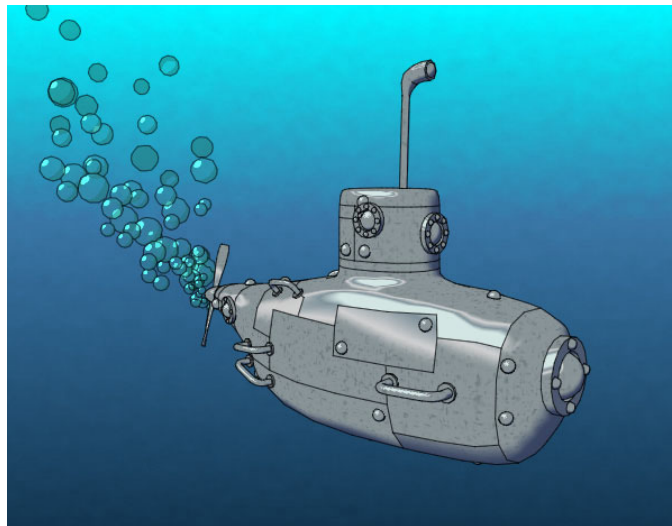


John Olausson

Aspekter på användarvänlighet hos ett datorprogram, avsett som forskningsprototyp

med tillämpning på SubTrack, en simulator för ubåtsföljning



TOTALFÖRSVARETS FORSKNING SINSTITUT

Ledningssystem

Box 1165

581 11 Linköping

FOI-R--0512--SE

Juni 2002

ISSN 1650-1942

Metodrapport

John Olausson

Aspekter på användarvänlighet hos ett datorprogram, avsett som forskningsprototyp

med tillämpning på SubTrack, en simulator för ubåtsföljning

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	Rapportnummer, ISRN FOI-R--0512--SE	Klassificering Metodrapport	
	Forskningsområde 4. Spaning och ledning		
	Månad, år Juni 2002	Projektnummer E7031	
	Verksamhetsgren 1. Forskning för regeringens behov		
	Delområde 41 Ledning med samband och telekom och IT-system		
	Författare/redaktör John Olausson	Projektledare Per Svensson	
Godkänd av			
Uppdragsgivare/kundbeteckning FM			
Tekniskt och/eller vetenskapligt ansvarig Per Svensson			
Rapportens titel Aspekter på användarvänlighet hos ett datorprogram, avsett som forskningsprototyp - med tillämpning på SubTrack, en simulator för ubåtsföljning			
Sammanfattning Rapporten kommer att behandla användarvänligheten hos en interaktiv, datorbaserad forskningsprototyp, tillämpad på ubåtsföljningsprogrammet SubTrack. Arbetet resulterar i ett designförslag för en användarvänlig version av SubTrack, anpassad för en användarroll: demonstratörens. En förbättrad gränssnittsdesign föreslås för demonstratören genom omstrukturering av gränssnittets komponenter. Endast få tillägg till programmets funktionalitet föreslås. Den tid som demonstratören har att sätta sig in i programmet är begränsad och hon måste klara sig utan hjälp, då demonstrationen ofta sker utanför den normala arbetsplatsen. Användarvänligheten behöver utvecklas så att användaren enkelt kan utföra sina uppgifter, med ett minimum av handgrepp. Rapporten innehåller en beskrivning av det befintliga SubTrack och den miljö där programmet har vuxit fram. En kartläggning har gjorts av de brister som finns för en demonstratörs interaktion, och ett förslag till förändringar för en utvecklad version av SubTrack ges. Rapporten avses främst att användas som ett designförslag för SubTracks användargränssnitt, vid en eventuell vidareutveckling. Rapporten skall dock även kunna användas som ett exempel på användarvänlig design vid utveckling av liknande prototyper, främst inom FOIs avdelning för Ledningssystem.			
Nyckelord Användarvänlighet, användbarhet, människa-datorinteraktion, MDI, gränssnitt, prototyp			
Övriga bibliografiska uppgifter	Språk Svenska		
ISSN 1650-1942	Antal sidor: 90 s.		
Distribution enligt missiv	Pris: Enligt prislista		

Issuing organization FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	Report number, ISRN FOI-R--0512--SE	Report type Methodology report
	Research area code 4. C4ISR	
	Month year June 2002	Project no. E7031
	Customers code 1. Policy Support to the Government	
	Sub area code 41 C4I	
Author/s (editor/s) John Olausson	Project manager Per Svensson	
	Approved by	
	Sponsoring agency The Swedish Army	
	Scientifically and technically responsible Per Svensson	
Report title (In translation) Aspects of usability of a computer program, intended as a research prototype - applied to SubTrack, a submarine tracking simulator		
Abstract <p>This report will discuss usability factors concerning an interactive computer-based prototype, applied to the submarine-tracking program SubTrack. The report results in a design proposition for a more user friendly SubTrack, suited for one actor, the demonstrator.</p> <p>A design proposition is made for SubTrack's new interface design. Proposed changes are mainly a reorganization of the interface components. Only a few additional functions are suggested as an addition to the program. The time for the demonstrator to learn the program is limited and she often has to manage on her own, though the demonstration often takes place outside the ordinary premises. The usability needs to be improved so that the user is able to solve her tasks easily, with a minimum of actions.</p> <p>The report contains a description of the present SubTrack and of the environment where SubTrack has been developed. An identification of the unfulfilled requirements of the demonstrator, and a design proposal for a developed version of SubTrack is given. The use of the report is mainly as a design proposal for SubTracks user-interface, in case of a development. This report may also be used as a support when developing interfaces for similar prototypes, mainly within FOI:s division Command and Control Systems.</p>		
Keywords Usability, Human-Computer Interaction, HCI, Interface, Prototype		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 90 p.	
	Price acc. to pricelist	

Innehållsförteckning

<u>1</u>	<u>Introduktion</u>	9
<u>1.1</u>	<u>Roll</u>	9
<u>1.2</u>	<u>Mål och syfte</u>	9
<u>1.3</u>	<u>Läsanvisning för rapport</u>	10
<u>1.4</u>	<u>Terminologi</u>	11
<u>2</u>	<u>Beskrivning av befintligt Sub Track</u>	12
<u>2.1.1</u>	<u>Syfte och mål med SubTrack</u>	12
<u>2.1.2</u>	<u>Roller och aktörer</u>	13
<u>2.1.3</u>	<u>Användningsfall, UML-notation</u>	14
<u>2.2</u>	<u>SubTracks funktionalitet</u>	15
<u>2.2.1</u>	<u>Simulering</u>	16
<u>2.2.2</u>	<u>Beräkningsfunktion</u>	16
<u>2.2.3</u>	<u>Beräkningspresentation</u>	16
<u>2.2.4</u>	<u>Programparametrar för initiering</u>	18
<u>2.2.5</u>	<u>Tillstånd</u>	18
<u>2.3</u>	<u>Användarkontroll</u>	19
<u>2.3.1</u>	<u>Menyer</u>	19
<u>2.3.2</u>	<u>Dialogfönster</u>	20
<u>2.3.3</u>	<u>Körkontroller</u>	20
<u>2.3.4</u>	<u>Zoom och panoreringsfunktioner</u>	21
<u>2.3.5</u>	<u>Direktmanipulation</u>	21
<u>2.4</u>	<u>Presentation</u>	21
<u>2.4.1</u>	<u>Huvudfönstret</u>	21
<u>2.4.2</u>	<u>Ritfunktion</u>	21
<u>2.4.3</u>	<u>Statusinformation</u>	22
<u>2.4.4</u>	<u>Hjälp och dokumentation</u>	22
<u>2.5</u>	<u>Startsekvens</u>	22
<u>2.6</u>	<u>Prestanda</u>	22
<u>3</u>	<u>Miljön och arbetsmetodiker</u>	23
<u>3.1</u>	<u>Intervju om arbetsmetodiker</u>	23
<u>3.2</u>	<u>Institutionen för Data- och Informationsfusion</u>	23
<u>3.3</u>	<u>Institutionen för Systemmodellering</u>	24
<u>3.4</u>	<u>Designmodell för SubTrack</u>	25
<u>4</u>	<u>Teoridel, orientering om användarvänlighet</u>	26
<u>4.1.1</u>	<u>Användarvänlighet</u>	26
<u>4.2</u>	<u>Exekverings- och utvärderingcykeln</u>	27
<u>4.2.1</u>	<u>Den interaktiva cykeln</u>	27
<u>4.2.2</u>	<u>Ramverket för interaktion</u>	28
<u>4.3</u>	<u>Shneidermans tre grundpelare</u>	28
<u>4.3.1</u>	<u>Riktlinjer och designprocesser</u>	28
<u>4.3.1.1</u>	<u>Riktlinjer</u>	28
<u>4.3.1.2</u>	<u>Designprocesser, prototyper</u>	29
<u>4.3.1.2.1</u>	<u>Evolutionär prototyputveckling</u>	30

4.3.1.2.2	”Throw-away” prototyputveckling	30
4.3.1.2.3	Inkrementell prototyputveckling	31
4.3.2	Verktyg för gränssnittsprototyper	31
4.3.3	Expertutvärderingar och användbarhetstest	32
4.3.3.1	Expertutvärdering	32
4.3.3.1.1	Kognitiv genomgång (expertutvärdering)	32
4.3.3.1.2	Heuristisk utvärdering (expertutvärdering)	33
4.4	Användardeltagande i designprocessen	33
4.5	Inmatning, användaren till systemet	35
4.6	Presentation, systemet till användaren	35
5	Designprocess, för ett användarvänligare SubTrack	36
5.1	Ursprunglig planering	36
5.2	Problem	36
5.3	Slutlig designmodell	36
6	Expertutvärdering av Sub Track	38
6.1.1	Syfte och mål med expertutvärderingen	38
6.1.2	Roller och aktörer	38
6.1.3	Användningsfall	38
6.1.4	Konventioner	38
6.2	SubTracks funktionalitet	39
6.2.1	Simulering	39
6.2.2	Beräkningsfunktion	39
6.2.3	Beräkningspresentation	39
6.2.4	Initieringsparametrar	40
6.2.5	Tillstånd	40
6.3	Användarkontroll	41
6.3.1	Menyer	41
6.3.2	Dialogfönster	43
6.3.3	Körkontroller	44
6.3.4	Zoom och panoreringarfunktioner	44
6.3.5	Direktmanipulation	44
6.4	Presentation	44
6.4.1	Huvudfönstret	44
6.4.2	Ritfunktion	45
6.4.3	Statusinformation	46
6.4.4	Hjälp och dokumentation	46
6.5	Startsekvens	46
6.6	Prestanda	46
7	Intervjuer med demonstratörer	47
7.1	Uppgiftsanalys	47
7.2	Användardeltagande i designförslag, användare A	48
7.2.1	Användartest	48
7.2.2	Användare A kommenterar SubTrack	48
7.3	Användardeltagande i designförslag, användare B	49
7.3.1	Användartestet	49
7.3.2	Användare B kommenterar SubTrack	51
7.4	Startsekvens	52

8 Resultat	53
8.1.1	Översikt 53
8.1.2	Begränsningar 53
8.1.3	Läsanvisningar 54
8.1.4	Roller och aktörer 55
8.1.5	Användningsfall 56
8.2 SubTracks funktionalitet	57
8.2.1	Simuleringen 57
8.2.2	Beräkningsfunktion 57
8.2.3	Beräkningspresentation 57
8.2.4	Programparametrar 57
8.2.5	Tillstånd 57
8.3 Användarkontroll	58
8.3.1	Menyrad 58
8.3.1.1	Nuvarande status 58
8.3.1.2	Designförslag 58
8.3.1.2.1	Huvudmenyn 58
8.3.1.2.2	Undermenyn 59
8.3.1.2.3	File 59
8.3.1.2.4	View 59
8.3.1.2.5	Tools 60
8.3.1.2.6	Help 60
8.3.1.3	Diskussion och argumentation 61
8.3.2	Dialogfönster 62
8.3.2.1	Nuvarande status 62
8.3.2.2	Designförslag 62
8.3.2.3	Diskussion och argumentation 64
8.3.3	Körkontroll 65
8.3.3.1	Nuvarande status 65
8.3.3.2	Designförslag 65
8.3.3.3	Diskussion och argumentation 66
8.3.4	Zoom- och panoreringsfunktion 67
8.3.4.1	Nuvarande status 67
8.3.4.2	Designförslag 67
8.3.4.3	Diskussion och argumentation 69
8.3.5	Direktmanipulation 70
8.3.5.1	Nuvarande status 70
8.3.5.2	Designförslag 70
8.3.5.3	Diskussion och argumentation 71
8.3.6	Initieringsparametrar 72
8.3.6.1	Nuvarande status 72
8.3.6.2	Designförslag 72
8.3.6.3	Diskussion och argumentation 74
8.4 Presentation	75
8.4.1	Huvudfönster 75
8.4.1.1	Nuvarande status 75
8.4.1.2	Designförslag 75
8.4.1.3	Diskussion och argumentation 75
8.4.2	Ritfunktionen 76
8.4.2.1	Nuvarande status 76

8.4.2.2	Designförslag	76
8.4.2.3	Diskussion och argumentation	78
8.4.3	Statusinformation	79
8.4.3.1	Nuvarande status	79
8.4.3.2	Designförslag	79
8.4.3.3	Diskussion	80
8.4.4	Hjälp och dokumentation	81
8.4.4.1	Nuvarande status	81
8.4.4.2	Designförslag	81
8.4.4.3	Diskussion och argumentation	81
8.5	Startsekvens	82
8.5.1	Nuvarande status	82
8.5.2	Designförslag	82
8.5.3	Diskussion och argumentation	83
8.6	Prestanda	84
9	Sammanfattning	85
9.1	Mål	85
9.2	Lösningssmetod	85
9.3	Resultat	86
9.3.1	Resultat enligt de heuristiska punkterna	86
9.3.2	Resultat enligt rapportens kapitelindelning	87
10	Referenser	88
11	Ordlista och akronymer	89
11.1	Ordlista	89
11.2	Akronymer	90

Figurförteckning

Figur 1: Omslagsbild, Copyright (c) 2002 David Gould	
Figur 2: En körning av en simulering med Kalmanfilter och automatisk bojutläggning. Ingen bojgrafik visas.	12
Figur 3: Användningsfallsdiagram över det befintliga systemet.....	144
Figur 4: Övergripande vy över det befintliga systemets samarbete (kollaborationsdiagram) Detta diagram är inte utfört strikt enligt UMLs riktlinjer, utan har tydliggjorts med textrutor.....	155
Figur 5: Körning av simuleringsssekvens med Kalmanfilter och automatisk bojutläggning. Innehåller alla grafiska komponenter som kan väljas med denna beräkningsmetod.....	177
Figur 6: Körning av simuleringsssekvens med Partikelfilter. De grafikkomponenter som inte kan väljas vid körning med Kalmanfilter visas här.	188
Figur 7: Befintligt menysystem.....	19
Figur 8: Befintligt zoom- och panoreringsdialogfönstret.....	21
Figur 9: Befintlig statusraden.....	22
Figur 10: Befintlig statusinformation om bojar.....	22
Figur 11: Relationen mellan användbarhet och användarvänlighet enligt Allwood.....	26
Figur 12: Exekverings- och utvärderingscykeln av Norman. Något mer tydliggjord.....	28
Figur 13: Houde och Hills modell för prototyper. Punkter beskriver tänkbara positioner för en prototyp inom modellen.....	29
Figur 14: Evolutionär prototypmetod.....	30
Figur 15: "Throw-away" prototypmetod.....	30
Figur 16: Inkrementell prototypmetod.....	31
Figur 17: Körning av simulering med Kalmanfilter.....	45
Figur 18: De dialogfönster som demonstratören vanligtvis öppnar, när en simulering skall köras i det befintliga SubTrack programmet.....	52
Figur 19: Programmets huvudkomponenter.....	54
Figur 20: Användningsfall för designförslaget.....	56
Figur 21: Menysystem. Om namnet avslutas med två punkter, öppnas en dialogruta när menyvalet aktiveras.....	59
Figur 22: Vilka grafiska komponenter som skall visas.....	63
Figur 23: Kontrollpanel för körning av simulering.....	65
Figur 24: Zoomning-, panorering- och återupprättningsfunktioner.....	68
Figur 25: Verktygsfältet. I detta dokument är verktygsfältet uppdelat i två delar p.g.a. utrymmesskäl. Innehåller de mest väsentliga funktionerna för körning av simulering, undantaget bojutläggning vilket sker direkt med mus.....	70
Figur 26: Initieringsparametrar. Generella inställningar.....	72
Figur 27: Initieringsparametrar. Inställningar för ickedynamiska grafikkomponenter. .	74
Figur 28: Målets grafikobjekt suddar ut de tidigare ritade ellipserna.....	77
Figur 29: Mål och prediktionsellipser som endast ritas väsentlig information.....	77
Figur 30: Statusrader. Den övre, till höger om menyn. Den undre, längst ned i programfönstret.....	79
Figur 31: Startdialog.....	83

1 Introduktion

Detta examensarbete kommer att behandla användarvänligheten hos en interaktiv datorbase-rad forskningsprototyp kallad SubTrack. Arbetet resulterar i ett designförslag för en utvecklad version av SubTrack, anpassad för en användare med rollen demonstratör.

SubTrack är ett datorprogram som utvecklats vid Totalförsvarets forskningsinstitut (FOI) för optimering av sonarbojutläggning vid ubåtsföljning. Vid första indikation placeras sonarbojar automatiskt ut enligt ett förutbestämt mönster, varefter bojarna rapporterar sin position och om de hör målet. Systemet beräknar med dessa uppgifter målets troliga position och visar denna grafiskt. Utläggningen av bojar sker antingen manuellt av användaren eller automatiskt av systemet. Den automatiska bojutläggningen beräknar den optimala positionen av bojen, så att den skall ha den bästa placeringen för den fortsatta simuleringen.

