af en anden MotionPlanner. Kun ændringer eller tilføjelser i det tekniske lag er nødvendigt.

Til eventuel understøttelse af en eller flere databasesprog, kan den tekniske platform udvides men interfaces til disse.

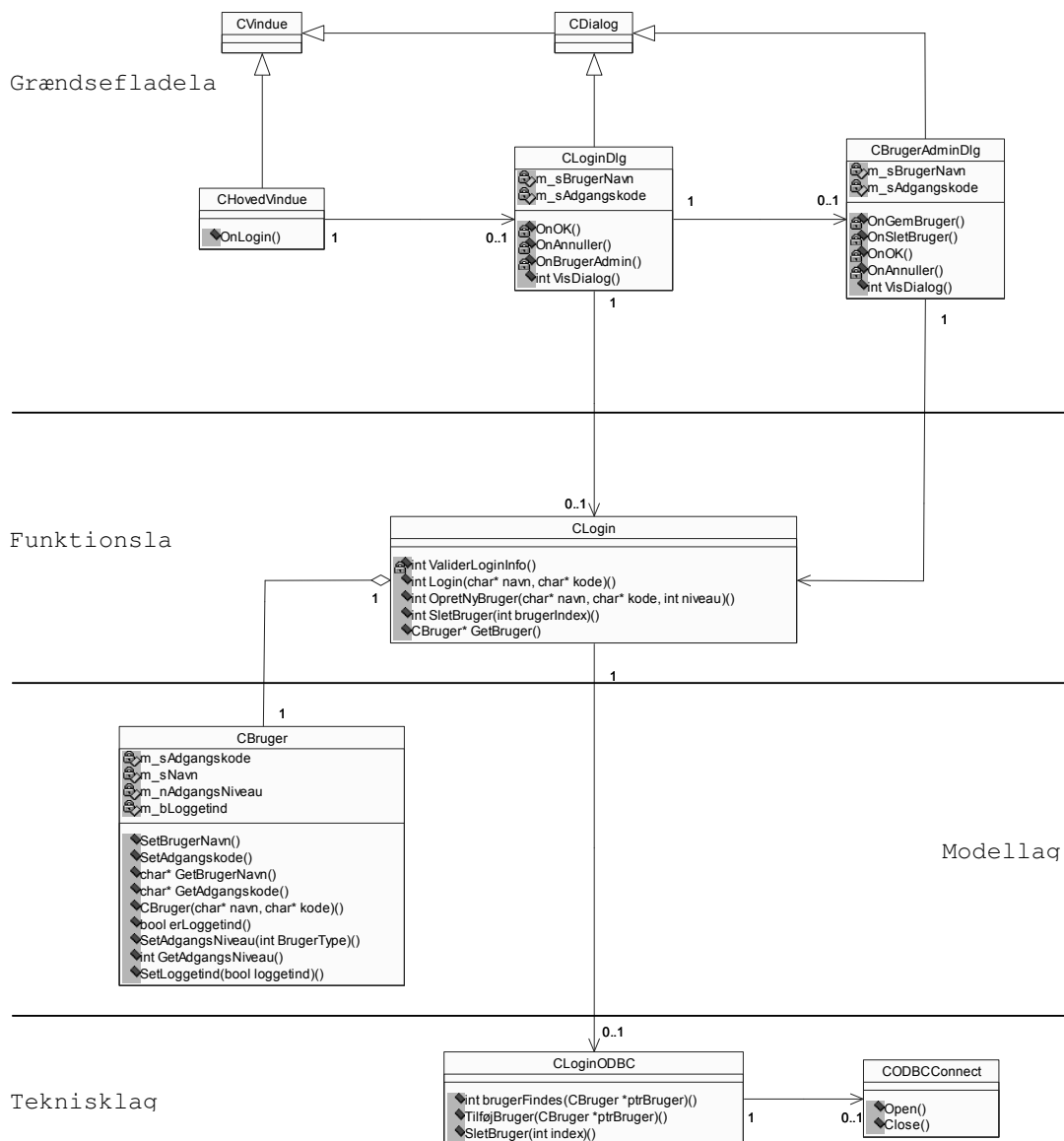
4.4.2 STANDARDS

UML standarden er benyttet sammen med OOA&D bogens forskrifter [MAT et al, 2001]. Dog er der enkelte undtagelser.

4.5 Komponenter

Idet følgende vil principper for design-struktur og klasser blive beskrevet. brugsmønstret Login og Tilføj bruger er udvalgt som eksempel og der er kun udarbejdet et endeligt design for disse brugsmønstre. Dvs. alle klasser med tilhørende funktioner og attributter for brugsmønstrene Login og Tilføj bruger designs her. De to brugsmønstre er udvalgt fordi de hænger tæt sammen. Resultatet tilføjes klassesdiagrammet fra analysedokumentet.

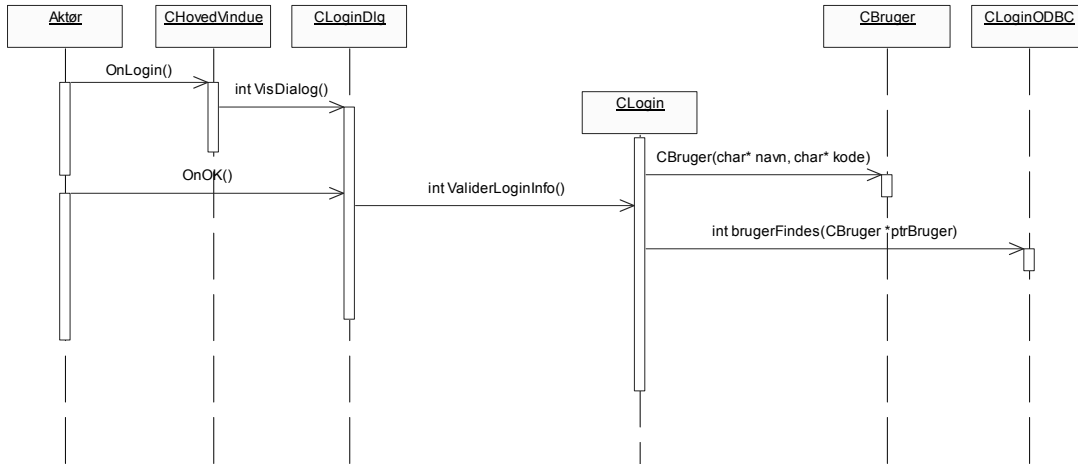
4.5.1 STRUKTUR



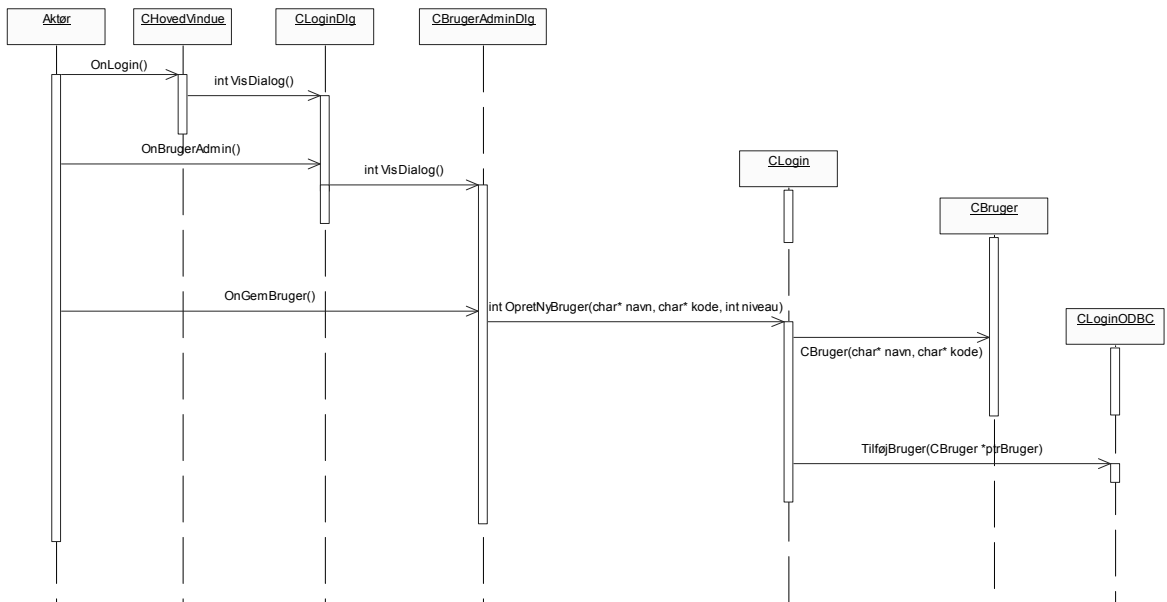
4.5.2 SEKVENSDIAGRAMMER

Der er her lavet sekvensdiagrammer for Login og Tilføj bruger.

4.5.2.1 Login



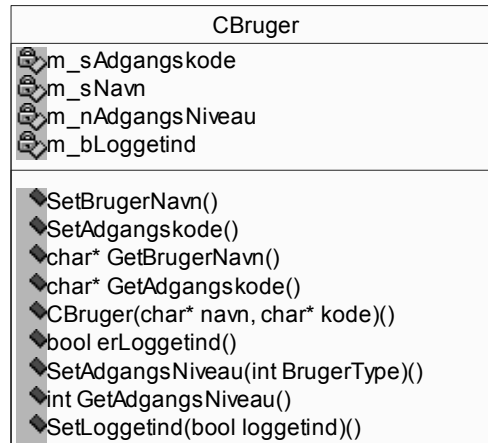
4.5.2.2 Tilføj bruger



4.5.3 KLASSER

I det følgende beskrives klasserne, som indgår i brugsmønsteret Login og Tilføj bruger.

4.5.3.1 CBruger



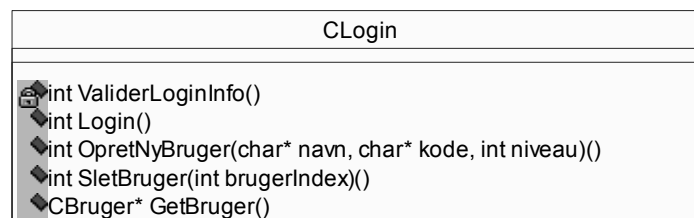
Klassen Cbruger svare til Bruger klassen fra analysemodellen og skal indeholde information om brugeren. Til det formål er klassen udvidet med operationer og attributter, som kan give klassen de ønskede egenskaber.

Attributterne holder på brugerens brugernavn og adgangskode, hvilken type bruger der er tale om, samt om brugeren er logget ind.

Det er valgt at gøre attributterne private, så de kun kan tilgås gennem de viste operationer.

Derudover er følgende nye klasser identificeret.

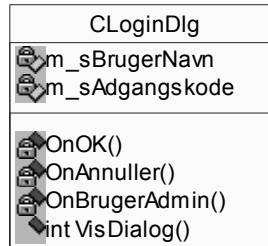
4.5.3.2 CLogin



Da vi vil tilstræbe en streng og lukket struktur, har vi lavet klassen CLogin til funktionslaget.

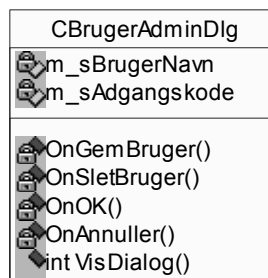
Klassen har til opgave at håndtere bruger login funktioner. Den kan validere, at en bruger forsøger at logge ind med korrekt brugernavn og adgangskode og returnere svar her på. Klassen anvendes også til at Oprette og slette brugere.