1.1 Roll

Utvärdering och förändring av programmet sker endast för en användarroll, demonstratören som redan har en grundläggande förståelse för vad programmet kan utföra. Demonstratören skall, med ett minimum av förberedelse, lätt kunna förevisa programmet för personer eller grupper. SubTrack är avsett att användas som demonstrator på möten eller mässor, för att skapa intresse för den forskning som gjorts.

Anledningen till att denna aktör väljs, är att demonstratörsrollen nyligen tillkommit i arbetet med SubTrack, varför en anpassning ännu inte gjorts för att uppfylla dennes behov. Det är viktigt att demonstratören kan hantera sina uppgifter, då den tid som hon har att sätta sig in i programmet ofta är begränsad och måste klara sig utan hjälp då demonstrationen ofta sker utanför den vanliga miljön.

1.2 Mål och syfte

SubTrack gränssnittsdesign föreslås förbättrat genom viss förändring av gränssnittet och vissa utökade möjligheter till interaktion.

Arbetet inriktas på att göra programmet användarvänligt, en undergrupp av användbarhet. Användbarhet har ytterligare undergrupper som är anpassning, acceptans och kompetens. Dessa berörs endast ytligt. Huvudsakligen används de heuristiska principerna och medverkan av användare för att designförslaget skall hålla en bra kvalitet.

Endast enstaka tillägg av funktionalitet till programmet föreslås. Den funktionalitet som programmet innehåller skall inte ha sin källa i användarvänlighet, utan i den forskning som ligger till grund för en datorbaserad prototyp. Den grundläggande funktionaliteten i programmet uppfyller idag de flesta behov som en demonstratör kan tänkas ha.

Ingen implementation av designförslaget utförs, varken som en ny version av SubTrack eller som prototyp som simulerar SubTracks beteende. Arbetet resulterar således endast i ett designförslag i rapportform.

Rapportens användning är främst som ett designförslag för användargränssnittet för SubTrack, vid en eventuell utveckling. Rapporten skall också kunna användas som ett exempel på en tillämpning av användarvänlig design, vid utveckling av liknande projekt. Användarna avses främst arbeta i projekt inom FOIs avdelning för Ledningssystem.

Om rapporten skall användas som stöd för utvecklingen i ett projekt, är det viktigt att anpassa metoder och verktyg till projektets förhållanden och skede i designprocessen. En identifiering av roller i projektet är en viktig början. Innan metoderna används är det viktigt att identifiera likheter och olikheter mellan SubTrack och det nya projektet, för att nå ett bra resultat.

1.3 Läsanvisning

Läsaren av denna rapport kommer i de första kapitlen, 2, 3 och 4, att få en beskrivning av SubTrack, miljön den finns i och teori om användarvänlighet. En utvärdering av SubTrack, kapitel 6 och 7, samt ett designförslag för ett mer användarvänligt SubTrack ges i kapitel 8. En diskussion om de resultat som uppnåts kommer i kapitel 9, följt av referenser, en ordlista, akronymer, samt bilagor.

Den orienterande delen börjar med kapitel 2, som ger läsaren en utförlig beskrivning av det befintliga SubTrack. Detta kapitel är strukturerat enligt samma kapitelindelning som expertutvärderingen och designförslaget, så att en jämförelse mellan dessa underlättas. Kapitel 3 innehåller en beskrivning av den arbetsmiljö där utvecklingen skett och de arbetsmetoder som nyttjas. Sista kapitlet i den orienterande delen är 4, som behandlar valda delar av teorier och metoder inom människa-datorinteraktion. Dessa teorier används i arbetet i expertutvärderingar och designförslag. Kapitel 5 beskriver den ursprungliga plan enligt vilket projektet skulle genomföras på, vilka problem som uppstod och hur det slutliga genomförandet skedde.

Utvärderingen, består av en expertutvärdering, kapitel 6, som genomförts med stöd av heuristiska punkter, en kognitiv genomgång samt användarintervjuer. Användarintervjuerna följer i kapitlet 7, men dess summering finns i det föregående kapitlet expertutvärdering. Kapitelindelningen för expertutvärderingen är kopplat mot indelningen av kapitel 4 och 8. Detta för att läsaren skall kunna följa en enskild del av systemet, från beskrivning, till utvärdering, och slutligen till designförslag.

Designförslaget får ett eget kapitel, nummer 8. Där presenteras ett designförslag för ett användarvänligare SubTrack. Efter detta kapitel följer en sammanfattning och diskussion om de resultat, som arbetet har kommit fram till, kapitel 9. Detta följs av de referenser som ingår i rapporten, kapitel 10. Sist kommer en ordlista som beskriver de termer som kan vara svåra eller tvetydiga, kapitel 11. Det finns även en akronymlista i detta kapitel.

1.3.1 Läsarkategorier

Det finns två huvudsakliga kategorier av läsare för denna rapport. Den första är läsare som vill ha en generell kunskap inom området användarvänlighet. Detta av allmänt intresse eller som stöd vid ett liknande projekt. Den andra gruppen läsare är de som arbetar med projektet SubTrack eller önskar en djupare kunskap i de förslag som utformats för att förbättra programmets gränssnitt.

Den första gruppen, de som redan har kunskap om SubTrack, kan behöva en repetition av vissa delar av programmet, det får läsaren i kapitel 2. I kapitel 3 följer en orientering i miljön och kapitel 4 ger läsaren en beskrivning av de verktyg och teorier som ligger till grund för den expertutvärdering och designförslag. Kapitel 5 behandlar den planerade process som skulle leda fram till ett nytt gränssnitt, men även de problem som uppkom och till sist hur det slutliga genomförandet skedde. Kapitel 6 beskriver de brister som identifierades och kan vara av intresse för en samlad bild av aktuell status. De intervjuer och det sätt de genomfördes går att utläsa i kapitel 7. Det slutliga resultatet finns i kapitel 8, numrerat så att olika delar skall vara spårbara tillbaka genom kapitel 6 och kapitel 2. För den i SubTrack insatta läsaren rekommenderas att sammanfattningen läses enligt kapitelindelning, 9.3.2.

Den andra gruppen, som söker en generell kunskap om människa-datorinteraktion, rekommenderas läsa resultatet i 9.3. Resultatet är uppdelat enligt de heuristiska punkterna, mer om detta i de *Heuristiska punkterna 4.3.3.1.2*. Fortsätt med att skumma igenom kapitel 3 för att skapa sig en bild av projektets miljö. Läsaren presenteras i kapitel 6 viktig läsning om de verktyg och teorier på vilket detta arbete vilar och i kapitel 5 återfinns beskrivningen av den process som fört arbetet framåt. Efter detta är det dags att läsaren skapar sig en bild av det

befintliga systemet, kapitel 3. Beträffande kapitel 6 och 8 väljs lämpligen något eller några delar som intresserar, annars riskerar läsaren att drunkna i information. Dessa delar har samma kapitelindelning och kan därför spåras lätt en föreslagen förändring från förslaget i kapitel 8 tillbaka till expertutvärdering i kapitel 6 och funktionen får en beskrivning av aktuell status i kapitel 3. Naturligtvis kan användaren gå motsatt väg om hon så önskar, men det är om kunskap saknas om SubTrack svårt att orientera sig i allt material. Kapitel 7 ger en beskrivning hur användardeltagande kan föras in i gränssnittsdesign, för att samla kunskap och för att skapa acceptans.

1.4 Terminologi

Med design åsyftas gränssnittsdesign.

Designprocess är den process som beskriver produktens olika utvecklingssteg.

Reaktiv planering innebär här, att aktuell sensorinformation från bojarna utnyttjas för att förbättra den fortsatta utplaceringen av nya bojar.

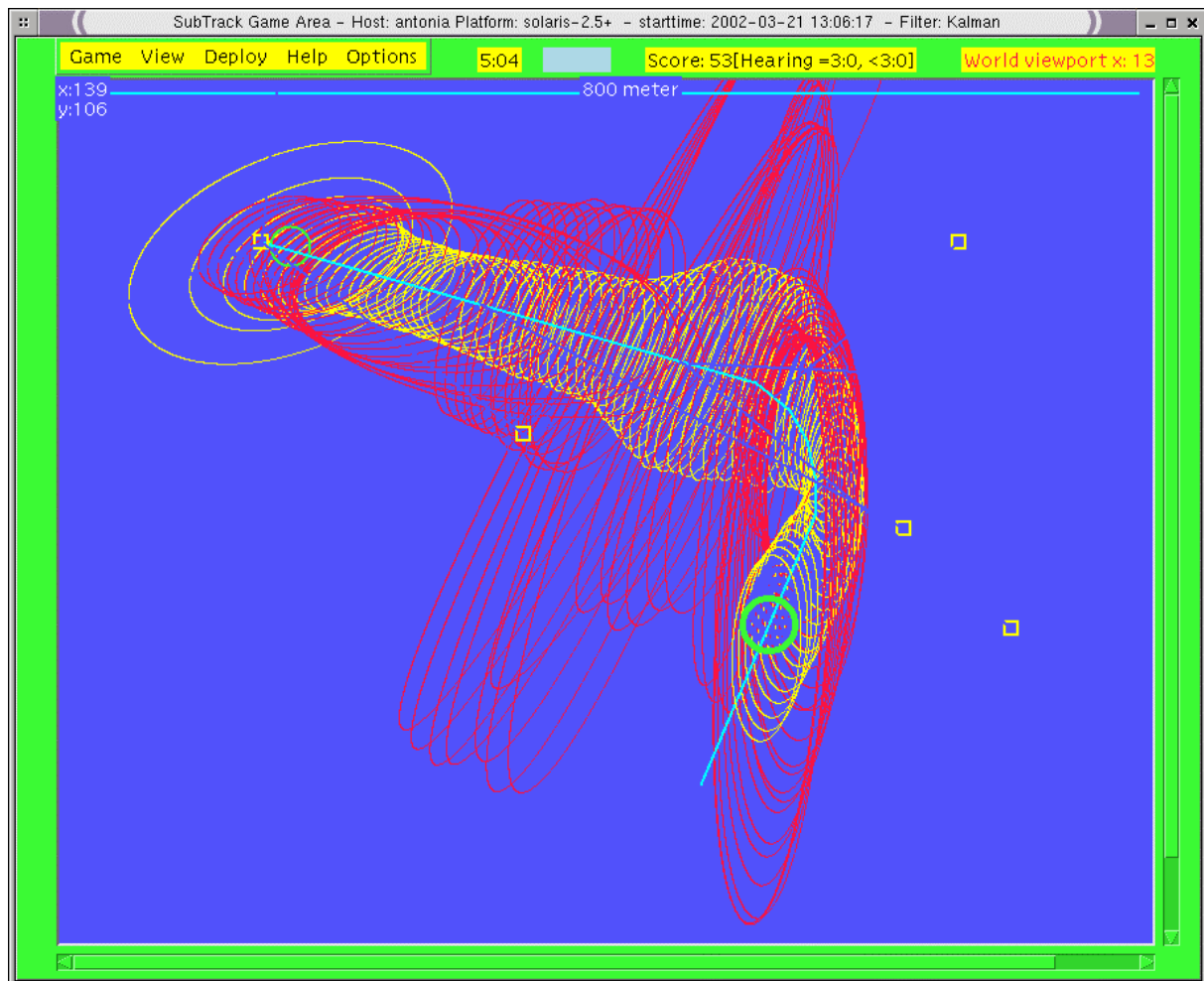
Simuleringssekvensen är den följd av händelser som sker mellan start till stopp av en simulering. Den har ibland gått under benämningen ”spel”.

Referenskorgen är en samling av tre program och en uppsättning av riktlinjer för design av ett gränssnitt. Dessa används som en referens för hur SubTracks gränssnitt skall utformas. Se *Konventioner, 6.1.4.*

2 Beskrivning av befintligt SubTrack

Detta kapitel beskriver hur det befintliga SubTrack fungerar, version 011001, med hjälp av bilder, UML-diagram och förklarande text. Kapitelindelningen har samma indelning som kapitlen om expertutvärderingen, kapitel 6, och designförslaget, kapitel 8. Syftet är att ge läsaren en god bild av hur SubTrack fungerar, men främst ge en beskrivning av hur interaktionen fungerar.

Kapitlet börjar med en beskrivning av de begränsningar som påverkar SubTrack. Funktionaliteten beskrivs i kapitel 2.2, för att senare i kapitel 2.3 behandla användarens kontroll över systemet. Sist ges en beskrivning av den presentation som systemet visar användaren, kapitel 2.4.



Figur 2: En körning av en simulering med Kalmanfilter och automatisk bojuetläggning. Ingen bojuetläggning visas.

2.1.1 Syfte och mål med SubTrack

Denna rapport är en del i en studie av problemet att simulera upptäckt och följning av en fiotlig ubåt med hjälp av sonarbojar i en grund skärgårdsmiljö. Forskningsprojektet har haft som syfte att undersöka tillämpbarheten av reaktiv planering och multisensordatafusion. Reaktiv planering innebär här att aktuell sensorinformation utnyttjas för att förbättra den fortsatta utplaceringen av bojar. Multisensordatafusion innebär att information från flera bojar kombineras i syfte att uppnå bättre mätresultat. Studien har i två metodrapporter [WFJS 96], [JS 97] presenterat lösningsförslag till delar av problemet.

Datorprogrammet SubTrack är ett verktyg för att simulera hur en operatör söker följa en okänd ubåtsbana genom att lägga ut en mängd passiva hydrofonbojar. Genom sammanvägning av bojarnas signaler skall målets position uppskattas och därefter predikteras. Positions-uppskattningen presenteras för operatören i varje tidssteg som en ellips som inrymmer målets position med en viss sannolikhet [WFJS 96]. Programmet har även optioner för automatisk bojutläggning och målföljning.

2.1.2 Roller och aktörer

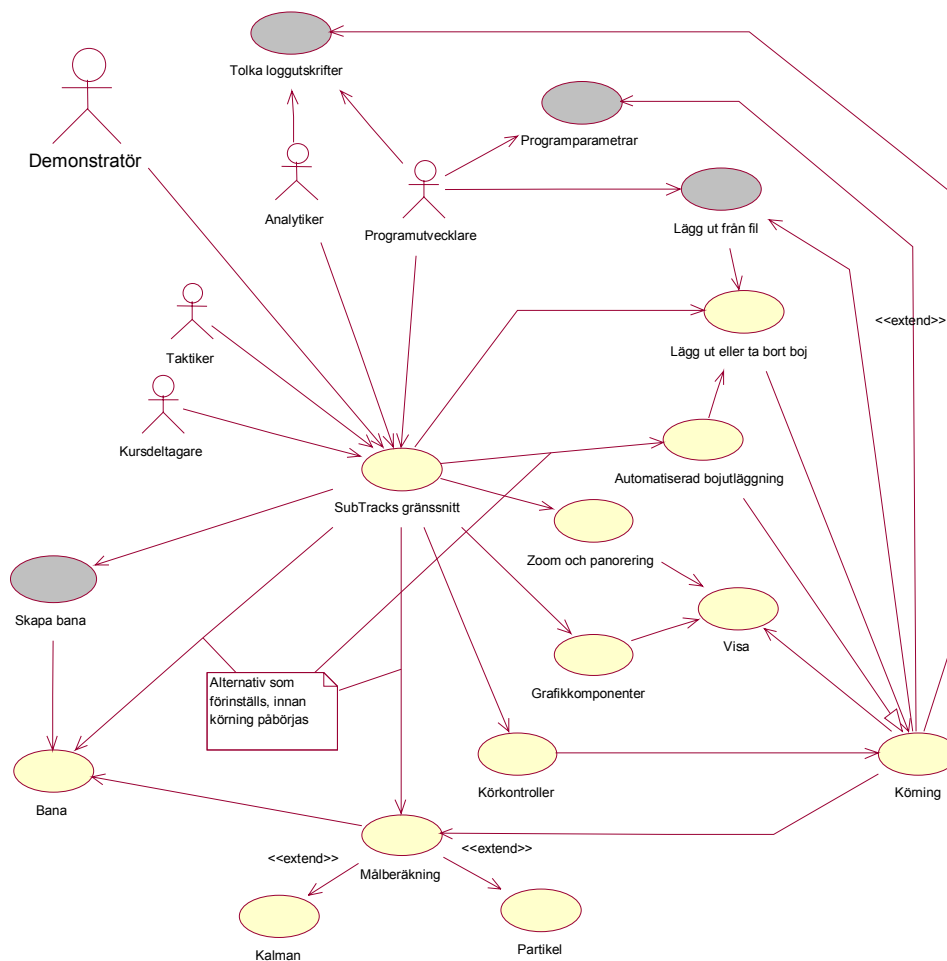
Programmet skall kunna nyttjas av olika användare med olika roller. Dessa har blivit fler under projektets fortskridande. Numrering sker enligt den ordning de uppkom:

1. Programutvecklaren – en användare som har i uppgift att utveckla programmet, samt behålla dess konsistens. En orsak kan vara att prototypen har visat på ytterligare möjligheter för SubTrack. Vidare skall programutvecklaren med enkla handgrepp kunna provköra SubTrack för att kontrollera att de förändringar som gjorts i programmet ger det resultat som var avsett.
2. Analytikern – analys av programmets funktionalitet, framför allt de matematiska delarna, skall underlättas genom att programmets parametrar på ett enkelt sätt kan manipuleras. Åtkomst av styrparametrar för loggfiler vid körning skall förenklas. Tillförlitlighet, få handgrepp och möjlighet till fördjupad hjälp är viktigt.
3. Taktikern – frivillig och intresserad användare för vilken en enkel och lättfattlig miljö skall skapas så att hon eller han på ett enkelt sätt kan använda programmet. Nyttjandet skall förbättra förståelsen för programmet och även skapa ett intresse för idén bakom SubTrack. Laboreringen kan vara att studera vilka egenskaper de föreslagna bojarna bör ha. Få handgrepp är viktigt.
4. Demonstratören – en användare med grundläggande förståelse för vad programmet kan erbjuda, skall enkelt kunna demonstrera programmet för personer eller grupper med olika bakgrund. Bakgrunden kan t.ex. vara militär, datateknik eller forskning. Ett mål är att åskådliggöra vilken potential forskningen har och att i annan eller vidareutvecklad form utanför FOI, klara uppgiften att följa en ubåt i ett skarpt läge. Tillförlitlighet, få handgrepp samt överskådlighet är viktiga egenskaper för programmet.
5. Kursdeltagaren – använder SubTrack för att den ingår i en kurs. En enkel och lättfattlig miljö skall skapas så att hon eller han med ett minimum av förberedelse och instruering skall kunna laborera med SubTrack. Möjlighet att återhämta från fel och få handgrepp är viktigt.