4.5.3.3 CLoginDlg



Denne klasse er en dialogboks, hvor brugeren kan indtaste brugernavn og adgangskode for at logge ind. Er brugeren logget ind som Configurator har han/hun også adgang til brugeradministration via denne klasse. Klassen har attributter til lagring af brugernavn og adgangskode, for at kunne indeholde brugerens indtastning.

4.5.3.4 CBrugerAdminDlg



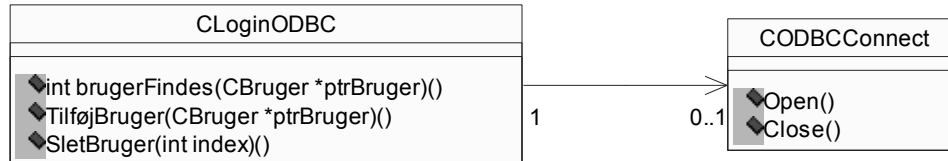
Dette er endnu en dialogboks, som kan tilgås af brugere der er logget ind som konfigurator. Klasse giver mulighed for at Tilføje og slette brugere.

4.5.3.5 CDialog og CVindue



Disse klasser står for den grafiske fremvisning af vinduer og dialogbokse.

4.5.3.6 CLoginODBC og COBCCConnect



Disse klasser er et interface til en database som skal indeholde alle brugeroplysninger. Med disse klasser kan brugere findes, tilføjes og slettes.

4.5.4 NYT BRUGSMØNSTER IDENTIFICERET: SLET BRUGER

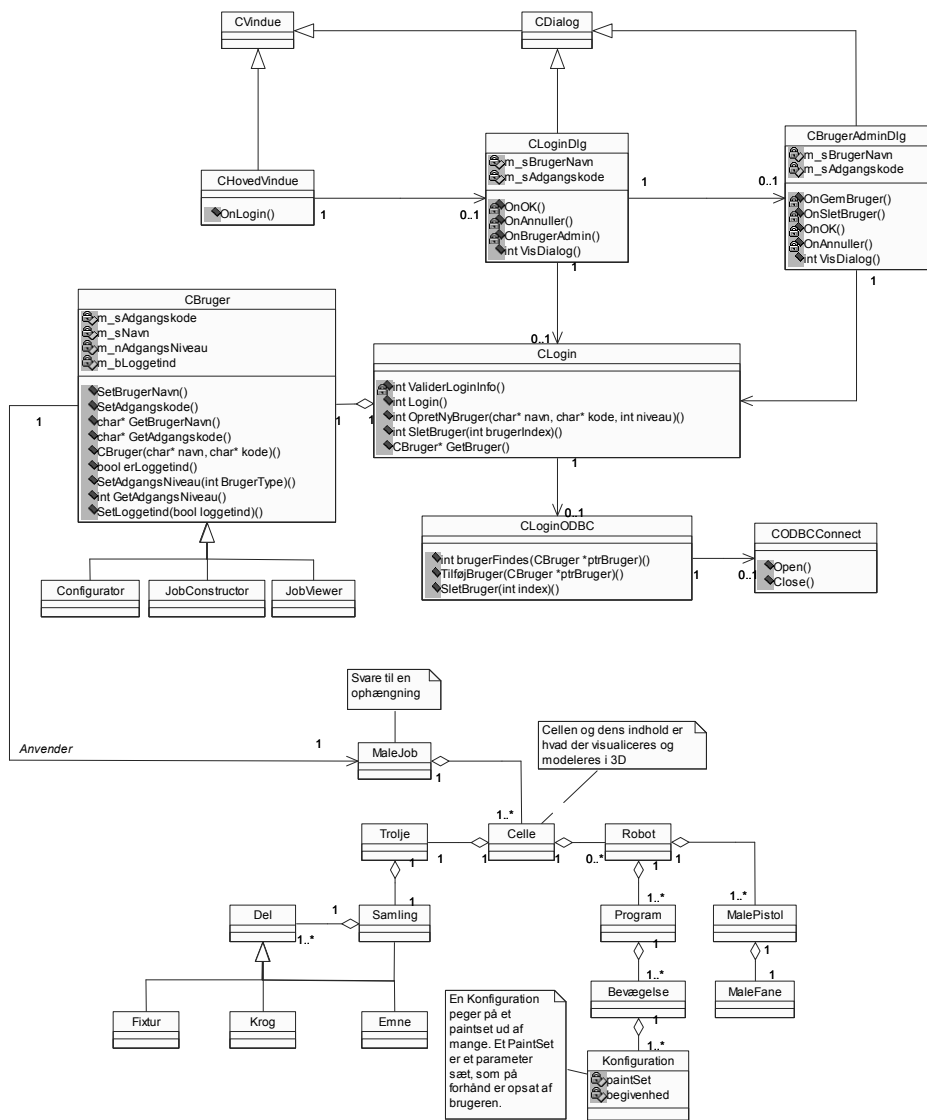
Der vil også være brug for at kunne slette brugere, som ikke længere skal have adgang til systemet. Et brugsmønster for sletning af bruger er tæt relateret til tilføj bruger. Derfor er klasserne også udstyret med operationer hertil.

Nye operationer:

```
CbrugerAdminDlg::SletBruger();
CLogin:: int SletBruger(int brugerIndex);
CLoginODBC::SletBruger(int index);
```

4.5.5 UDVIDET KLASSEDIAGRAM

Her ses klassediagrammet fra analysen, udvidet med funktionalitet for Login, Tilføj- og Slet-bruger.



5 INDFØRING AF METODEELEMENTER I NYKREDIT HHV. INROPA A/S

I dette afsnit forsøger vi at besvare den del af problemformuleringen som går på, at afklare om udvalgte elementer fra OOAD-metoden nævnt i [MAT et al, 2001] men fordel kan indføres i Nykredit hhv. Inropa.

Hvordan er elementerne udvalgt?

- 1) Vores fokus/oplevelse ved at bruge og arbejde med metoden som er beskrevet i nærværende rapport.
- 2) Vores erfaring fra arbejdet med metode/udvikling i Nykredit hhv. Inropa, hvor vi muligvis ser os i stand til vide hvad der er behov for.
- 3) Der skal foretages vurdering af både analyse og design elementer

I Nykredit indeholder udviklingsprojekterne i stigende grad implementering af Web-baserede brugergrænseflade, der bygges op med funktions og metode lag på hosten. Den objektorienterede JAVA udvikling skal således bygges sammen med de øvrige komponenter og her vil det være oplagt, at se på om Nykredit samlede udviklingsmodel skal justeres både i design processen, men også i analyse- og kravspecificerings faserne i projektførelsen. Med den indsats og strategisk betydning som Nykredit tillægger udviklingsprojekterne, vil vi hævde at projekternes succes (bl.a.) afhænger af hvad udviklingsmetode har at tilbyde under hele forløbet. Der er således behov for objektorienterede tiltag under hele projektførelsen, fra forretningsanalyse til design og implementering.

I Inropa med dets historie og nuværende produktionskapacitet, er der behov for metode i rette dosering som virksomheden har mulighed for at iværksætte. Endvidere må ledelsen udstikker kravene og retningslinier for hvad der skal bruges ressourcer på.

Vi har udvalgt emner som vi vil forsøge at beskrive fordele og ulemper ved ift. indførelse og brug i de 2 virksomheder. Områderne er udvalgt så de dækker emner under analyse og design. Endvidere er et udvalgskriterie, at vi selv ser området som et godt redskab i metoden og at netop det område gør en forskel. Vi må erkende, at bedømmelseskriteriet er lille, dvs. erfaringen med metoden er begrænset, men vi støtter os til egen erfaring ved brug af andre udviklingsmetoder.

Områder som vi valgt at fokusere på er 1)BATOFF-kriteriet, 2)Hændelsestabel , 3)Klassediagram, og 4)Arkitekturmønstre.

Ved indførelse af eksempelvis ny teknologi, nye forretningsgange eller rutiner i organisationer, kan der opstå en vis form for modstand mod det nye der er på vej. Denne modstand kan variere fra nedlæggelse af arbejde til stiltiende accept med nedsat produktivitet og motivation for den enkelte. Se evt.[Lippitt et al, 1996] og [Fry et al, 1989].

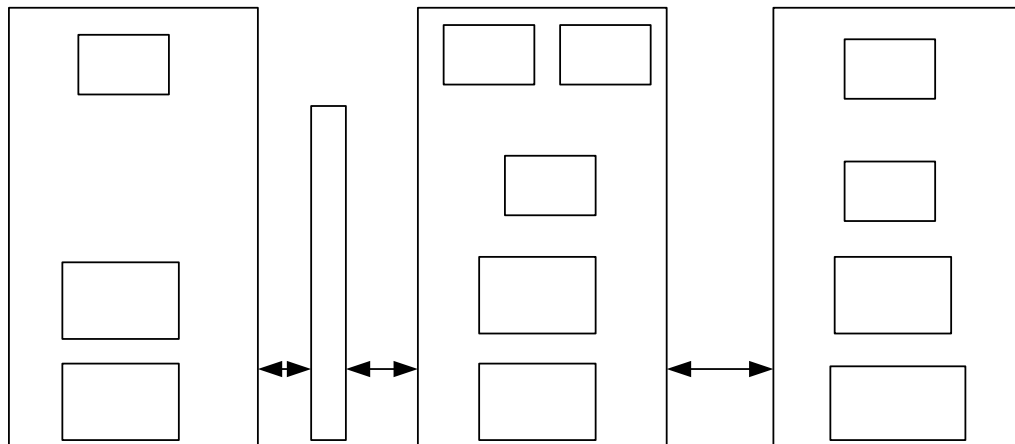
I den følgende præsentation vil disse aspekter ikke bliver behandlet, idet vi ser emnet som udenfor denne rapport problemområde. Endvidere er emnet vidtfavnende og så vigtigt, at det kræver omfattende og intensiv behandling som ikke kan honoreres her.

For at give læseren en bedre baggrund til at forstå de enkelte begrundelser, beskrives de 2 virksomheder kort i det følgende.

5.1 Virksomhedsbeskrivelse

Nykredit koncernudvikling har siden 1980 udviklet systemer til Realkreditvirksomheden Nykredit. I starten af 1990'erne blev de 3270 baserede systemer suppleret med Client/server systemer. Nykredit valgte i 1998 at satse på selvudviklingsstrategien, hvor forretningsområderne Bank og Forsikring på den baggrund kom til. Med teknologien og behovet startede udvikling af Web-baserede systemer midt i 1990'erne, så som Bank og Portal løsninger til enkelte forretningsområder. Gennem alle årene er data gemt centralt i DB2 databaser og udvikling af tynde interfaces er stadig et metodekrav. Endvidere er der krav om at forretningslogik placeres i host moduler. Udviklingsmetoden er løbende blevet ajourført til det aktuelle behov og hvad der på længere sigt er mest hensigtsmæssig for at koncernudvikling kan servicere dets kunder. De små projekter/forvaltning har mulighed for, at benytte den del af det samlede metodeapparat, som de har behov for. Metodeafdelingen har ansvaret for implementering af metoder og servicering heraf. Udvikling af systemer er primært foretaget via CBD, hvor entiteter og relationer er nøgleord. Forretningsanalysen foretages med stor inddragelse af kunden og via metoden UCCBD som der pt. benyttes stor energi på at indføre i alle koncernens udviklingsprojekter.

Figur 5.1 nedenfor viser overordnet Nykredits driftsmodel fra modellaget til Brugergænsefladen.