Eftersom det är tidskrävande att utveckla ett designförslag för alla användare, måste omfånget begränsas. Detta arbete begränsas därför till att behandla en användarroll, nämligen demonstratörens. Anledningen till detta val är att demonstratören är en sent tillkommen roll och att denne har liten tid att sätta sig in i programmet och måste klara sig utan en handledare. Demonstratören kan alltså inte förvänta sig ett program som kan köras utan instruktioner och träning, dock skall det vara intuitivt och lättlärt. Demonstratören har troligen en bakgrund inom utvecklingen av SubTrack och för att ett intresse skall finnas att låta denna person demonstrera programmet, besitter hon troligtvis någon specialkunskap som tillämpas av SubTrack.

Andra aktörer, som nyttjar samma funktionalitet som demonstratören, kan dra nytta av förändringar som designförslaget medför.

2.1.3 Användningsfall, UML-notation



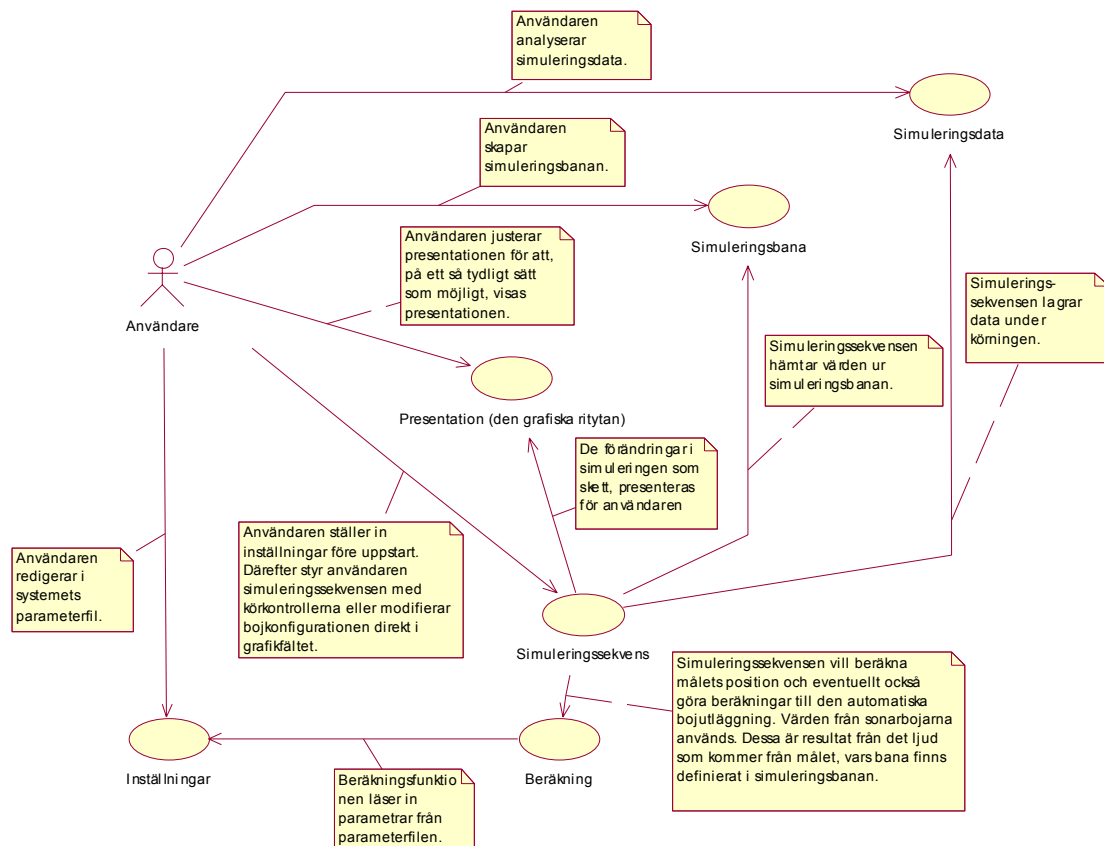
Figur 3: Användningsfallsdiagram över det befintliga systemet.

Beskrivningar av användningsfall är en högnivåabstraktion av systemet, som används för att skapa den kravspecifikation implementationen bygger på. Användningsfallsdiagrammet är på en hög abstraktionsnivå för att en tydlig bild av systemet skall framträda, även för personer utan någon kunskap om programmering.

Innan ett användningsdiagram skapas är det viktigt att rollerna, som systemet skall kunna fungera för, är noggrant analyserade. Aktörer är en bra utgångspunkt för att förstå och hitta de viktigaste användningsfallen. En analytiker bör dock vara medveten att systemet kan ses ur flera synvinklar, såsom t.ex. administratörens eller ledningens. Den viktigaste synvinkeln kan identifieras och användas som en mall för att strukturera kravspecifikationen.

Ett diagram över användningsfall kan ha olika omfattning av detaljer. Detaljeringsgraden bestäms av vilken betydelse det specifika användningsfallet har. Användningsfallen har gjorts med ”UML distilled” som stöd [Fowler].

2.2 SubTracks funktionalitet



Figur 4: Övergripande vy över det befintliga systemets samarbete (kollaborationsdiagram) Detta diagram är inte utfört strikt enligt UMLs riktlinjer, utan har tydliggjorts med textrutor.

Simuleringssekvensen har i andra sammanhang liknats vid ett spel, därav heter huvudmenyn ”Game”. I fortsättningen kommer den dock att refereras till som simulering.

Operatören laddar in en målbana från en fil (skapad med program PathEd) varvid en röd cirkel på skärmen visar var ubåtsbanan börjar. När operatören har startat simuleringen kan han med musklickning på valfri plats lägga ut enskilda bojar, eller grupper av bojar, i olika mönster. Har operatören valt automatisk bojutläggning, sköter programmet om utplacering av nya bojar, operatören kan dock lägga till och ta bort bojar enligt egna önskemål.

Programmet beräknar vid varje tidpunkt och för varje boj om målet hörs. Om minst fyra bojar hör målet samtidigt, beräknas ett område som med viss sannolikhet innehåller målets position.

Om området ej är alltför stort, presenteras området som en ellips, vars yta, utsträckning och riktning, ger ett mått på noggrannheten i positionsberäkningen.

Operatören kan studera taktik och bojegenskaper genom att frysa tiden, varefter målets buller vid den aktuella tidpunkten består och operatören kan pröva effekten av olika bojutplaceringar. Slumpningen fortsätter dock inte under det frysta tillståndet av spelsekvensen, eftersom detta hade resulterat i att mätellipsen hade rört sig fram och tillbaka, och därmed omöjliggjort jämförelse mellan olika bojkonfigurationer. Att ellipsen inte skulle ligga stilla, beror på att en slumpfaktor skulle ha slumpat ut nya osäkerhetsvariabler för bojarnas position, vid varje ny beräkning.

När en simulering har nått banans slut, inträder automatiskt stopptillståndet och ingen förändring av simuleringen är längre möjlig. För att starta en ny simulering måste programmet avslutas och startas på nytt.

Vissa resultat från bojutläggning och mätdata sparas för efterföljande analys. Alla operatörskommandon finns samtidigt tillgängliga och operatören får själv ta beslut om vilka som skall vara tillåtna i en viss situation, vilken beror på avsikten med spelet: att tävla om bästa målföljning, att träna användning och arbetsätt eller att studera egenskaper hos olika parametrar eller programmets beteende.

2.2.1 Simulering

Programmet är avsett att användas för olika syften, beroende på vilken roll användaren har. Återigen bör poängteras, att endast rollen som demonstratör kommer att diskuteras i rapporten.

De olika syften som en simulering med SubTrack kan användas till:

- studera sambandet mellan målets och bojarnas parametrar för ökad förståelse.
- studera metoder att placera ut bojar på bästa sätt för en given ubåtsbana.
- träna en operatör till att förstå spelet och uppnå bästa resultat.
- studera den automatiska bojutläggningen.
- tävla mellan operatörsstyrd och automatisk bojutläggning, jämföra antalet utlagda bojar.
- ge underlag för att formulera krav på bojars egenskaper.

2.2.2 Beräkningsfunktion

En stor del av arbetet med SubTrack har varit fokuserat på att tillämpa olika beräkningsalgoritmer. Denna rapport kommer inte att beröra någon av dessa, utan syftar endast till att underlätta hanteringen av programmet. För den intresserade finns nedan en kortfattad beskrivning:

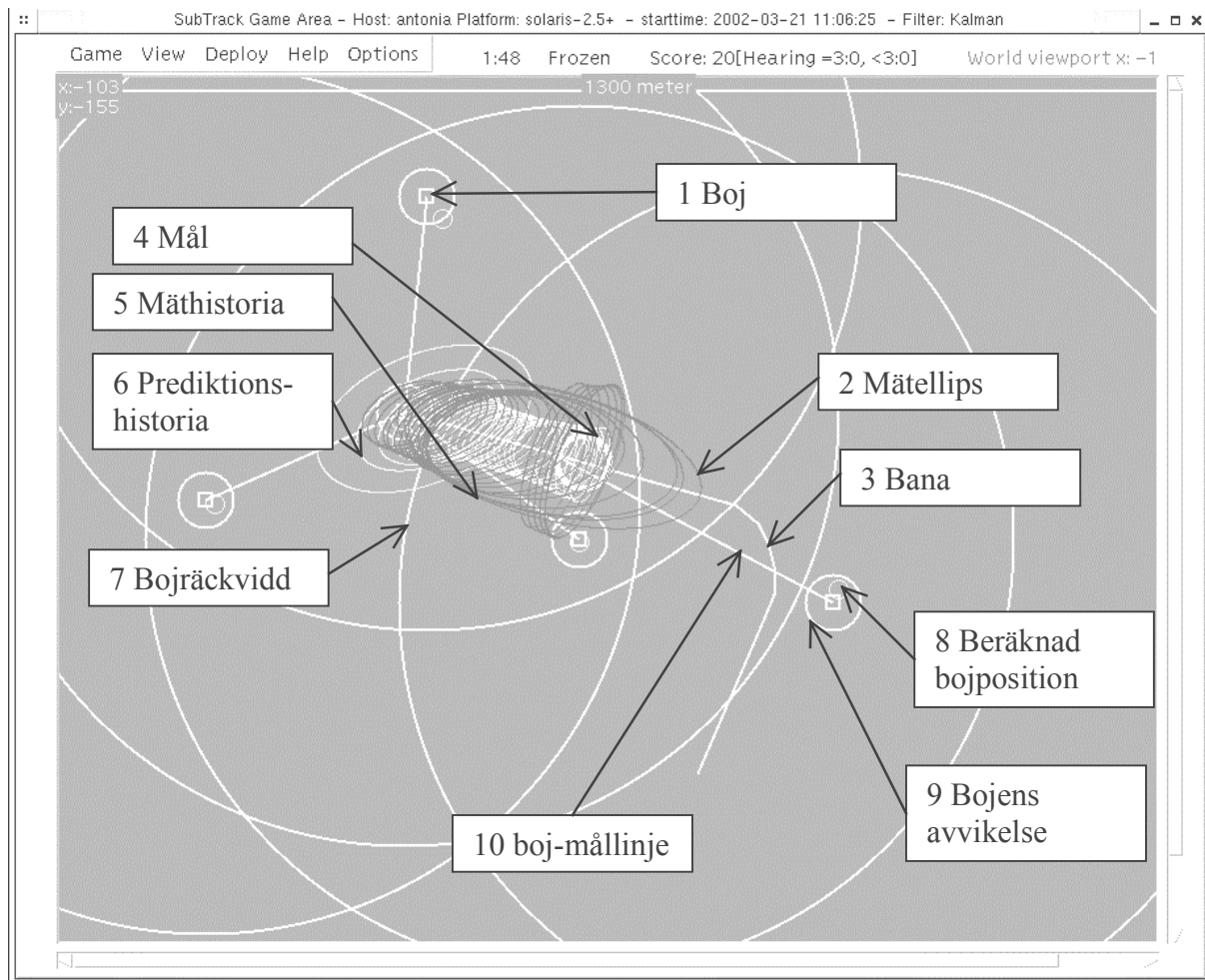
Beräkningarna i programmet utförs med två alternativa metoder. De betecknas Kalmanfilter respektive Partikelfilter.

Metoden med Kalmanfilter innebär att ubåtens position beräknas med en analytisk metod, minstakvadratmetoden. Bojarnas (simulerat uppmätta) positioner, anses ha ett normalfördelat fel med känd varians. Bojutläggningsalgoritmen utnyttjar ett Kalmanfilter, baserat på en enkel rörelsemodell, för att prediktera framtida positioner. Utläggningssalgoritmen utnyttjar den analytiska algoritmen för att pröva och bedöma tilltänkta bojplaceringars kvalitet.

Partikelfiltret innebär att en Monte Carlo metod beräknar ubåtens position och prediktion av dess rörelse. Ett så kallat partikelmoln av samplingar ur tillståndsrummet för målet (ubåten) används. Molnet uppdateras statistiskt, med hänsyn till nya (simulerade) mätningar. De analytiska formlerna för positionsberäkning kan helt undvikas, men i gengäld måste ett ganska stort antal enkla beräkningar göras på ”partiklarna”. Osäkerhetsmålet för bojarnas positioner är samma som i Kalmanfallet. Bojplaceringarnas kvalitet bedöms med ledning av predikterad framtida spridning i partikelmolnet.

2.2.3 Beräkningspresentation

Det grafiska fönstret ger en presentation av spelets händelser och kan ses som en virtuell verklighet. Användaren kan interagera med simuleringsssekvensen direkt, eller genom menyalternativen. Dessa förändringar presenteras direkt i det grafikfönstret.



Figur 5: Körning av simuleringssekvens med Kalmanfilter och automatisk bojutläggning. Innehåller alla grafiska komponenter som kan väljas med denna beräkningsmetod.

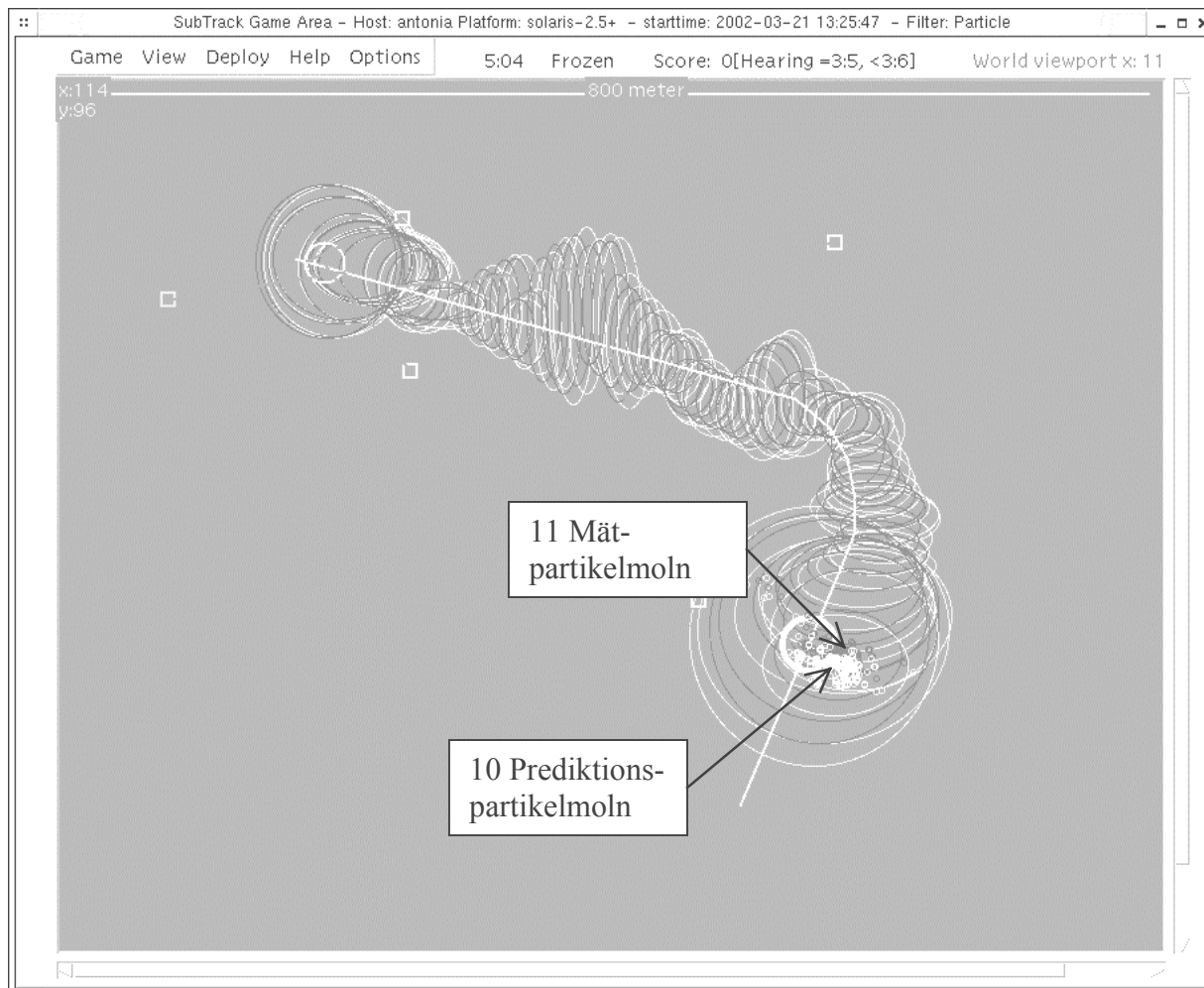
En grafisk komponent som visualiseras i grafikfönstret. Den kan var förvald eller vald av användaren.