Figur 5.1 Nykredits driftsmodel

Inropa A/S

Opstart i 1995 og har nu 3 ansatte, med direktør der varetager strategi, markedsføring og kundefremskaffelsen. 2 udviklere der benytter Visual Basic (VB) til udvikling og dokumentation. Primært nyudvikling og i begrænset omfang vedligeholdelse af systempotefølgen (procentvis 80/20). I analysenfasen benyttes klassediagrammering til den grundlæggende specificering med kunden, men i øvrigt benyttes der ingen metode til at understøtte udviklingen og dokumentering.

Tendensen i industriel produktion går i retning mod at tilpasse produkterne til kundebehovene samt at minimere lagerstørrelserne. Dette betyder, at der er behov for en stadigt stigende fleksibilitet i virksomhedernes produktionsudstyr. Det er velkendt, at industrirobotten er en af de mest fleksible maskiner, der findes i industrien, forstået således, at de kan omstilles til meget forskellige opgaver og bevægelsesmønstre. Dog er det et stigende problem, at omstillinger til nye bevægelsesmønstre er omkostningstunge, fordi der ikke foregår nogen produktion i den tid, hvor omstillingen foregår.

Inropa udvikler software, der giver industrielle malerobotter en form for håndværksmæssig intelligens, således at disse får lettere ved at overtage det usunde arbejde i industriens malelinier. Vores software gør robotten i stand til at omstille sig ved egen hjælp, så den uden besvær kan male forskellige emner uden at skulle omprogrammeres.

Det er muligt at anvende Inropa's løsninger til en række andre overfladeprocesser i industrien.

Inropa er en kommerciel knopskydning af et flerårigt EU projekt, FlexPaint, med deltagelse af bl.a. Aalborg Universitet.

5.2 BATOFF-kriterier

5.2.1 NYKREDIT

Vi ser at BATOFF-kriterier med fordel kan supplere den eksisterende Nykredit dokumentation på flere områder. Der kan opnås en god oversigt over projektets væsentlige elementer, som kunden og den projektansvarlige nødvendigvis må være enige om. Der kan i højere grad opnås et fælles grundlag at starte systemudviklingen på. Det er muligt, at en del af informationen allerede er til stede i de nuværende Nykredit dokumenter, men at samle grundkernen et sted, er en god ide. Nykredit har flere parametre der skal beskrives i opstartsfasen, når kunde vil specificere den påtænkte IT-udviklingsopgave, men et element som filosofi ser vi som et godt nyt element til supplement af dette.

Vi ser ingen ulemper ved indførelse af denne aktivitet og vi vil foreslå at elementerne indarbejdes i den nuværende metode og ikke blot bliver et appendix til samme. Vores erfaring siger at tilvalgte elementer i høj bliver fravalgt (metoden er lavet til situationsbestemte projekter), så derfor skal de primære fokuselementer trækkes frem og synliggøres.

5.2.2 INROPA

Inropa kan med fordel indføre denne aktivitet, idet det vil give en fast struktur og bedre grundlag ved review. Det er altid en god ide at fastholde de overordnede krav overfor kunden og har virksomheden en fast procedure for dette, vil det hjælpe Inropa og kunden med klart at definere aftalegrundlaget. Dermed kan omkostningskrævende tilbageløb og tvister med kunden muligvis undgås.

Vi kan ikke se at der er åbenlyse ulemper for Inropa ved, at indføre BATOFF-kriterierne.

5.3 Hændelsestabel

Hændelsestabel beskriver sammenhængen mellem klasser og hændelser og den giver identifikation af om enkelte klasser har ansvar for mange hændelse og enkelte hændelser sammenknytter mange

koblede klasser. Hændelsestabelen vil senere i processen blive opdateret med antal gange den enkelte hændelse forekommer pr. klasse (0 eller 1, 0 eller flere gange)

5.3.1 NYKREDIT

Fordelen ved at bruge hændelsestabel aktiv i Nykredit under analysearbejdet er relateret til det grundlæggende i en objektorienteret metode, dvs. det at tænke klasser/objekter og hændelser der sker på disse. Vi ser flere fordele ved at tænke på denne måde, primært ved den stigende udvikling af web-baserede systemer hvor JAVA udviklere tænker og koder i disse termer. Der vil ikke være behov for at ”oversætte” fra den nuværende relationelle/entitets tankegang, hvis fokus flyttes til at beskrive analysen på denne måde. Der vil i mange år fremover i Nykredit være udviklere som har baggrund i relationer/entitets metoden og det er der også behov for til eksempelvis vedligeholdelse af eksisterende legacy systemer, men vi ser ingen problemer med at uddanne flere i objektorienteret metode og lade de 2 metoder leve side om side, hvor vi sandsynligvis vil se, at objektorienterede metoder vil vinde frem. Den eneste ulempe vi kan forudse er, at der kan opstå en periode hvor sproget skal forenes mellem udviklingsgrupperne. Dette kan evt give nogle frustrationer og ses i dårlig kvalitet af dokumenter og programmer.

5.3.2 INROPA

Inropa kan med fordel gøre brug af aktiviteten hændelsestabel, idet aktiviteten er grundlæggende og samtidig sætte struktur på de elementer som skal diskuteres med Kunden. Inropa har i nogen udstrækning benyttet denne aktivitet, men mere på den uformelle måde. Vi ser det som en klar fordel at Inropa på denne struktureret måde beskriver sammenhængene mellem klasser og hændelser.

5.4 Klassediagrammer

Klassediagrammet som er et resultat af strukturaktiviteten [MAT et al, 2001], giver et samlet billede af sammenhængen mellem klasser og objekter i modellen af problemområdet.

5.4.1 NYKREDIT

Der vil kunne opnås synergi effekt til de efterfølgende faser, når modellering foretages med udgangspunkt i samme metode, dvs. til JAVA klasserne der implementeres og instantiering/oprettelse af objekterne. Nykredit benytter udviklingsværktøjet ALL fusion Gen [www3.ca.com] til udvikling af de primære Realkredit applikationer, dog ikke til udvikling af web præsentationsinterface og det indeholder hjælp til oprettelse af diagrammer der viser cardinaliteten og relationerne. Den forståelsesmæssige side ser vi ikke som et problem, da opgaven er en kendt kompetence.