Förvald grafikkomponent:

1. Boj ("Buoy") – en gul fyrkant, som blir ljusgrå om den är inaktiv.
2. Mätellips ("Measurement ellipse") – röd ellips, som ritas om varje tidssteg.

Användarvald grafikkomponent:

3. Bana ("Path") – visar den bana som målet kommer att följa.
4. Mål ("Target") – visar målets aktuella position.
5. Mäthistorik ("Measurement ellipse history") – visar alla mätellipser som skapats under simuleringen.
6. Prediktionshistorik ("Prediction ellipse history") – visar var beräkningsfunktionen, med en viss sannolikhet, beräknar att målet är. Den visar också alla tidigare ritade ellipser.
7. Bojräckvidd ("Buoy range") – visar den räckvidd där varje hörande boj uppfattar ljud
8. Beräknad bojposition ("Estimated buoy position") – den position som bojen mäter att den har i ett positioneringssystem, är inte alltid korrekt. Detta simuleras genom att en slumpfaktor adderas till positionsangivelsen.
9. Bojens avvikelse ("Buoy position disturbance") – med 90% konfidens (sannolikhet) befinner sig bojens position inom denna cirkel.
10. Boj-mållinje ("Buoy-target connection line") – linje mellan bojens position och den position som målet hade när bojen lades ut.



Figur 6: Körning av simuleringssekvens med Partikelfilter. De grafikkomponenter som inte kan väljas vid körning med Kalmanfilter visas här.

Vid val av partikelmetoden, för beräkning av målets position och prediktion, måste partikelmoln för inmätning väljas av användaren. Det finns inte, som för Kalmanmetoden, ett förval av partikelmoln för inmätning. De grafiska komponenterna för Partikelfiltermetoden är:

11. Prediktionspartikelmoln ("Particle prediction cloud") – var målet förutsågs vara, genom beräkning med Partikelfiltermetoden.
12. Mätpartikelmoln ("Particle estimate cloud") – var målet mäts in, genom beräkning med Partikelfiltermetoden.

2.2.4 Programparametrar för initiering

Initieringsparametrarna ligger i en fil som läses in vid programstart. Vissa av dessa parametrar kan det finnas intresse att redigera, för att demonstratören skall påvisa vissa beteenden hos SubTrack. Andra kan däremot inte tillåtas att ändras i initieringsfilen, dels för att den innehåller grundläggande fysikaliska egenskaper såsom ljudets hastighet, dels kan parametrarna vara av för komplex natur för att enkelt kunna redigeras. För mer information om dessa parametrar, se *Initieringsparametrar 8.3.6*.

2.2.5 Tillstånd

SubTracks tillstånd kan delas in i tre grupper, simuleringstillstånd, mustillstånd och tangentbordstillstånd. Simuleringstillståndet är ryggraden i SubTrack och kan genom användaren inträda i kör-, stopp-, respektive frystillstånd. Det finns ett initialt tillstånd vid programstart och ett när programmet nått slutet på simuleringsbanan som sätts av systemet. Mustillstånd

och tangentbordstillstånd är två interaktionstillstånd som förändras antingen av användaren eller automatiskt av systemet.

Under körning eller fryst läge befinner sig musen i ett normaltillstånd, då utplacering och borttagning av bojar kan ske genom ett musklick från användaren. Bojar läggs ut med vänster musknapp och tas bort med höger. Musen inträder, när stopptillstånd inträtt för simuleringen, ett tillstånd där planerad bojutläggning kan ske, användaren tillåts att placera ut 10 stycken bojar bestämda med position och inmatat tidssteg. Denna funktionalitet är mindre lämplig, enligt en av utvecklarna. Funktionen är inte intuitiv och användaren når inte funktionaliteten genom lämpliga kommandon. Användaren måste bli informerad om funktionen eller upptäcka den genom en slump. Ingen logisk förklaring ges till användaren, om vad som sker.

Musen behövs för ytterligare uppgifter i grafikytan, nämligen zoomning och panorering. Det är tillstånd som aktiveras i dialogfönster, och användaren får återkoppling genom att muspekaren förändras.

Inmatningsfält kan inträda i ett tillstånd, då det kan ta emot inmatning från tangentbordet. Detta sker genom att användaren klickar med musen i ett inmatningsfält. Först därefter kan användaren mata in önskade värden. För att inmatningsfältet skall registrera inmatningarna, krävs ett returslag på tangentbordet.

2.3 Användarkontroll

Interaktionen i programmet begränsas till interaktionsformen WIMP, se *Inmatning, användaren till systemet, 4.5*. Det går även att starta från en kommandorad, och samtidigt också skicka med argument vid start av programmet.

2.3.1 Menyner

I beskrivningen av menysystemet är huvudmenyn markerad i fetstil, både i figuren nedan och i den efterföljande texten.

Game	View	Deploy	Help	Options
Load	Tracks	Single	About	Tracking method
Run	Files	Lattice	Batch File	
Run until	Buoys	Square	Manual	
Step time	Path Edit	Triangle	Mouse	
Stop	Zoom/Pan	Circle	Score	
Freeze	Refresh	Random	DeployPattern	
Screen Copy		from file	SetUp file	
Exit			Zoom/Pan/Refr	

Figur 7: Befintligt menysystem.

Game	Hanterar laddning av bana, körning av simulering, skärmdump och avslut av programmet.
Load	Laddar den bana, som simulerar en okänd ubåt.
Run	De nedanstående menyvalen är körkontroller för simuleringssekvensen.
Run until	
Step time	
Stop	
Freeze	

Screen Copy Exit	Skärmdump av grafikfönstret, fungerar inte. Kommer inte att behandlas. Avslutar programmet
View	Hanterar grafikobjekt, dynamisk information, statisk information, start av tillhörande program och grafikfönsterfunktioner.
Tracks	Dialogfönster där inställning sker, av vilka grafiska objekt som skall visas.
Files	Visar vilka filer som systemet använder sig av.
Buoys	Visar befintlig status för bojarna.
Path Edit	Startar ett program som skapar banor, som simuleringssekvensen använder sig av.
Zoom/Pan	Öppnar ett dialogfönster som tillåter användaren att zooma, och att panorera, dvs. flytta fokus på den grafiska ritytan. Användaren kan också begära att hela banan ritas ut eller att hela banan och alla bojar ritas ut. Se <i>Zoom och panoreringsfunktioner, 2.3.4.</i>
Refresh	Ritar om det grafiska fönstret.
Deploy	Anpassningar vid utläggning av boj, men inte om utläggningsmetod skall vara manuell eller automatisk.
Single Lattice Square Triangle Circle Random from file	De nedanstående menyvalen är mönster för manuell utläggning. En förutbestämd placering och tid har skapats för en bana. Denna används istället för manuell eller automatisk utläggning.
Help	Hjälp och information om SubTrack.

Options

Tracking Method Användaren kan välja beräkningsmetod och bojutläggningsmetod.

2.3.2 Dialogfönster

En speciell variant av dialogfönster är en palett. Det är ett fönster som innehåller verktyg, som t.ex. används till att zooma och panorera. I rapporten kommer ingen skillnad göras mellan paletter och dialogfönster, trots att paletter fyller en något annorlunda funktion.

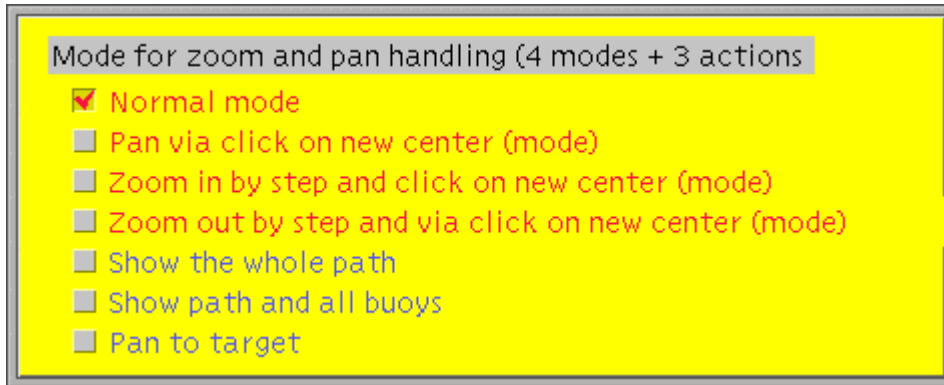
Dialogfönstren och paletterna finns i olika storlekar och i olika utseenden. Det är inte alltid konsekvent namngivning mellan menyamn och dialogfönsternamn, men det är menynamnet som kommer att användas för identifiering av funktionaliteten. Dialogfönstren kommer inte att beskrivas närmare i detta kapitel, utan de beskrivs mer utförligt i expertutvärdering och designförslag.

2.3.3 Körkontroller

Körkontroller används för att styra simuleringssekvensen, med kommandon som kör, stopp, frys osv. Samtliga återfinns i undermenyer, under huvudmenyn "Game".

Körkommandona liknar en bandspelare och en CD-spelare, men denna metafor är dock inte helt genomförd.

2.3.4 Zoom och panoreringsfunktioner



Figur 8: Befintligt zoom- och panoreringsdialogfönstret.

Under ”View” och ”Zoom/Pan” återfinns kommandot som öppnar dialogrutan där zoom- och panoreringsfunktionerna återfinns. Det fönster som öppnas är indelat i två grupper. Den första består av funktioner som förändrar musens tillstånd, och agerar genom direktmanipulation, t.ex. zooma ut. Den andra består av funktioner som exekverar ett kommando, t.ex. visa hela banan. Grupperingen sker genom att färgerna på funktionsnamnen är olika.

2.3.5 Direktmanipulation

Direktmanipulationen är en naturlig del i en interaktionsmetod av typen WIMP, där pekaren utför direktmanipulation. Direktmanipulationen består av två grupper, som båda fungerar i grafikfönstret. Grupperna är:

- Grafikfönster – bojar som läggs ut eller som tas bort från simuleringen.
- Zoom och panoreringsfunktioner – dessa manipulerar den presentation användaren visas, genom att påverka systemets uppritning i grafikfönstret.

2.4 Presentation

Programmet består av fyra huvudkomponenter som alla syftar till att visualisera en kategori. Vissa är statiska och förändras inte annat än inom sina ramar, t.ex. menyraden. Andra är dynamiska till sin natur, såsom grafikfönstret:

- Grafikfönster som visualiserar simuleringsssekvensen, men även ger möjlighet till interaktion genom direktmanipulation med musen.
- Statusraden visar intressant statusinformation om simuleringen. Statusraden återfinns till höger om menyraden.
- Menyraden ger användaren möjlighet att styra programmets uppträdande.
- Dialogfönster, som alla öppnas genom menyraden.

Presentationen är inte anpassad för en annan pixelupplösning än 1024*768. SubTrack är en prototyp, varför användaren tvingas anpassa sig till denna skärmstorlek.

2.4.1 Huvudfönstret

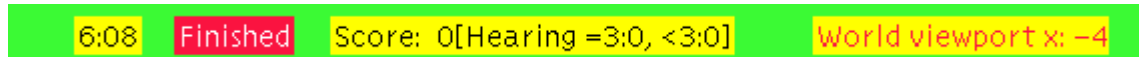
Huvudfönster är ramen som omsluter alla andra komponenter, som visualiseras.

2.4.2 Ritfunktion

Ritfunktioner har redan beskrivits, för att ge läsaren en uppfattning om vad programmet kan utföra. Se *Beräkningspresentation*, 2.2.3.

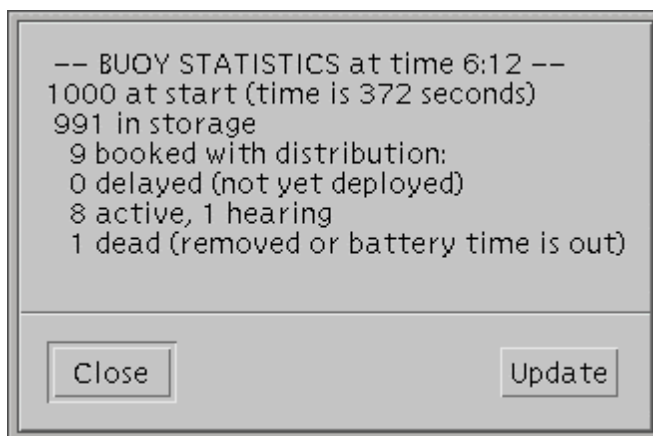
2.4.3 Statusinformation

Statusinformationen återfinns på två ställen, nämligen i statusraden höger om menyraden och i menyvalet "Buoys", under "View". Statusraden innehåller information, som kan vara viktig, för en användare under en körning. Det aktuella tidssteget är placerat först från vänster, centralt i användarens blickfång. Därefter kommer aktuellt tillstånd för simuleringsssekvensen; visas både i färg och i text. Nästa objekt är "Score" som visar endast värden för hur bra målföljningen har varit, beroende på antal hörande bojar. Finns endast för Kalmanfiltret, saknar mening för Partikelfiltermetoden. Slutligen ett mått på grafikytans koordinater längst till höger.



Figur 9: Befintlig statusraden.

Menyvalet "Buoys" innehåller information om antal tillgängliga bojar, antal använda, antal utplacerade, antal planerade men inte utplacerade, antal hörande och antal inaktiva.



Figur 10: Befintlig statusinformation om bojar.

2.4.4 Hjälp och dokumentation

Hjälpen har uppkommit ad hoc under programmets utveckling och formulerats ur utvecklarens användningsroll. Det finns även hjälp som beskriver systemets uppbyggnad. Detta är användbart när systemet inte uppför sig normalt, kanske för att alla systemparametrar inte är korrekta. Inga dialogfönster innehåller hjälpknappar, utan användaren är tvungen att söka denna hjälp under huvudmenyn "Help".

2.5 Startsekvens

Ett kommande kapitel för 6 och 8.

2.6 Prestanda

Interaktionsmöjligheten upphör när Partikelfiltret börjar beräkna var bojar skall läggas ut, eller när en bana representeras av en splinefunktion. Partikelfiltrets beräkning stjäl all processorkraft, och lämnar inget utrymme för interaktion från användaren. Problemet med splinefunktionen beror troligast på att beräkningen av banan blir för komplex.

3 Miljön och arbetsmetodiker

Detta kapitel beskriver arbetsmiljö och arbetsmetoder som används inom en av de programutvecklande grupperna inom FOI. Syftet med detta kapitel är att orientera läsaren om den miljö där SubTrack har utvecklats och där det också skall fungera.

De ingående kapitlen behandlar intervjuer med anställda som arbetar inom forskningsprojekt samt en beskrivning av det sätt enligt vilket SubTrack hittills har utvecklats på. Det finns i kapitlet benämningar på metoder att skapa prototyper, se *Designprocesser, prototyper 4.3.1.2*.

3.1 Intervju om arbetsmetodiker

Intervjuer gjordes med anställda på institutionen för Data- och Informationsfusion och institutionen för Systemmodellering, för att utröna hur arbetsmetoderna såg ut vid framtagandet av datorbaserade prototyper för demonstration av forskning och resultat. De personer som intervjuades har alla en längre erfarenhet av projektarbeten inom FOI, och tidigare FOA. Vissa av dessa har även en bakgrund som projektledare.

Syftet med intervjuerna var att erhålla en förståelse för den miljö där SubTrack skapats. Målet var att på ett bra sätt kunna skapa ett användarvänligare SubTrack, genom ett lämpligare val av designmodell för utvecklingsprocessen.

En slutsats som kan dras av intervjuerna, var att merparten av de program som skapas på FOI är prototyper för att uppvisa någon form av forskning och resultat. Som kapitel *Designprocesser, prototyper 4.3.1.2*, beskriver, delas prototyper vanligtvis upp i tre kategorier, evolutionär, ”Throw-away” och inkrementell.

3.2 Institutionen för Data- och Informationsfusion

Enligt intervjuerna, användes inom avdelningen för Data- och Informationsfusion oftast den evolutionära metoden, eftersom målen ständigt flyttas framåt och förändras. Vid vissa tillfällen var målen tydliga, så att skapandet av prototyper kan liknas vid den inkrementella metoden. Prototypen validerades med jämna mellanrum av projektmedlemmarna, för att se om prototypen uppfyllde de krav som kunde ställas på programmet.

När skapandet av prototypen skedde enligt inkrementell metod, skapades stubbar av de väsentliga delar som identifierades i systemet. Stubbar är ofullständiga komponenter i ett system, de utgörs i det tidiga skedet endast av en beskrivning av komponentens gränssnitt och av dess funktionalitet. Implementationen av dessa stubbar sorterades efter prioritet, beroende på hur svår denna del ansågs vara att förverkliga, samt hur väsentlig denna del är för systemet som helhet.

Den metod som utvecklare A använder är:

- Starten sker inkrementellt. Det finns en uppgift att utforska, och metoden att programmera en prototyp är enkel, för att påvisa denna forskning. Ingen större hänsyn tas till andra användare av prototypen, än utvecklaren självt. Det är forskaren själv som skall kunna visa att ett område är intressant, för att kunna få fortsatta resurser till forskningen.
- Han fortsätter iterera, tills de primära målen har uppnåtts. Budgeten är vanligtvis begränsad; den räcker inte till någon avancerad interaktion. De personer som skall bedöma projektets potential har en stor erfarenhet av dessa situationer; de förleds inte av en glättad yta.
- Om klartecken för projektet ges, övergår designprocessen till ett evolutionärt skede.

Utvecklare A har tre ledstjärnor:

- Spårbarhet – utvecklingsstegen, indata samt utdata, måste kunna spåras beträffande ursprung, tid osv.
- Versionshantering – ett evolutionärt arbete måste erbjuda möjlighet att backa till en tidigare version.
- Transparens – andra utvecklare och användare måste ha ett klart gränssnitt att koppla mot. Ingen förändring av in- och utdata skall ske, mellan olika versioner.