Der kan opstå tvetydige tolkninger og resultatet kan i yderste konsekvens give tilbageløb i udviklingsprocessen, hvis den nuværende metode, der i stor udstrækning vedrørende entiteter og relationer, så at sige blandes med klasser/objekter og strukturer.

Det er vigtigt at gøre sig klart at de 2 metoder adskiller sig på væsentlige områder og dermed er det tilrådelig at holde disse 2 metoder adskilte.

Hvordan skal værktøjsunderstøttelse finde sted fremover? Kan ALL Fusion Gen bruges sideløbende med objekt værktøj eller skal nye projekter fra dag 1. starte med at definere problemområdet inklusiv klasser/objekter/hændelser i nyt objekt værktøj? Vores bud er, at der findes mulighed for at forene disse elementer i ALL Fusion Gen.

5.4.2 INROPA

Inropa vil med fordel kunne bruge klassediagrammer i udviklingsprocessen, så der kommer mere standardisering til brug her og nu, men også på længere sigt, hvor dokumentationen vil øge fremdriften på nye udviklingsaktiviteter, men også til fejlsøgning og øvrige vedligeholdelsesopgaver. Teknikken er hurtig at tilegne sig og de fleste udviklingsværktøjer indeholder faciliteter til hjælp under bearbejdelsen. Som tidligere nævnt benytter Inropa allerede denne diagrammeringsform på den grundlæggende specificering med kunden og det er således et spørgsmål om at udvide dette så også detaljerne kommer med. Dette er også en hjælp i designfasen, når analysen er foretaget på en højere detaljeringsgrundlag.

Der er ikke umiddelbart nogen ulemper forbundet for Inropa ved at indføre og bruge klassediagrammer på denne mere koncentreret måde .

5.5 Brug af arkitekturmønstre

Mønster begrebet er en generel beskrivelse af et problem og en tilhørende løsning. Der tales om arkitektur-, design-, OO-design- og brugergrænseflademønstre mfl.. Det er en god måde at styrke kvaliteten og effektiviteten i analyse og design [MAT et al, 2001]. I analysen kan rollemønsteret eksempelvis bruges til at modellere de forskellige roller, som personer kan indtage over tid. I design er der arkitektur mønstre som omhandler 4 mønstre: lagdelt, komponenter og delsystemer, grundarkitekturen og endelig klient-server arkitekturen. Ved valg af arkitektur er der flere hensyn at tage og de skal afvejes. Der er krav og hensyn fra kunden (BATOFF), tekniske hensyn såsom tilgængelig, hardware og sikkerhed, men grundlæggende skal arkitekturen afprøves, evalueres og beskrives.

5.5.1 NYKREDIT

Nykredit har igennem mange år brugt ressourcer på at udarbejde den nødvendige og mest hensigtsmæssige driftsmodel (se figur 5.1). Denne driftsmodel skal i det daglige understøtte forretningens behov, men der skal også på længere sigt være mulighed for tiltag som ikke kræver en større omlægning af drift og systemer. Fra 1980'ernes mainframe (3270-emulering) systemer til klient-server i midt 90'erne og nu web-baserede løsninger, kan Nykredit siges at have arkitekturen med i alle overvejelser, når nye tiltag skulle iværksættes. Det er en svær omstillingproces, når der indføres ny arkitektur i Nykredit, idet alle spørger metodeafdeling om vejledning til hvordan metoden skal bruges i det daglige arbejde.

Nykredit ulempe:

For enkelte udviklere vil dette muligvis blive set som en spændetrøje, hvor den kreative proces ikke til fulde kan udnyttes. Vi ser dette som et element der ikke skal bagataliseres og derfor bør katalogets indhold nøje afpasses dertil. Vi ser en ulempe ved at der benyttes mønstre som ikke passer til den specielle situation og der opstår en sub optimering, hvor nogle af design kvalitetskravene (FURPS: Functional, usability..) optimeres, medens andre krav som kunden specifikt har prioriteret højest ikke opfyldes.

5.5.2 INROPA

For Inropa vil det være en fordel at tænke over og nedfælde disse arkitekturovervejelser, for derved at imødegå de fremtidige krav til produktudvikling. Det vil især være et krav, hvis Inropa påtænker at udvide produktpaletten og dermed skal servicere flere forskellige kundetyper. Her tænkes der på

om Inropa vil udvikle total-systemer eller plug-on-play værktøjer til de førende produkter på markedet.

For Inropa kan vi kun se at ressourceindsatsen til specificering og vurdering af arkitekturen skal tages fra de daglige arbejdsopgaver og det vil i første omgang ramme kunden og udvikling af nye produkter.

5.6 Opsummering af fordele/ulemper

Nykredit	Fordele/ulemper
BATOFF	Fordel: Giver overblik og skaber fælles grundlag mellem kunde og projektets øvrige interessenter. Ulempe: hvor skal de enkelte interessenters behov angives? BATOFF er således kun et supplement til den eksisterende dokumentation
Hændelsestabel	Fordel: Klasser/hændelse begreber er i fokus i alle projekters faser. Målrettet mod objektorienterede sprog som JAVA. Ulempe: ny metode/begreber skal indarbejdes og forenes mellem udviklingsværktøjer.
Klassediagram	Fordel: Der opnås synergi til den efterfølgende design/konstruktionsfase når der benyttes samme metode. Ulempe: Ved overgang til ny metode og sprogbrug kan der opstå uklarheder og misforståelser. Der kan være problemer med værktøjsunderstøttelse (All Fusion Gen) af denne metode.
Arkitekturmønstre	Fordel:Fælles standard og struktur. Ulempe: kreative/nye ideer opdyrkes ikke
Inropa A/S	
BATOFF	Fordel: Et godt redskab til at fastholde krav/overblik overfor kunden og kan bruges i review sammenhæng. Ulempe: Ingen.
Hændelsestabel	Fordel: Grundlæggende specificeringsmetode som både intern og ekstern mod kunde giver informationer der kan bearbejdes og foretages review på. Ulempe: Ingen.
Klassediagram	Fordel: Flere detaljer kommer frem til brug for analyse, men også i den efterfølgende designproces. Ulempe: Ingen.
Arkitekturmønstre	Fordel: Det giver en konkurrencefordel at have tænkt over disse elementer. Ulempe: Det tager tid fra øvrige opgaver i virksomheden og er det virksomhedens strategi?