När utvecklare B arbetar i ett projekt, går mycket på rutin och han använder vad som fungerat tidigare. Modellen som används, är till stor del erfarenhetsbaserad. Eftersom ett forskningsarbete ofta inte har ett klart mål är det väsentligt att kunna backa tillbaka till den tidigare versionen, om en ny version av ett program inte fungerar som avsett. Mycket av arbetet bedrivs ad hoc, det görs lite oplanerat, ”i farten”.

En del av forskningen slutförs aldrig, varför en implementation av användarvänlighet i slutet av projektet är att föredra. I vilket skede som användarvänlighet skall komma i designprocessen, är beroende på projektets storlek. Vad som är ett stort projekt, beror både på antal medarbetare och på den tid som lagts ned i projektet.

3.3 Institutionen för Systemmodellering

Inom institutionen för systemmodellering bedrivs enligt utvecklare C arbetet efter två huvudspår. Institutionen för systemmodellering står, så att säga, på två ben. Det första området är metodikforskning inom modellering och simulering (M&S), den andra är tillämpad modellering av olika domäner. Vad som skall levereras är kunskap och i vissa fall är denna kunskap paketerad som en modell.

Metodikforskning inom modellering och simulering, M&S

Om forskning om nya metoder skall göras, används oftast ”throw-away” prototyper. Dessa prototyper används av forskaren för att verifiera sina resultat genom simuleringar med tänkbara scenarion, men förs sällan vidare till brukarna av forskningsresultaten. Prototypen tillför sällan en användare av rapportresultaten ett mervärde genom visualisering eller möjlighet till interaktion med prototypen.

Tillämpad modellering

En modelleringsexpert som deltar i ett projekt, där syftet är att modellera en domän, har sällan ansvaret för ledning och planering av projektet. Arbetsprocessen kan variera från veritabelt kaos till formella arbetsprocesser, beroende på vem som är beställare eller initiativtagare. En kaotisk process kan vara något som valts som metod för att målet varit oklart, eller att kaoset skall bidra till att alla lösningar skall få chansen att prövas. Vid tillämpad modellering, skapas de datorbaserade prototyperna främst genom evolutionära och inkrementella tillvägagångssätt.

Vanligtvis finns inga tydliga mål, designprocessen skall resultera i identifieringen av behoven. Målet är i varje fall inte att skapa en modell. Oftast är kunskapen om området så otydlig att processen kan liknas en evolutionär utveckling av prototypen. Efter varje färdig implementation värderas produkten och beslut tas om funktionaliteten är till belåtenhet.

I vissa situationer väcker en prototyp ett intresse hos en kund. I detta skede övergår prototypprocessen från evolutionär till inkrementell. I första iterationen implementeras de delar som är svårast och mest väsentliga. Ovanligare, men dock existerande, är att forskningsprototypen redan har klart definierade mål. I detta fall kan processen direkt liknas vid en inkrementell process.

3.4 Designmodell för SubTrack

SubTrack har utvecklats av tre FOI-anställda och tre examensarbetare. De FOI-anställda har gjort inledande design av systemet, formulerat mål och styrt examensarbetarna. Senare har de skött vidareutvecklingen av programmet.

SubTracks mål har varierat med uppnådda framgångar och efter genomförande av delmål. Detta har påverkat den designprocess som används för skapandet av SubTrack. De huvudsakliga milstolpar som uppnåtts under vägen är:

- Visa att målets position kan bestämmas, med given konfidens, med sonarbojar för positionering av ett undervattensmål.
- Kan användaren följa ett mål genom att styra bojutläggningen utgående från osäkerheten i positionsbestämningen? Detta kallas reaktiv planering.
- Kan operatören ersättas med automatiserad följning av målet och automatisk utläggning av bojar?
- Prova och jämföra andra beräkningsalgoritmer för målföljning och bojutläggning.
- Demonstrera resultaten för andra forskare och intressenter inom den marina domänen.

Det är först i det sista skedet med demonstrationerna, som användarvänlighet blivit riktigt aktuellt. Demonstratörens uppgift behöver underlättas så att åskådarna enkelt skall kunna urskilja hur demonstratören arbetar och följa de sekvenser som demonstrationen består av.

Idén till SubTrack väcktes 1994 och förverkligades med olika mål och avgränsningar under tre olika etapper:

- 95-96 Modell av mätdatainsamling, samordning och design av manuell bojutläggning. Denna del av utvecklingen hade tydliga milstolpar att uppnå, det var frågan om en inkrementell utveckling av prototypen. Under vissa skeden var dock delmålen otydliga, varför de inblandade utvecklarna provade sig fram till en lösning, en evolutionär utveckling.
- 96-97 Automatisk målföljning och bojutläggning, baserad på Kalmanfiltermetoden. Utvecklingen fick ske enligt en evolutionär metod. Det övergripande målet var klart, men det gick inte att sätta upp en uppsättning milstolpar varför detta räknas som en evolutionär utvecklingsperiod.
- 98-01 Automatisk målföljning och bojutläggning, baserad på Partikelfiltermetoden. Denna etapp påbörjades av FOIs engelska motsvarighet, DERA. Designprocessen blir tydlig först när arbetet med att överföra DERAs resultat till programspråket Eiffel påbörjas inom FOI. Konverteringen skedde huvudsakligen av utvecklare A, som ensam arbetade med programmet enligt en ”Top-Down” metod, inkrementell utveckling.

4 Teoridel, orientering om användbarhet

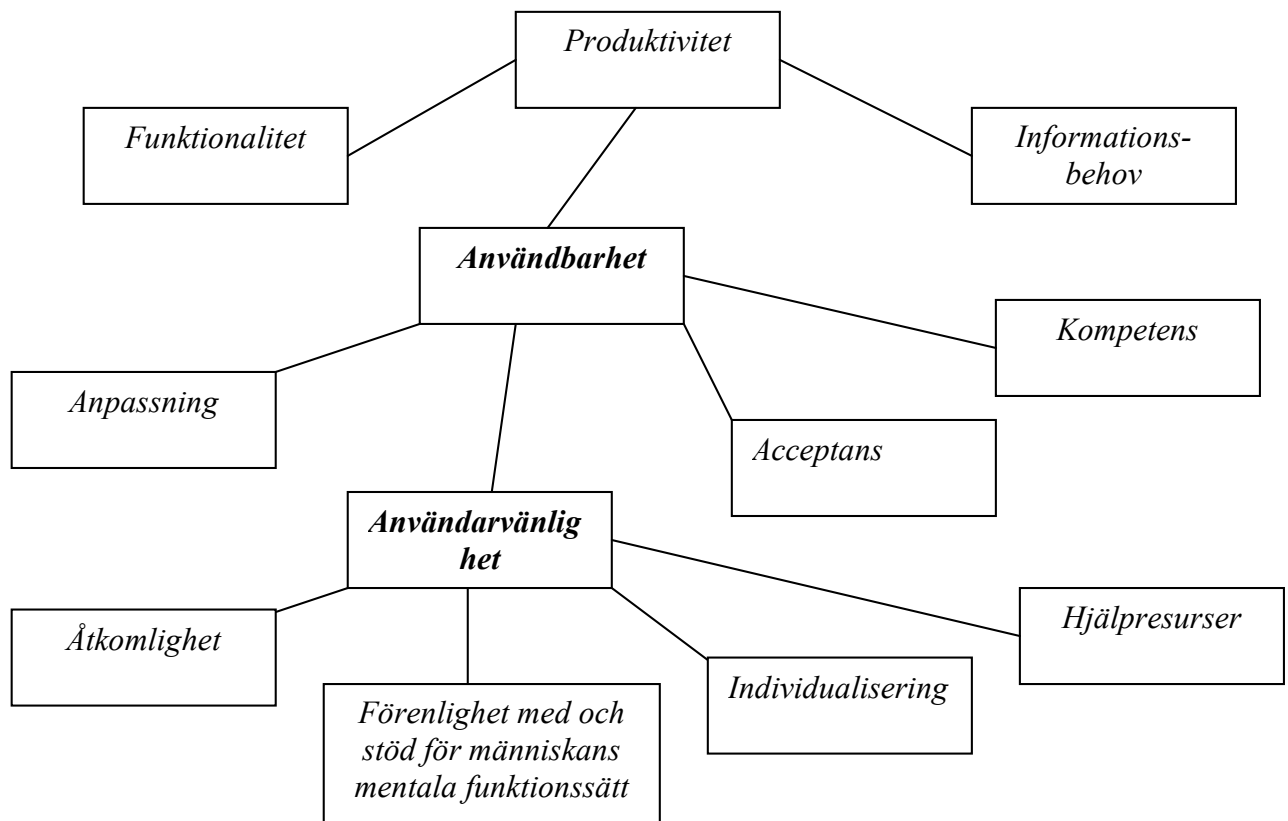
Detta kapitel innehåller den teori som rapporten och designförslaget stödjer sig på. Syftet är att läsaren skall kunna orientera sig inom området människa-datorinteraktion, få en bild av vilka teorier och modeller som har använts, samt kunna använda detta kapitel som ett stöd vid expertutvärdering och designförslag.

Kapitlet innehåller ett avsnitt, 4.1.1, om användbarhet inom människa-datorinteraktion. Nästa avsnitt, 4.2, beskriver modeller för användbarhet. Kapitel 4.3 beskriver Shneidermans pelare för god gränssnittsdesign, som följs av 4.4, hur användaren kan göras delaktig i framtagande av en god design för interaktion. Detta följs av ett avsnitt om användarens inmatning i interaktionen, 4.5, som naturligtvis måste följas av ett avsnitt om systemets presentation till användaren, 4.6.

4.1.1 Användbarhet

Det finns många författare och organisationer som givit sin syn på användbarhet och användbarhet. Några exempel är boken "Human-Computer Interaction" [Dix, Finlay, Abowd, Beale], boken "Designing the User Interface [Shneiderman]" och ISO med standard 9241.

Ett centralt och därför viktigt begrepp som används i denna rapport är användbarhet. Detta begrepp används ofta på ett vagt och opreciserat sätt. Ett snarlikt benämnt begrepp är användbarhet. På grund av denna likhet kan det finnas risk att de olika begreppen används för att beskriva samma fenomen. Det kan därför vara viktigt att göra en distinktion mellan dessa begrepp och att klargöra relationen mellan dem. Allwood [Allwood] beskriver användbarhet som en del av användbarhet enligt figur nedan:



Figur 11: Relationen mellan användbarhet och användbarhet enligt Allwood.

Detta arbete behandlar tämligen få delar av anpassning, arbetet är avgränsat till demonstratörens roll. Ingen kompetenshöjning kommer att ske förutom för den som direkt arbetar med designförslaget. De andra användarna är redan väl kunniga om programmet. En viss del av begreppet acceptans kommer att beröras men merparten av arbetet berör användarvänlighet, varför användbarhet fortsatt inte kommer att behandlas.

Ben Shneiderman

Förbättrad gränssnittsdesign innebär, enligt Ben Shneidermans, att uppnå ”The Eight Golden Rules of Interface Design” [Shneiderman]:

1. Konsekvent utformning av gränssnittets olika delar.
2. Erbjud genvägar för vana användare.
3. Erbjud informativ feedback, från system till användare.
4. Dela upp problem i delproblem.
5. Förebygg fel och skapa en enkel hantering av uppkomna fel.
6. Ångra funktion
7. En användare skall vara initiativtagaren till händelser, snarare än att svara på händelser
8. Begränsa belastning på korttidsminnet

Framgångsrik design av ett program bygger dock på att utvecklaren går längre i sitt arbete än att bocka av listor över subjektiva riktlinjer. Utvecklaren måste ha en djupare förståelse för gruppen av användare och de uppgifter som de skall utföra i sin användarroll.

En designer som vill uppnå ett användbart gränssnitt, skall ställa sig frågan: ”hur skapar jag på bästa sätt en bra interaktion mellan användaren och funktionaliteten i programmet?”. Det blir under designen av programmet ibland nödvändigt att funktionalitet läggs till, för att interaktionen skall kunna lösas på ett bra sätt.

4.2 Exekverings- och utvärderingscykeln

För att fastställa kraven som kan ställas på interaktion mellan system och användare, måste en förståelse uppnås för de steg en användare går igenom när hon interagerar med en dator.

4.2.1 Den interaktiva cykeln.

Normans modell av interaktionens beteende, den interaktiva cykeln, beskriver användarens och systemets interaktion. Det är troligen den mest inflytelserika modellen inom människa-datorinteraktion, troligen eftersom den ligger närmast vår intuitiva förståelse av interaktionen mellan den mänskliga användaren och datorn [Dix, Finlay, Abowd, Beale]:

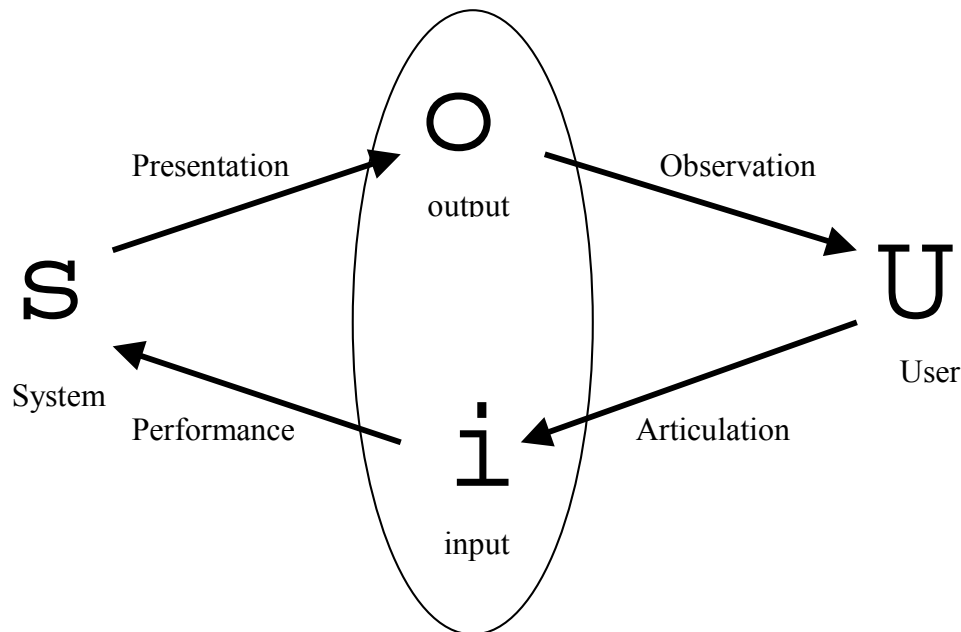
- Fastställ målet, i användarens terminologi.
- Formulera intentionen. Målet definieras i användarens terminologi, varför detta troligtvis måste översättas till domänens terminologi.
- Specificera handlingssekvensen. Intentionen skall delas upp i delmål.
- Utför handlingen.
- Mottar systemets tillstånd.
- Tolkar systemets tillstånd
- Utvärdera systemets tillstånd i förhållande till målet och intentionen

Norman anser att det finns en evalueringsklyfta mellan den fysiska presentationen av systemets tillstånd och användarens förväntningar. Om användaren snabbt kan utvärdera presentationen i termer av hennes mål, då är evalueringsklyftan liten [Dix, Finlay, Abowd, Beale]. För att hålla denna klyfta liten finns i den heuristiska utvärderingen två punkter som heter: ”synlighet av systemets status” och ”koppling mellan system och den verkliga världen”. Se *Heuristisk utvärdering*, 4.3.3.1.2. De heuristiska punkterna har inte skapats av Norman, men det finns onekligen kopplingar mellan dessa.

4.2.2 Ramverket för interaktion.

Norman anser att varje pil i figuren 11, har ett eget språk. Uppgiften att lösa, är att få övergångarna mellan dessa delar att stämma överens, så att ingen konflikt mellan dessa uppstår. Normans modell för denna interaktion visas i figur 10 [Dix, Finlay, Abowd, Beale].

Denna rapport använder delvis denna modell i upplägget för kapitlen. I beskrivning av SubTrack, expertutvärderingen och designförslaget är alla kapitel uppdelade enligt artikulation (användarkontroll) och observation (presentation). Detta ger inte en uppdelning av alla de fyra pilarna, endast de på användarens sida, men det beror på att det inte utfördes en implementation av designförslaget.



Figur 12: Exekverings- och utvärderingscykeln av Norman. Något mer tydliggjord.

4.3 Shneidermans tre grundpelare

Människa-datorinteraktion är ett område, som enligt [Shneiderman] står på tre grundpelare. Första pelaren är riktlinjer och processmodeller, andra pelaren är verktyg för gränssnittsprototyper och den tredje pelaren, expertutvärderingar och användartester.

4.3.1 Riktlinjer och designprocesser

Det finns en mängd olika riktlinjer och processmodeller. Dessa bygger på teorier om vad användarvänlighet innebär och på modeller som byggts upp för att förklara olika beteenden. Vilka av riktlinjerna och processmodellerna som väljs, är beroende på projektets natur och av den erfarenhet och kunskap som utvecklaren eller utvecklarna besitter.

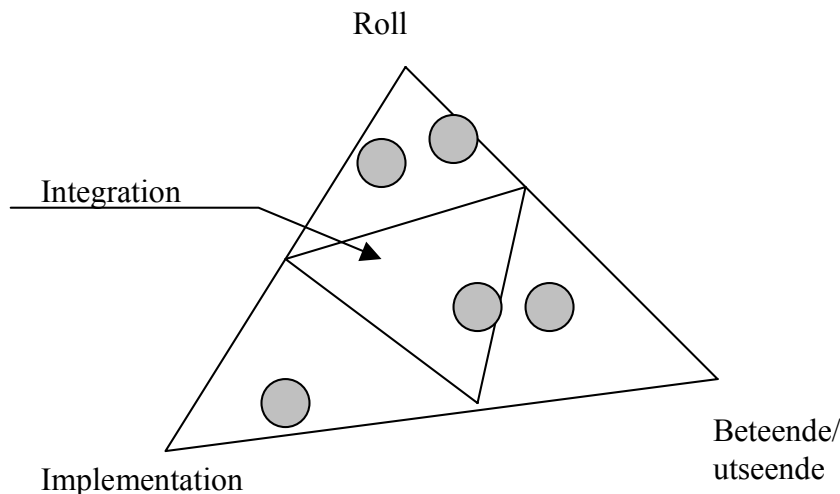
4.3.1.1 Riktlinjer

Riktlinjer utarbetas för större projekt, så att en konsekvent utformning sker av systemets olika delar. Det finns riktlinjer för de olika plattformarna, vissa är dokumenterade, andra outtalade. Microsoft har inga uppdaterade riktlinjer, enligt deras supportavdelning troligtvis beroende på att de vill behålla sin särart. Java har en komplett dokumentation för hur deras "Look and Feel" skall se ut, detta för att den skall få ett konsekvent utseende på olika plattformar. Designförslaget för SubTrack förlitar sig på en korg med vanliga program från ett antal olika plattformar och tillverkare av mjukvara, se *Konventioner*, 6.1.4.