6 GENERELLE KVALITATIVE OPLEVELSER

Som nævnt i problemformuleringen vil vi her forsøge at opremse vores erfaringer, oparbejdet lærdom og oplevelser under projektførelsen. De nævnte områder nedenfor er udvalgt fordi de har fyldt meget under udarbejdelsen af nærværende rapport og vi synes at områderne er vigtige at fastholde til egen og andres fremtidige brug.

Problem- og Anvendelsesområde: hvad er denne opdeling til gavn for?... Vi fokuserede hurtigt på anvendelsesområdet og forsøgte at diskutere emner relateret til det kommende systems funktionelle operationer, idet vores fokus var rettet mod systemet der skulle udvikles. Vi ser nu en fordel i, at starte med analyse af problemområdet i de situationer, hvor der er tale om nyudvikling og der findes ingen eller kun lidt viden om dette område. Dermed vil der i højere grad fremkomme forretningsmæssige termer som kan komme den efterfølgende analyse til gavn, dvs. systemet bliver udviklet til gavn for forretningen og ikke omvendt.

Review: Der er ikke afholdt formelle review [Pressman,2005] af afsnit eller ved faseskift som litteraturen foreskriver. Det er ikke et bevidst valg men mere en kombination af 2 forhold. For det første er der løbende afholdt vejledermøder, hvor vi har antaget, at de i nogen udstrækning har fungeret som review. Det andet forhold er et tidsmæssig aspekt hvor vi ikke har kunnet afsætte ressourcer til det. Vi vil her gerne videregive, at de uformelle reviews bør foretages på delaktiviteter, men ved faseskift, f.eks. ved overgang fra analyse til designfasen, bør der afholdes et formelt review, så producenten har et klart billede af kvaliteten af materialet og derefter kan justere indhold/kvalitet.

Tilstandsdiagrammer: De var svære at komme i gang med og noget af det skyldes manglende viden om teknikken, men også manglende forståelse for brugen af diagrammerne i den kontekst som de udvikles til. Vi kan se en fordel i brugen af tilstandsdiagrammerne ved kommunikation med brugeren hvor diskussion om arbejdsprocessens korrekte forløb skal afklares, men i højere grad når programkode skal udvikles, idet der på det tidspunkt er behov for at kende objekternes tilstands og livscyklus.

Brugerinvolvering: Vi har fravalgt at lade en kunde/interessent gennemlæse dele af analyse/design dokumentet og dermed give feedback på det indholdsmæssige og evt. komme med ideer til alternative løsninger. Dette var et bevidst valg fra dag 1 og begrundelse har været, at viden om problemområdet var til stede i gruppen. Denne antagelse holder stadig til en vis grad, dog kan vi se en mulighed for at få suppleret BATOFF, funktionslisten og ikke mindst brugergrænsefladen med input udefra. Det er svært, at afgøre hvad vi har mistet ved dette valg og vi vil hævde, at der skal vægtige argumenter til at fravælge denne aktivitet.

Processen: Vi oplevede en iterativ proces med megen tilbageløb, som skyldes (vores) manglende viden, men også ”sådan er det i virkeligheden”. Det at gøre noget helt færdigt førend næste aktivitet kan starte, er ikke altid muligt og hensigtsmæssig. Vi forsøgte at udarbejde dokumenter til de enkelte faser i takt med kurset fremdrift, for derved at have teori frisk i erindring, men da viden fra foregående fase ikke var (helt) på plads, blev der bygget ovenpå et i forvejen usikkert fundament. I den sidste hektiske 3.del af rapportens kalender tid, blev der rettet i de primære analyse dokumenter såsom hændelsestabel og tilstandsdiagrammer. Dette er muligt her med denne opgaves kompleksitet og så stadig bevare overblikket, men i større projekter vil dette tilbageløb give megen frustration og ikke virke professionelt overfor kunden.

Generelt om brugen af metoden: Vi ser metoden [MAT et al, 2001] som forholdsvis let at gå til. Især analyse-delen giver en god struktur til, at få foretaget et analyse arbejde, der hvis de nævnte anbefalingerne følges, giver en god kvalitet af resultatet. Design-delen har været sværere at gå til, og her savner vi programmeringsunderstøttelse, som vi har valgt fra. Dette kunne givetvis have tilføjet de elementer som ville binde analyse og design sammen og give en godt udgangspunkt for den efterfølgende konstruktionsopgave.

Erfaring: Det helt naturlige spørgsmål der nu må stilles er - hvad vil vi gøre anderledes næste gang? og her tænkes der både i (EVU) projektsammenhæng og i det daglige arbejde:

- Føre dagbog så detaljer huskes og kan bearbejdes til mere kvalitet fremadrettet
- Forsøge at estimere opgavernes omfang mere realistisk, så fundamentet bliver mere stabilt og blive enige om hvornår næste fase kan påbegyndes.
- Afsætte tid til review i projektplanen og sikre at de afholdes
- Inddrage brugerrepræsentant til kvalitetssikring og forsøg at få alternative løsninger derfra.

7 KONKLUSION OG PERSPEKTIVERING

7.1 Konklusion

Vi har gennemført og dokumenteret analyse og design aktiviteter på udvikling af malerobot program. Analyse og design er foretaget stort set som beskrevet i [MAT et al, 2001].