4.3.1.2 Designprocesser, prototyper

Prototyper är endast en del i designprocessen, men för SubTrack är detta det sätt som programmet har vuxit fram. SubTrack har utvecklats, liksom flertalet andra produkter på FOI, som prototyp för att visualisera forskning av något slag. Stycket om prototyper syftar till att ge en beskrivning av de huvudgrupper som finns för prototyper. En prototyp för interaktion, kan fylla tre syften [Houd, Hill]:

- Roll – vilken roll kommer slutprodukten att spela i en användares liv, tillsammans med en beskrivning av den funktionalitet som prototypen uppvisar.



Figur 13: Houde och Hills modell för prototyper. Punkter beskriver tänkbara positioner för en prototyp inom modellen.

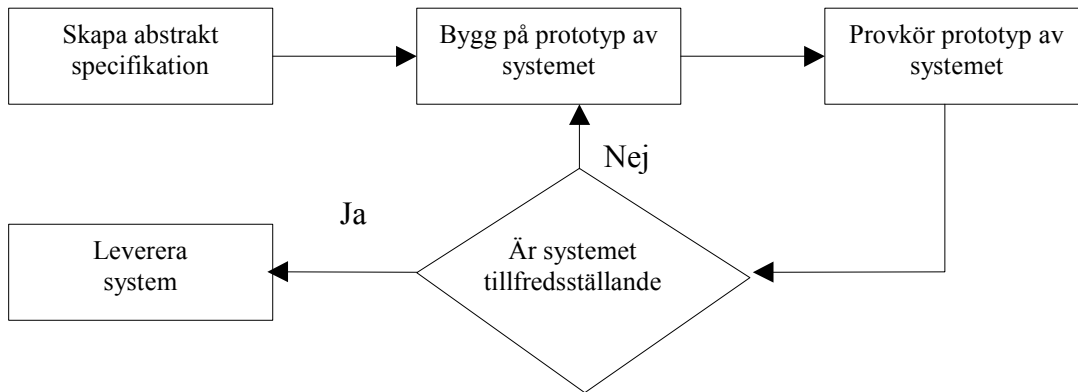
- Beteende och utseende– demonstrerar slutprodukten tänkta utseende och hur det ”känns” att interagera med den. Prototypen skall även ge en konkret uppfattning av hur slutprodukten kommer att ”se ut”.
- Implementation – undersöka och demonstrera hur en viss funktionalitet kan implementeras i praktiken.

Prototyper som representerar alla dessa dimensioner är integrerad, och är en i stort sett färdig produkt. Det är dyrt och komplicerat att föra en prototyp till integrationsstadiet. Om den ändå når detta stadium, är det föga troligt att prototypen klarar att uppfylla alla krav. Vanliga problem är prestandaproblem och ostrukturerad kod som resulterar i svårigheter att underhålla ett sådant system.

I de följande underkapitlen, finns en beskrivning av de tre huvudgrupperingarna som prototyper vanligtvis indelas i:

- Evolutionär
- ”Throw-away”
- Inkrementell

4.3.1.2.1 Evolutionär prototyputveckling



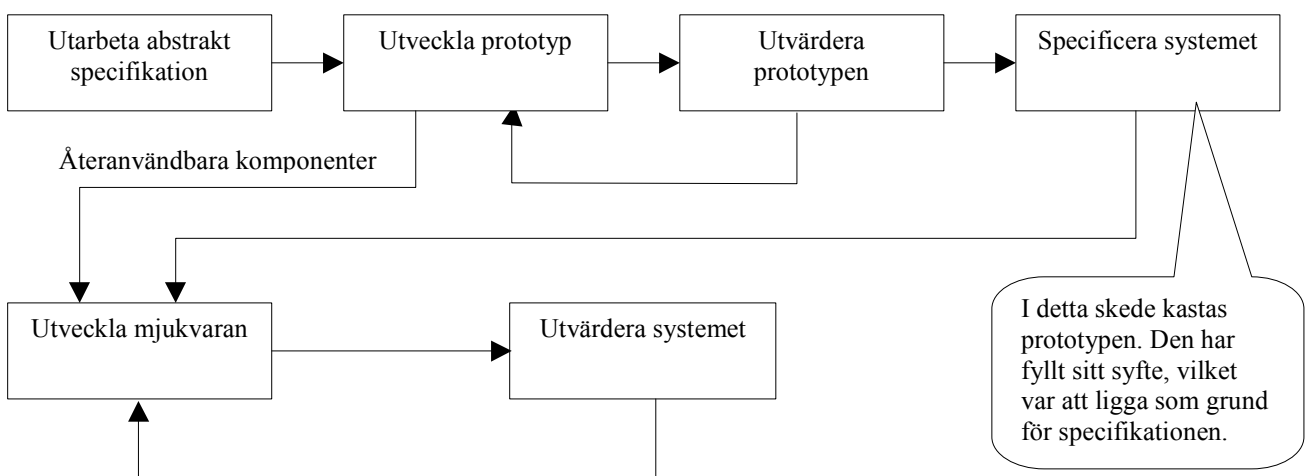
Figur 14: Evolutionär prototypmetod

Evolutionär utveckling [Sommerville], är den enda realistiska metoden att skapa en prototyp om målen är svåra eller omöjliga att definiera. Målen förändras ofta med de resultat som forskningsarbetet leder till. Det går inte att sätta upp annat än diffusa mål. Nyckeln till framgång är att använda system som erbjuder en snabb iteration mellan versionerna.

Validering är bara aktuell om det finns mål att mäta mot, så i detta fall är verifiering den enda lösningen. Verifiering innebär att utvecklingsgruppen utvärderar om prototypen kan utföra de uppgifter den är avsedd att sköta. Det finns svagheter med evolutionär utveckling:

- Det är inte möjligt att sätta upp milstolpar för att kontrollera utvecklingen i projektet, eftersom strikta mål saknas. Utvecklingen av prototypen är också så snabb att det inte är kostnadseffektivt att efter hand sätta upp formellt beskrivna mål.
- Ständig förändring av koden leder lätt till en rörig och ostrukturerad kod, som är mycket svår och kostsam att underhålla.
- Det är inte möjligt att driva större projekt på detta sätt, eftersom koordinering av flera utvecklare försvåras av att det inte finns tydliga mål uppsatta.

4.3.1.2.2 "Throw-away" prototyputveckling



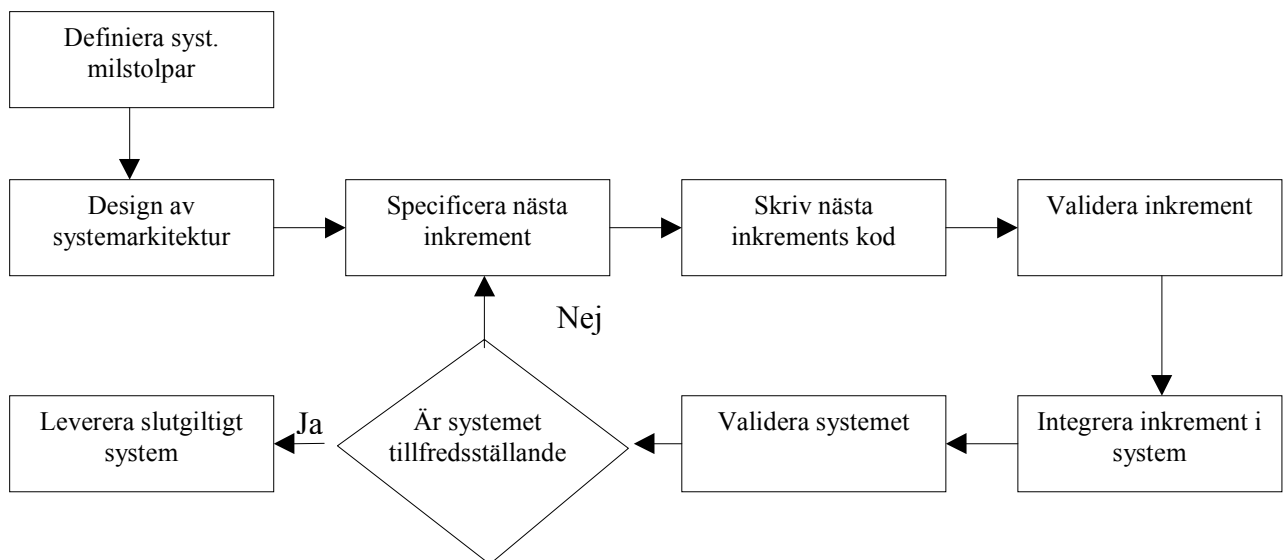
Figur 15: "Throw-away" prototypmetod.

”Throw-away” [Sommerville] kan fungera som en utökning av kravanalysen, med syfte att minska kostnaderna för den totala designprocessen. Om en ”Throw-away”-prototyp skapas, blir resultatet en tydligare kravspecifikation som kan erbjuda ledningen ytterligare information, bl.a. om riskerna i processen. Dessa komponenter kan ibland återanvändas i processen, men utvecklaren bör motstå frestelsen att förvandla denna prototyp till en slutprodukt. Det finns svagheter med denna metod:

- Egenskaper som prestanda, säkerhet, tålighet och tillförlitlighet kan ha blivit ignorerade, för att snabbt kunna utveckla prototypen.
- Under utvecklingen kan prototypen ha blivit förändrad för att uppnå en viss användares mål. Detta resulterar ofta i en rörig kod, som är komplicerad att underhålla.

4.3.1.2.3 Inkrementell prototyputveckling

En metod som kombinerar fördelarna med den evolutionära metoden med nödvändiga styrfunktioner för en utvecklingsprocess med flera deltagare [Mills]. Metoden är ”Top-Down”, de viktigaste delarna utförs först. Systemets komponenter utvecklas inkrementellt och



Figur 16 : Inkrementell prototypmetod.

levereras inom det ramverk som beskriver gränssnittet för systemets komponenter. Efter leverans ändras ingenting, varken i kod eller i ramverk, såvida inte felaktigheter uppdagas. Design av komponenter som inte har levererats, kan påverkas genom återkoppling från en användare av redan levererade komponenter.

Inkrementell utveckling av prototyper har fördelen att den lättare kan planläggas eftersom vanliga standarder för mjukvaruutveckling kan följas. Nackdelen är att arkitekturen på systemet måste bestämmas innan systemets kravspecifikation är färdig.

4.3.2 Verktyg för gränssnittsprototyper

Verktyg som på ett enkelt sätt kan skapa prototyper av ett önskvärt gränssnitt, bidrar starkt till möjligheten att formulera de krav som en användare kan tänkas ställa på ett system. Många allvarliga problem med ett gränssnitt kan undvikas om en prototyp skapas tidigt i designprocessen [Shneiderman].

Exempel på verktyg som kan användas för detta prototypskapande kan vara allt från penna och papper till ett drag- och släppverktyg för implementation, som Microsoft Visual C++ eller Borland Builder. Det är den senare typen, som Shneiderman anser vara en av de tre pelare, som god gränssnittsdesign står på.

Det fanns inte någon tillgänglig kompetens inom institutionen om verktyg för skapande av gränssnittsprototyper. Detta är naturligt, eftersom de allra flesta produkter inom FOI inte är ämnade för ovana användare.

4.3.3 Expertutvärderingar och användbarhetstest

Människa-datorinteraktion (MDI) är inte en exakt vetenskap. Det finns en uppsjö av olika metoder och tekniker som kan användas. Vissa av dessa är direkt motsägelsefulla, eftersom delade meningar råder om vad som är användarvänligt. Valet är beroende av projektets art, typen av produkt, tillgängliga kunskaper och resurser, men alla metoder syftar till att underlätta utvecklingen av ett användarvänligt gränssitt.

4.3.3.1 Expertutvärdering

Fördelar med en expertutvärdering är att den kan utföras tidigt och under hela utvecklingsarbetet. Detta resulterar i en snabb och effektiv återkoppling, till en låg kostnad. Nackdelar med metoden är att den kräver erfarenhet, och att utvecklaren aldrig helt kan sätta sig in i en användares roll eller miljö. De metoder som kommer att användas i denna rapport, är heuristisk utvärdering och kognitiv genomgång.

Alternativ till de ovan nämnda expertutvärderingarna är granskning och modellbaserad utvärdering. Svårigheterna med granskning är att det behövs dels ett utvecklat referensbibliotek med resultat av experiment och dels kunskap att använda detta bibliotek. Granskning använder den mängd experimentella och empiriska resultat som finns inom det psykologiska området och från området människa-datorinteraktion. Experimentella försök kan inte förväntas ge samma resultat inom olika domäner, varför samlade erfarenheter måste användas med omsorg. Modellbaserad utvärdering kräver förkunskap i att använda den modell enligt vilken granskning skall ske. Modellbaserad utvärdering använder modeller för att styra och underlätta arbetet. Exempel på sådana är GOMS (Goals, Operators, Methods and Selections) [Card, Moran, Newell_1] och "keystroke-level" [Card, Moran, Newell_2].

4.3.3.1.1 Kognitiv genomgång (expertutvärdering)

En expert på människa-datorinteraktion sätter sig in i en användares roll och försöker förstå hur en användare hade agerat och resonerat. Med denna metod syftar hon till att identifiera problem med användarvänlighet. Denna metod kräver: [Shneiderman]

- En detaljerad sekvensbeskrivning av händelser som systemet består av. Kod och system studeras, för att finna de tänkbara sekvenser som kan inträffa vid olika val.
- De uppgifter en användare skall utföra. Studier av kravspecifikation ger insikt i vilka mål som en användare kan tänkas ha.
- Specifikation av den sekvens av åtgärder som en användare måste utföra för att genomföra en uppgift. För att vägleda utvärderaren finns frågor som behöver besvaras. Den första är om en användare kommer att försöka nå det underförstådda mål som handlingen skulle resultera i. Utvärderaren vill veta, om en användare kommer att uppfatta att den korrekta funktionen finns tillgänglig. När användaren gjort valet, kommer hon att förstå att detta var det korrekta valet? När handlingen har utförts, kommer användaren att förstå att händelsen gav rätt resultat?
- Specifikation av vem användaren är, genom en etnografisk studie eller genom projektets målformulering.

4.3.3.1.2 Heuristisk utvärdering (expertutvärdering)

En expert utvärderar systemet efter ett antal punkter, under praktisk användning. Systemet måste delas i flera delar för att den heuristiska utvärderingen skall bli grundligt utförd. De heuristiska punkter som används i denna rapport, kommer från [Dix, Finlay, Abowd, Beale]. Studier av arbete enligt denna metod har visat att 75 % av alla fel upptäckts. De kategorier som skall värderas är:

1. *Synlighet av systemets status* – systemet skall alltid hålla användaren informerad av vad som händer, genom lämplig återkoppling inom rimlig tid.
2. *En koppling mellan system och den verkliga världen* – systemet skall tala användarens språk och använda de konventioner som finns i den vanliga världen, så att information blir synlig i en naturlig och logisk ordning.
3. *Användarkontroll och frihet* – användare väljer oftast funktioner av misstag och behöver väl markerade ”nödutgångar” för att kunna lämna oönskade tillstånd utan att behöva genomgå en lång dialog med programmet. ”Ångra” och ”gör om” är funktioner som underlättar för användaren.
4. *Konsekvens och standarder* – användare skall inte behöva fundera om olika ord, situationer eller åtgärder kan betyda samma sak. Plattformens konventioner skall följas.
5. *Förhindrande av fel* – en väl genomtänkt design som förhindrar fel är ännu bättre än bra hjälp när fel redan har uppkommit.
6. *Igenkänning snarare än memorering* – gör objekt, åtgärder och valmöjligheter synliga. Användaren skall inte behöva minnas information från en tidigare del av dialogen med systemet. Instruktioner för hur systemet skall hanteras skall alltid vara synligt eller enkelt att komma åt vid behov.
7. *Flexibilitet och effektivitet vid nyttjande* – acceleratorer som inte är synliga för amatören kan ofta öka hastigheten för den vane användaren så att systemet tillfredsställer båda användarrollerna. Tillåt användare att anpassa funktioner som ofta används.
8. *Estetisk och minimalistisk design* – dialogen med systemet skall inte innehålla information som sällan eller aldrig används. Onödigt information gör oftast de viktiga funktionerna mindre synliga.
9. *Hjälp användare att inse, diagnostisera och återhämta sig från fel* – felmeddelande skall skrivas i vanligt skriftspråk (inga koder), klart ange vad som är fel och föreslå en lösning.
10. *Hjälp och dokumentation* – all hjälp skall vara enkel att söka, fokusera på användarens uppgift, lista konkreta steg som skall göras och skall inte vara för stor.