For at afprøve enkelte elementer fra metode, er der udvalgt 4 elementer som tænkes indført i Nykredit hhv. Inropa. Der er specificeret fordele og ulemper ved de nævnte metodeelementer og stort set alle elementer vil give fordele for de nævnte virksomheder. Nykredit udviklingsafdeling har en vis størrelse og kapacitet der gør, at behovet for metode understøttelse ikke er så markant som Inropa.

Nykredit kan især gøre brug af den objektorienterede tankegang, hvor disse elementer er i fokus og lade denne metode tankegang blive aktiv i hele udviklingsprocessen fra projektstart til implementering.

Inropa kan have behov for mere metodeunderstøttelse og de nævnte tiltag i metoden [MAT et al, 2001] vil være en god start. Der er mulighed for, at Inropa gradvis kan bruge de dele af metoden som passer til den aktuelle arbejdssituation. Dog må analysegrundlaget være på plads, så analyse og design udvikles under de samme metodeforudsætninger.

Vi har valgt ikke at udvikle systemer efter design dokumentet, så vi har ikke erfaring med i hvor grad metoden kan supplere i denne fase.

De kvalitative oplevelser ved brug af metode er hovedsagelig positive, idet metoden indeholder en god struktur og vejledning til fremdrift. Vi har dog måttet opleve en del ændringer til analyse og designet, som i væsentlig grad skyldes egne forhold, f.eks. (under) estimering af hvor lang tid de enkelte opgaver tager.

7.2 Perspektivering

Vi vil følge anbefalinger fra kapitel 5, hvor tiltagene stort set kan iværksættes her og nu.

Vi tror på, at metodeelementerne skal ud og bruges i Nykredit og Inropa. I Nykredit skal det ske via møder med metodeafdeling og intern i udviklingsafdeling. I Inropa vil brug af metoder i første omgang skulle aftales med Direktøren, men vi tror på, at fordelene er så gode, at der hurtig er accept til det.

Det har aldrig (for os) været et spørgsmål om OOAD-metoden [MAT et al, 2001] kan bruges eller ej, vi ser det som en opskrift, hvor enkelte eller flere elementer kan tages ud og bruges i det daglige arbejde. Der er således tale om at metoden kan og skal bruges til det aktuelle behov og situation.

LITTERATURLISTE

[MAT et al, 2001] Objekt Orienteret Analyse & Design, 3.udgave 2001,
Lars Mathiassen, Andreas Munk-Madsen, Peter Axel Nielsen, Jan Stage ISBN 87-7751-153-0

[AAUProjekt, 2000] Tema: Crisplant Århus, projektrapport i virksomhedsanalyse udarbejdet på
VT8 AAU af Jesper A. Andersen, Lars L. Lilleholt m.fl.

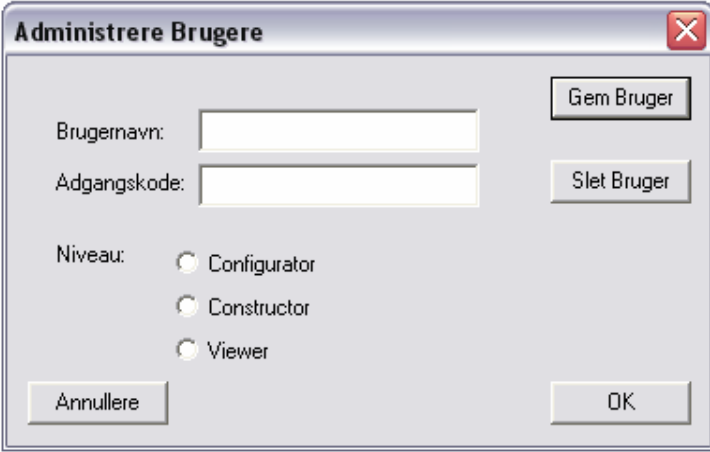
[Larman, 2000] Applying UML and Patterns, Prentice Hall
Craig Larman

[Lippitt et al, 1989] Implementing organizational change
Gordon L. Lippitt, Petter Langseth, Jack Mossop ISBN 0-87589-622-7

[Fry et al, 1989] Strategic analysis and action 2.nd Edition
Joseph N. Fry, J. Peter Killing ISBN 0-13-850496-2

[www3.ca.com] CA's hjemmeside med produktbeskrivelser af Programmeringsværktøjer.

APPENDIX



Administrere Brugere

Brugernavn:

Adgangskode:

Niveau: Configurator
 Constructor
 Viewer

Buttons: Gem Bruger, Slet Bruger, Annullere, OK

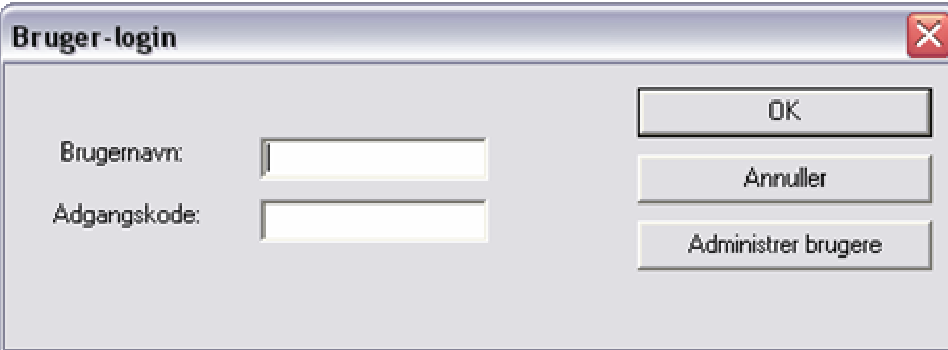


Nyt MaleJob

Navn:

Placering:

Buttons: OK, Gennemsg..., Annullere



Bruger-login

Brugernavn:

Adgangskode:

Buttons: OK, Annuller, Administrer brugere