4.4 Användardeltagande i designprocessen

Användardeltagande syftar till att förbättra arbetsmiljö och uppgifter genom att användaren deltar, inte bara som en liten del i användarexperiment, utan i hela designprocessen. Även om användardeltagande inte kan nyttjas fullt ut i designprocessen, så kan även mindre delmoment av användardeltagande uppmuntras. Det är upp till projektledaren att göra en bedömning av hur stor mängd användardeltagande som är lämpligt. Tillvägagångssättet har tre fördelar:

- Korrektare information om uppgifterna [Shneiderman]
- Användaren blir mer benägen att acceptera det slutliga systemet om hon eller han har varit delaktig i utvecklingsprocessen [Shneiderman]
- Metoden är iterativ, designen är föremål för utvärdering i varje iteration [Dix, Finlay, Abowd, Beale]

Det har dock inte bara fördelar [Shneiderman]:

- Metoden är resurskrävande och kan avsevärt förlänga implementationsfasen.
- Kan resultera i motstånd mot programmet från användare vars förslag avslagits.
- Kan tvinga designern att kompromissa för att tillmötesgå inkompetenta deltagare.

För att involvera användare i processen måste utvecklaren hitta metoder att tillgodogöra sig den kunskap som användaren besitter. De olika metoder och verktyg, som används vid en studie, förändrar alltid den kontext som användaren befinner sig i. Därför gäller det att alltid försöka göra påverkan minimal. Några exempel på tekniker att utföra en studie är frågeformulär, muntliga intervjuer, enkäter och fokusgrupper. Det är viktigt att hitta en lämplig notation för att beskriva interaktionen mellan människa och system. Metoder för detta kan vara tillståndsdigram, textnotation eller grafisk notation.

Metoder kan även användas vid användardeltagande i designarbetet, så att designern kan tillvarata de idéer om förbättringar som användaren har [Dix, Finlay, Abowd, Beale]:

- Brainstorming – ganska ostrukturerat, men eftersom all information nedtecknas utan att värderas så hittas mycket värdefullt.
- Storyboarding – klarar av att beskriva dagliga aktiviteter såväl som potentiell design och den effekt den kommer att ha.
- Workshops – kan fylla de kunskapsluckor som finns hos deltagarna samt skapa en bättre fokuserad bild av designen. Roller kan bytas för att få insikt i vilka problem som andra deltagare i designprocessen kan ställas inför.
- Penna och papper – tillåter att designlösningar går igenom utan att nyttja stora resurser. Motsättningar mellan användarens krav och de designlösningar som designern föreslagit belyses.
- PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration) är en metod som tillåter användare som deltar i designutvecklingen, att rita gränssnitt och sedan använda bitar av papper, plast och tape för att skapa enkla tidiga prototyper. Dessa scenarion spelas sedan in på video för att presenteras för arbetsledning, användare och designers.

Användartester

Representativa användare, ungefär 5 stycken, utför realistiska uppgifter och validerar sedan resultatet tillsammans med experten. Metoderna kan vara experimentella eller fältstudier. Fördelar med användarmedverkan i designen, är att man får en inblick i hur användaren arbetar och interagerar mot systemet. Användarmedverkan kan även användas på ofullständiga system och ger en bra återkoppling. Nackdelarna med systemet är att personerna kan bli påverkade av testsituationen och att användaren riskerar att bli överkritisk.

När ett kunskapsområde skall undersökas, t.ex. grafiskt gränssnitt, är det inte lämpligt att flytta försökspersoner från sin naturliga miljö. Vid en experimentell situation kan man styra sina försök hårt med följden att alla oberoende variabler kan hållas under strikt kontroll. Det som dock förloras, är den miljö vari användarna vanligtvis arbetar. Om denna faktor är viktig, kan fältstudier användas i vilka utvärderaren kommer till användaren. Idealet hade varit att utföra båda typerna men detta kan oftast inte göras av tids och kostnadsskäl.

4.5 Inmatning, användaren till systemet

För att användaren skall kunna interagera med ett datorsystem, så har många olika metoder utvecklats. Nedan är huvudgrupperna specificerade:

- Kommandostyrning – snabbt för vana användare, men inte intuitivt för ovana.
- Meny – styrs med 2D pekarföremål, som mus eller ljuspenna. Meny är en delmängd av WIMP nedan.
- Frågor och svar – systemet frågar frågor. Frågorna är oftast av typen ja eller nej.
- Frågedialog ("Query dialog") – ett mer avancerat frågebaserat språk. Använder fraser som av typen naturligt språk, men har en syntax som begränsar frågornas komplexitet.
- Formulär ("Form-fill"), ett formulär, med ifyllbara alternativ.
- "Spread sheets" – en mer avancerad än formulär. En variabelförändring, kan påverka andra variabler.
- Peka och klicka – direktmanipulation. Manipulering via klick direkt i programfönstret, som vanligen använder sig av tekniken hypertext. Ett interaktionssätt som naturligtvis är närbesläktat till WIMP, men inte kräver en mus som 2D pekare, utan kan vara pekskärm.
- WIMP – står för "Windows", "Icons", och "Menus", "Pointers" (alternativt "Mouse" och "Pull-down menu"). Detta tillsammans med knappar, verktygsfält (samlingar av knappar), paletter (menyer som kan göras fritt flytande och som tydliggör det tillstånd som programmet befinner sig i) och dialogrutor. Meny, frågedialog, formulär och direktmanipulation ingår, eller kan ingå, i ett WIMP-gränssnitt. Exempel på WIMP är Microsoft Windows för IBM kompatibla PC, MacOS för Macintosh kompatibla datorer och varianter av X-Windows för UNIX.
- Naturligt språk – inte en aktuell teknik för SubTrack, eftersom detta kräver ytterligare hård- och mjukvara, utöver skärm, tangentbord och mus.
- Tredimensionellt interface – inte en aktuell teknik för SubTrack, eftersom detta kräver ytterligare hård- och mjukvara, utöver skärm tangentbord och mus.

I det designförslag som görs för interaktionen i SubTrack, kommenteras flera av dessa interaktionsmetoder, se *Resultat*, 8.

4.6 Presentation, systemet till användaren

När systemet skall presentera information är det viktigt att detta anpassas för att passa det syfte som utvecklaren avser visualisera. Det finns olika sätt att låta systemet presentera informationen, t.ex. i form av grafik, för att skapa en överskådlighet över händelserna. Detta är bra för en demonstration, men för analytikern kan loggfilerna vara viktigare. Olika parameteruppsättningar kan t.ex. behöva jämföras tillsammans med någon av beräkningsalgoritmerna.

Information kan också presenteras numeriskt med loggfiler, som exakt kan visa händelserna vid varje tidssteg och när något väsentligt inträffade. Loggfilerna kan bestå av siffror, text, tabeller eller eventuellt diagram. Demonstratören har inget behov att visa upp loggfilen, det är fullt tillräckligt att hon berättar att möjlighet att tolka systemet finns. Det finns i den visuella presentationen några grupperingar som skall övervägas. Strävan är enligt de heuristiska punkterna att göra en estetisk och minimalistisk design. En indelning i tre kategorier [Linköpings Universitet] följer nedan:

- Layoutprinciper – hierarkiska, rutnät, luft, balans och konsistens.
- Typografi och text – så få typsnitt som möjligt, begränsat antal storlekar, använd storlek för att skapa hierarki och läsordning – orientera läsaren i texten, svart text på vit bakgrund, använd inte VERSALER, använd talande rubriker.
- Färg – så få färger som möjligt, använd inte enbart färgkodning, komplettera med t.ex. position och form, blått och rött skär sig, färgblindhet (röd-grön).

5 Designprocess, för ett användarvänligare SubTrack

Detta kapitel innehåller en beskrivning av det sätt enligt vilket designförslaget för ett användarvänligare SubTrack utvecklats. Syftet är att läsaren skall ges en bild av den planering som ursprungligen gjordes tillsammans med en beskrivning av hur designprocessen slutligen genomfördes.

Kapitlet innehåller en beskrivning av hur designprocessen var planerad, 5.1. Under arbetets gång, dök problem upp, 5.2, varför en förändring av designprocessen skedde, 5.3.

5.1 Ursprunglig planering

När examensarbetet startade var målet att en ny version av SubTrack skulle utarbetas och implementeras. Målet har förändrats under arbetets gång och slutligen lett till att arbetet enbart blivit en rapport, dock mer omfattande än om implementation hade kunnat ske.

Den designprocessmodell som låg till grund för examensarbetet, har tagits ur en kurs på KTH som heter människa-datorinteraktion, kurskod 2D1406. En designmodell är en planering av hur arbetet med produkten skall ske. Exempel på andra designmodeller är RUP, vattenfallsmodellen, m.fl. Dessa modeller är generella för att kunna passa en stor mängd projekt, från start till en produkt som skall levereras till kund. Detta är inte riktigt tillämpligt för SubTrack, då det är en prototyp som inte är avsedd för en slutanvändning med överlämnande, underhålls- och dokumentationsfas.

Det första momentet var att utföra studier av strukturen på FOI och av vilka designmodeller som användes, formella eller informella. Detta skulle kunna vägleda valet av modell för utvecklingen av SubTrack. Efter det planerades en expertutvärdering, följd av användartester av fyra till fem personer. De två momenten sammanställs, varvid de brister som uppdragets resulterar i ett designförslag, som kan realiseras och sedan testas av nya användare. Ett mätbart resultat visar om problemen har avhjälpats. Designförslaget skall ses som en hypotes och det är därför inte nödvändigt att, mer än skissmässigt, beskriva och argumentera för de föreslagna förändringarna.

5.2 Problem

Efter visst arbete med programmet, framstod möjligheten att få några bra resultat av användartester som tämligen små genom att använda det befintliga SubTrack. Detta eftersom programmet helt enkelt var för långt ifrån en analyseringsbar produkt. Vid diskussioner med en handledare på KTH Nada, rekommenderades att endast expertutvärdering skulle ske före implementation. För att skapa stöd för expertutvärderingen, gjordes dock två användartester och intervjuer, då användare fick föreslå förändringar av SubTrack. Se *Användardeltagande i design*, 7.2, 7.3.

Implementationen av programmet var tänkt att ske i programmeringsspråket Eiffel, ett objektorienterat språk. Implementationen kunde inte genomföras pga. tidsbrist, efter problem med installation av Eiffel Studio. Vidare uppstod problem under konverteringen av det befintliga SubTrack från X-Windows under Unix till MS Windows på PC.

5.3 Slutlig designmodell

Eftersom implementation inte kunde genomföras, förändrades målet med examensarbetet till att enbart bestå av en rapport. Att skapa en prototyp som kan simulera SubTrack ansågs inte lämpligt, eftersom det varken fanns lämplig mjukvara eller kunskap om någon sådan inom institutionen. Rapporten itererades istället i designprocessen, för att kunna öka tyngden i expertutvärderingsfasen, gjordes en kognitiv genomgång som adderades till expertutvärder-

ingen. Experten skall i en kognitiv genomgång försöka sätta sig in i användarens situation, genom att tänka igenom varje steg som en användare kan vilja utföra.

Designmodellen kom att se ut enligt nedan (viss iteration skedde även under arbetets gång):

1. Designmodell – planering av de steg som utvecklingen av SubTrack skall följa.
2. Studier – organisationen och de designprocesser som finns inom FOI. Främst skedde intervjuer på institutionen Data- och informationsfusion, men intervjuer skedde även av anställda på institutionen för Systemmodellering.
3. Modeller – det befintliga SubTrack studeras och dokumenteras, för att skapa en övergripande förståelse för vad systemet kan utföra, samt hur det är uppbyggt.
4. Designmodell – ökad detaljering, efter att en ökad förståelse av SubTrack har uppnåtts.
5. Expertutvärdering – de heuristiska punkterna genomgås för att identifiera svagheter i gränssnittet, först på ett övergripande plan och sedan i mer detaljerad form, t.ex. studier av hur menysystemet fungerar.
6. Uppgiftsanalys – en erfaren demonstratör visar hur en demonstration bör utföras.
7. Användartester – användare får utföra några uppgifter för att identifiera svårigheter och felaktigheter med användargränssnittet. Intervjuer utförs samtidigt för att användarna skall kunna komma med förslag på förändringar i systemet.
8. Implementation – påbörjades men kunde inte utföras, varför en förändring av designmodellen var nödvändig. De nästkommande stegen skulle, enligt den första planeringen, ha varit användartest 2, samt utvärdering av användartest 2.
9. Utökade modeller över systemet – för att kunna göra en kognitiv genomgång, krävs en detaljerad bild av systemet.
10. Expertutvärdering, kognitiv genomgång – en expert försöker förstå systemet utifrån att följa de sekvenser, som experten tror en användare kommer att utföra.
11. Rapportskrivande.

6 Expertutvärdering av SubTrack

Detta kapitel innehåller en expertutvärdering. Syftet är att analysen skall uppdaga brister i interaktionen mellan människa och system. Verktygen att uppdaga dessa brister är expertutvärdering, kognitiv genomgång och användardeltagande. Brister som identifierats finns under varje enskilt kapitel.

Detta kapitel inleds med en orientering av faktorer som påverkade expertutvärderingen. Första underkapitlet behandlar SubTracks funktionalitet, 6.2, som i stort sett behålls oförändrad. Följande kapitel, 6.3, behandlar användarens inmatningar till systemet, som följs av kapitel, 6.4, som handlar om systemets presentation till användaren. Avslutningsvis kommer ett kapitel om startsekvensen, 6.5, som beskriver den startsekvens som oftast sker. Rapporten har anpassats, så att en konsekvent styckesindelning finns till kapitlen 2 och 8.

6.1.1 Syfte och mål med expertutvärderingen

Syftet med en expertutvärdering, är att en expert inom människa-datorinteraktion analyserar ett system för att finna brister i användarvänligheten. Den första metod som använts för SubTrack, är expertutvärdering enligt heuristiska punkter, 4.3.3.1.2. Den andra metoden är kognitiv genomgång, 4.3.3.1.1. Den andra metoden användes för att förfina den första expertutvärderingen, eftersom implementation och användartester inte kunde genomföras.

En iteration har skett, därför kan det vara svårt att spåra ursprunget till en identifierad brist i användargränssnittet. Ett förslag som finns i expertutvärderingen, kan ha sitt ursprung i användartesterna, men när itereringen skett så har detta förslag förts till expertutvärderingen. Under rapportskrivningen eftersträvades spårbarhet från beskrivning av SubTrack till expertutvärdering och slutligen i designförslaget.

6.1.2 Roller och aktörer

Av de roller som SubTrack kan ikläda sig för att passa olika aktörers behov, är det endast demonstratörens behov som skall beaktas.

6.1.3 Användningsfall

Användningsfallen från figurerna i 2.2 och 2.1.3 används för att experten skall få en bild av systemet. Modellen förblir oförändrad, dock sker senare en förändring i användningsfallsdiagrammet under framtagningen av designförslaget, för att visualisera de föreslagna förändringarna.

6.1.4 Konventioner

De två plattformar som är aktuella för SubTrack är MS Windows på PC eller X-windows. Från dessa plattformar valdes fyra stycken program eller gränssnittsriktlinjer som ska fungera som en måttstock, för vad som anses som konventioner. De program som ligger i denna korg:

- Adobe Acrobat Reader – ett program, som återfinns på fler plattformar än vad som SubTrack avses användas på. Detta program varierar något i utformning, beroende på plattform, men huvudsakligen är den konsekvent över flertalet.
- Microsoft Word – ett program som finns till MS Windows, men även till Macintosh operativsystem.
- Frame Maker – finns både till X-Windows och MS Windows.
- Touch and Feel, Java – inte ett program, utan en samling av riktlinjer hur Javaprogram skall se ut. Java är, som säkert är bekant, plattformsoberoende. Detta får till följd att Java-program i X-Windows, använder den fönsterhantering som användaren nyttjar. Ramen till ett program kan därför skilja, medan allt annat är identiskt.

Det finns riktlinjer för design av program under olika operativsystem, men de är inte alltid lika tydligt uttalade som i MS Windows fall. I fallet X-Windows blir det lite mer komplicerat, eftersom det finns en hel uppsjö fönsterhanteringssystem ges utvecklare en stor frihet att välja ett eget utseende. Exempel på sådana är CDE (Common Desktop Environment), KDE och Gnome. Dessa fönsterhanteringssystem sköter de grundläggande funktionerna mot operativsystemet, de kapslar in programmen.

Microsofts riktlinjer för gränssnittsdesign genomgår en ständig förändring. Vid kontakt med deras supportavdelning klargjorde de att det inte fanns några heltäckande riktlinjer för gränssnittsdesign. Det var enligt dem inte önskvärt att alla program till MS Windows såg lika ut. De önskade att andra tillverkare skapade sitt eget gränssnitt. En slutsats som kan dras av detta, är att Microsoft önskar att skapa en egen form som kan ge dem en egenart.

När konflikter har uppdragats mellan de olika plattformarnas konventioner, exempelvis placeringen av huvudmenyn hjälp, har valet av konvention fallit på Microsofts. Detta dels för att det är det utseende som flest användare, och även åskådare av demonstrationen, har kommit i kontakt med. En ytterligare faktor har varit att det bland X-Windows riktlinjer inte har kunnat identifieras lika konsekventa konventioner.

6.2 SubTracks funktionalitet

SubTrack är en prototyp för att påvisa forskning, och detta arbete berör endast användarvänligheten av denna prototyp. Expertutvärderingen kommer alltså inte att analysera den grundläggande funktionaliteten hos programmet.

6.2.1 Simulering

Simuleringssekvensen har vissa brister. Viktigast är en funktion att kunna backa i programmet. Detta är en form av ångra-funktion så att användaren kan ta sig ur ett oönskat tillstånd. Detta skulle öka användarens frihet. Användare A, som intervjuades under användartesterna i kapitel 7, ansåg dock att simuleringssekvensen skulle ses som ett spel, och i spel kan man inte ångra sig. Demonstratörens roll är dock att förevisa ett program, görs ett fel skall detta lätt kunna ånras och demonstrationen skall kunna fortsätta.

Fler brister i SubTrack är att programmet måste startas om för att en ny simulering skall kunna påbörjas. Detta ger inte användaren känsla av kontroll och frihet.

Det tar lång tid att använda körkontrollerna och det borde finnas genvägar för att erbjuda användaren effektivitet vid nyttjandet.

Automatisk bojutläggning fungerar dåligt, sekvensen att räkna ut var bojarna skall placeras tar lång tid. Detta är ett problem som beror på den matematiska Partikelfilteralgoritmen i programmet. Detta ger konsekvenser för användarvänligheten, men är inget problem som kan lösas i denna rapport.

6.2.2 Beräkningsfunktion

Som tidigare påpekats, finns det begränsningar i den matematiska funktionen Partikelfilter, men dessa kommer inte att behandlas i denna rapport.

6.2.3 Beräkningspresentation

Den grafiska komponenten bojars räckvidd ger ett grötigt utseende. För mycket information presenteras helt enkelt, det är inte en estetisk och minimalistisk design. Om istället räckvidden av ljudet från målet visas, ökas prestanda på ritfunktionen. En mer minimalistisk design erhålls, därigenom ökar också synligheten av systemets status.

När Partikelfiltret används som beräkningsmetod, finns inget förvalt moln som visar målets uppskattade position. När Kalmanfilter används finns en mätellips förvald, detta borde även gälla när Partikelfilter används. Konsekvent funktionalitet genom programmet är viktigt.

6.2.4 Initieringsparametrar

Dessa parametrar kan redigeras i en fil av användaren. Detta kräver kunskap och försiktighet av användaren, och en demonstratör är inte alltid välbekant med systemet. För demonstratören kan det dock vara intressant att kunna visa den påverkan vissa av dessa parametrar har för systemet. En kontrollerad form för redigering skulle erbjuda användaren användarkontroll och frihet, flexibilitet och effektivitet, samtidigt som det skulle förhindra fel.

6.2.5 Tillstånd

De tillstånd som finns i den nuvarande versionen av SubTrack, är till större delen tillfredsställande. Vissa brister i körkontrollerna som påverkar tillståndet för simuleringssekvensen har dock identifierats under expertutvärderingen och användartesterna.

Stopp i körkontrollen fyller endast ett behov som inte det frysta tillståndet gör. Musen inträder, vid stopptillstånd för simuleringen, i ett tillstånd för bojutläggning då användaren tillåts att placera ut 10 stycken bojar på inmatad position och vid ett angivet tidssteg. Denna funktion är, enligt utvecklaren själv, mindre lämplig. Funktionen är inte intuitiv och användaren når inte funktionaliteten genom lämpliga kommandon. Användaren har genom en slump blivit medveten om denna funktion, det ges ingen förklaring varför detta sker.

Dialogfönstret för planerad utplacering av bojar bör nås genom ett kommando i menyn och tillståndsförändringen hos musen är inte önskvärd. Denna förändring skulle öka användarkontroll och frihet, förhindra fel, samt göra att systemets status skulle bli mer konsekvent.

Vi återgår till tillståndet stopp. Med ovanstående förändring skulle det inte finnas anledning att ha mer än ett av tillstånden fryst eller stopp. Då det frysta tillståndet tillåter experimentering med bojkonfigurationen, är det frysta tillståndet det naturliga valet då användarens kontroll och frihet ökas. Kopplingen till den verkliga världen blir dock inte bra om stopp inte finns för en simulering, namnet kanske skall behållas som stopp och funktionaliteten som fryst.

När simuleringen har nått sitt slut och programmet inträder sitt sluttillstånd, går bojar inte längre att manipulera. Att inträda det tidigare frysta tillståndet verkar lämpligare och ökar användarens kontroll och frihet.

Vid inmatning till inmatningsfält krävs ett returslag för att variabeln i programmet skall ändras. Detta är en källa till fel, och bör åtgärdas genom att när inmatningsfältet förlorar fokus registreras inmatningen.

6.3 Användarkontroll

De nuvarande metoderna med vilka användaren kontrollerar programmet är bra, men ibland är de lite otympliga. Snabbare metoder för att styra programmet behövs, framför allt gäller detta körkontroller för simuleringssekvensen, samt zoom och panoreringsfunktionerna. Någon form av direktmanipulation är nödvändig för att öka effektiviteten.

6.3.1 Menyner

- Indelningen av funktionerna, valet av namn och ordningen i menyraden är inte tillfredsställande. Dålig koppling till konsekvens och standarder i andra datorprogram.
- Inte samma namn i menyval som i dialogruta. Inte konsekvent, med andra ord. Ett exempel är menyvalet "Zoom/Pan" som heter "Modes for zoom and pan handling".

Game	Menynamnen som styr spelet (run until, freeze osv.) ger inte tydliga ledtrådar till vad de kan tänkas utföra. Liknelsen till en videobandspelare kan vara lättare att relatera till. Koppling till den verkliga världen är viktig, för att skapa en metafor som användaren kan relatera till.
Load	Skall finnas tillgängligt, även om parametrar för <i>val av bana</i> satts vid start från kommandoprompt. Bör inte ligga under "Game" funktionen men under en blivande "file" funktion.
Run	
Run Until	Inmatning skall inte kräva ett returslag, åtgärd skulle förhindra fel. Se <i>Tillstånd, 6.2.5</i> . BUG Tiden på menyraden och "Run until next" stämmer inte. "Run until next" är svår att förstå vad den innebär. "Stop now" borde vara "Stop" Att "Step 4 seconds" finns i dialogrutan "Run until next stop time" är inte självklart.
Step time	"Step time" är samma sak som "Step 4 seconds" ovan. Dålig konsekvens.
Stop	Se <i>Tillstånd, 6.2.5</i> .
Freeze	Bättre med "still"? Se <i>Tillstånd, 6.2.5</i> .
Screen Copy	Fungera ej och är felplacerat, placering under "view" hade varit bättre. Eftersom funktionen inte fungerar, behandlas den inte i rapporten.
Exit	Bra, korrekt placerad, om den vänstra huvudmenyn döps till "File".
View	Under detta menynamn bör det finnas objekt som förändrar visningen, i eller på huvudfönstret. Detta för att designen skall vara konsekvent, och använda samma standarder som vanligt förekommande programvaror. Exempel på objekt som visualiseras, kan vara statusfältet, eller målet som ritas ut i det grafiska fönstret.
Tracks	Dåligt namn, dialogrutan innefattar alla grafiska objekt. Användaren skall förstå vad funktionen gör, kanske någon metafor. Gruppera funktioner som hör ihop, det ökar synlighet av programmets status. För långa namn, de skall vara minimalistiska. Färgval relaterar till färgen de skrivs ut med, det är dock svårt att läsa, rött på blå botten är inte bra, fel kan uppkomma. Ett ytterligare grafiskt objekt läggs till, ljudets räckvidd från målet, "Sound Range". Se <i>Beräkningspresentation, 6.2.3</i> .

Grafikkomponenter som ingår i dialogfönstret, Se *Ritfunktion*, 6.4.2.:

Path
Target
Measurement ellipse history
Prediction ellipse history
Buoy range
Estimated buoy position
Buoy position disturbance
Buoy-target connection line
Particle prediction cloud
Particle estimate cloud, se *Beräkningspresentation*, 6.2.3.

Files	Borde finnas under huvudmenyn hjälp, eftersom det är information om vilka filer som programmet använder. Layouten bör ses över, för brett dialogfönster och typsnittet borde minskas, minimalistisk och estetisk design. Kort beskrivning av vad dialogrutan gör, en hjälp bör finnas direkt i dialogrutan.
Buoys	Bör samlas med andra variabler, som dynamiskt förändras under körning. Detta skulle öka synligheten av status. Kombination av antal förbrukade bojar och startantal är snyggare. Den viktiga informationen skall fokuseras.
Path Edit Zoom/Pan	Berörs ej, ligger utanför rapportens område. Dela upp grupper av “modes” och “actions” med samling i grupper, snarare än att gruppera med färger. Snabbknappar för nedanstående funktioner (förutom normal mode) skulle förenkla. Kanske ett utseende på ikonerna i form av ett förstoringsglas. Se <i>Direktmanipulation</i> , 6.3.5. Det är genomgående för långa namn. De saknar dessutom en symbol, så att användaren associerar till något. En mer minimalistisk design av fönstret önskvärt. De olika funktionerna i zoom och panoreringsfönstret:

Normalt mode (tillstånd) – efter att någon av tillstånden nedan har inträtts, måste mustillståndet återställas till normaltillstånd. Detta är svårt att minnas, vilket användartesterna tydligt visade. Synligheten av status måste ökas, alternativt måste en annan lösning finnas, som kan förhindra fel.

Pan via click on new center (tillstånd) – vanligt förekommande i andra program, dock är en hand som drar grafikytan till önskad position vanligare.

Zoom in by step and click on new center (tillstånd) – bra funktion, följer plattformstandard.

Zoom out by step and via click on new center (tillstånd) – bra funktion, följer plattformstandard.

Show the whole path (exekverande kommando) – bra funktion, dock svårt att hitta liknande zoomfunktioner i andra program.

Show path and all buoys (exekverande kommando) – bra funktion, dock svårt att hitta liknande zoomfunktioner i andra program.

Pan to target (exekverande kommando) – specialfunktion för SubTrack, bra.

Refresh	Bra.
Deploy	Det mönster (geometrisk figur) som väljs för bojutläggningen behöver ingen egen huvudmeny. Vidare bör val markeras med en bock, radioknapp eller liknande, för att öka synlighet av systemets status.
Single till Random	Dessa bojutläggningsmönster förändras inte, de används sällan enligt intervju med användare A.
From file	Förändras inte. Denna fil kräver erfarenhet för att kunna skapas, och det erbjuds ingen felkontroll. En användare som önskar nyttja denna funktion, exempelvis för att slippa den tunga beräkningen som Partikel-filtrets automatiska bojutläggning resulterar i förväntas rådfråga utvecklare eller annan användare med kunskap om hur denna fil kan skapas och hur olika parametrar manipuleras.
Help	Borde placeras längst till höger. Innehållet borde innehålla både hjälp för att köra en simulering och information om programmets komponenter och parametrar. Den hjälp som finns nedan, kan grupperas så att färre undermenyer behöver finnas. En del av hjälpen nedan har en stark koppling till ett specifikt dialogfönster, där den bör finnas som en hjälpknapp.
About	Skall ligga längst ned av undermenyerna, för att följa konventionerna vanliga fönsterprogram.
Batch file	Eftersom funktionen inte fungerar, behandlas den inte i rapporten.
Manual	Ger endast referenser till manualer, borde kunna ge en kortfattad beskrivning av programmet.
Mouse	Det är inte musens funktionalitet som skall beskrivas, utan användarens direktmanipulation. Fokus skall vara på användarens uppgifter.
Score	Bör återfinnas under en allmän hjälp, i programmets statusinformation.
Options	Onödig huvudmeny, som endast innehåller en undermeny och det är inte minimalistisk design. Detta är ett av de moment, som ofta ändras vid start av en ny simulering. Se <i>Startsekvens</i> , 6.5.

6.3.2 Dialogfönster

Dialogfönstren har lagts under rubriken användarkontroll, trots att inte alla dialogfönster är interaktiva. Vissa fönster har innehåller bara information, vanligtvis som text, om möjligheten att stänga textfönstret inte räknas till interaktion. Trots skillnaden i interaktion mellan vissa fönster, finns inom gruppen dialogfönster mer som förenar än som skiljer.

Dialogfönstrens design skall vara konsekvent, så att användaren känner igen sig. Det är också viktigt att designen hålls minimalistisk och estetisk, för att användaren skall trivas.

- Olika utseende på dialogfönster, bör göras konsekvent.
- Möjlighet att stänga alla dialogrutor måste skapas, för att användaren skall få kontroll.
- Raderna av text i dialogfönster blir ofta lång och omständlig att läsa. En spaltuppdelning eller en rullista i ett mindre fönster hade varit att föredra. Konsekvent layout måste eftersträvas.
- Grupperingen av funktionerna, valet av namn och ordningen i menyraden är inte tillfredsställande. Detta får följdverkningar för dialogfönstren, genom att information och verktyg inte kan organiseras på ett sätt som följer konventioner. Kopplingen blir dålig till verkligheten, och även till den virtuella verklighet som grafikfönstret representerar.
- Det är inte alltid samma namn i menyval som i dialogruta. Inkonsekvent.

6.3.3 Körkontroller

Beskrivs under "Game", i stycket Meny, 6.3.1. Körkontrollerna blir i designförslaget ganska omfattande, varför de har fått ett eget stycke.

6.3.4 Zoom och panoreringsfunktioner

Kommentarer om funktionerna i "Zoom/Pan" finns i stycket Meny, 6.3.1. Zoom- och panoreringsfunktionerna blir i designförslaget ganska omfattande, varför de har fått ett eget stycke. Brister som inte hör till en befintlig funktion finns kommenterade nedan:

- Zoomning genom att markera den storlek som skall zoomas genom att markera en ruta. Detta kan öka användarens känsla av kontroll.
- Att öka eller minska zoomfaktorn, utan att förändra tillståndet för musen, är önskvärt då det skulle effektivisera användandet.
- Annan pekare än de "fyra pilarna", förslagsvis ett förstoringsglas när zoomning är aktuellt. Detta kan öka synligheten av systemets status.
- En "pekande hand" som muspekare lämpligt när förflyttning av fokus används. Detta skulle också öka synligheten av systemets status.
- När pekare är i zoomtillstånd, kan inga dialogrutor visas eftersom huvudfönstret alltid tar tillbaka fokus. Detta kan vara en svaghet i grafikbiblioteket som SubTrack använder, Eiffel Vision.

6.3.5 Direktmanipulation

Det sätt som användare placerar och redigerar bojar på är bra. Ytterligare funktioner i programmet kan förbättras genom att funktioner nås snabbare, med färre kommandon. Möjlighet att nå funktionalitet via direktmanipulation som är väsentlig för simuleringskontroll eller zoomning vore värdefullt. Eventuellt skulle popupmenyer kunna aktiveras i huvudfönstret eller i statusfältet.

6.4 Presentation

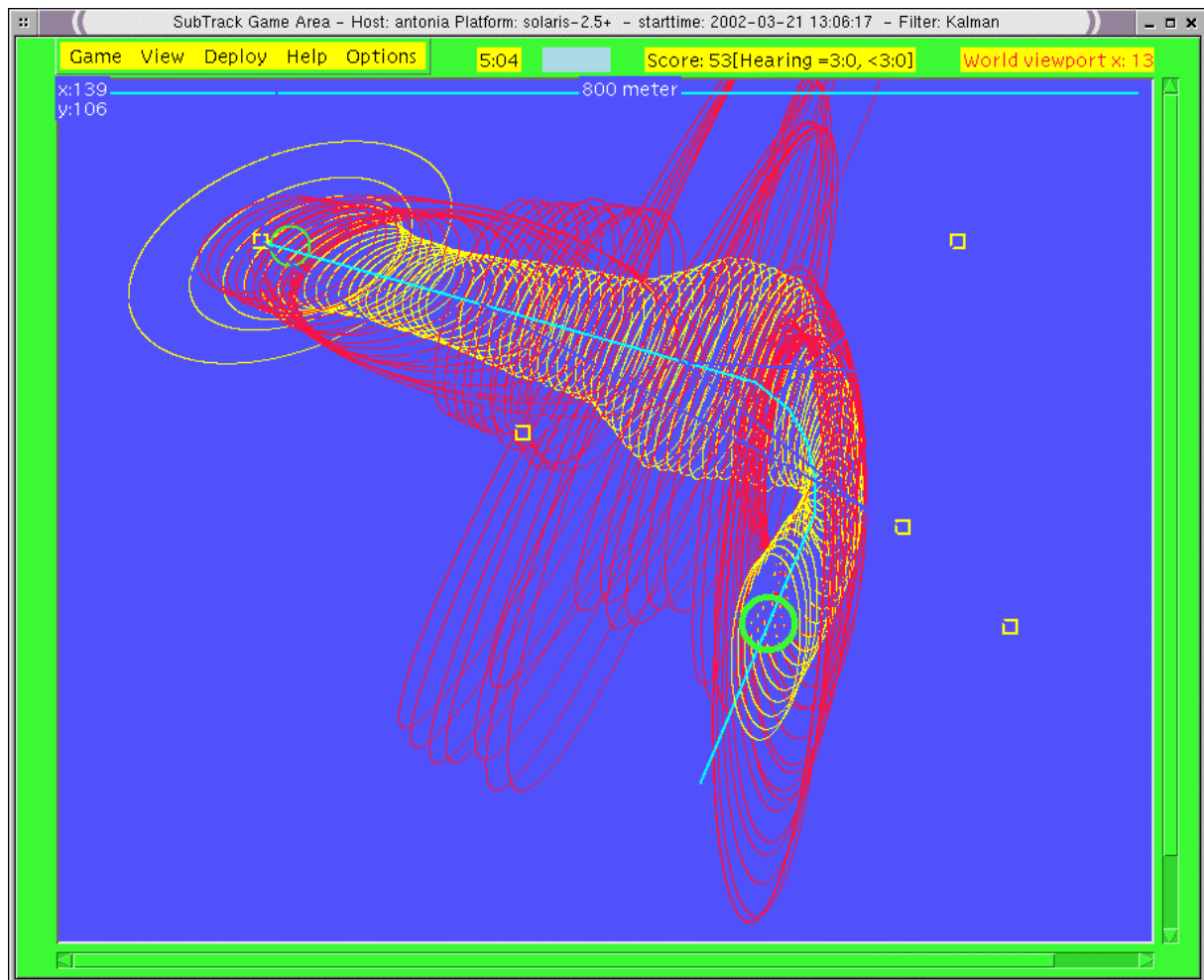
När zoomning har gjorts, läggs inte alltid bojarna ut som avsett. De dyker upp på andra ställen än där muspekaren befunnit sig. Felet kräver mer ingående studier av när och hur detta inträffar, felet lämnas tills vidare utan åtgärd.

När en dialogruta flyttas skall det som funnits bakom ritas om. Detta är en brist, som troligtvis beror på ett outvecklat grafikbibliotek, Eiffel Vision. Det finns idag ett bättre bibliotek, Vision 2. Ingen implementation kommer att ske i designförslaget, varför detta lämnas till senare implementation.

6.4.1 Huvudfönstret

Grälla färger och en bred ram är varken estetisk eller minimalistisk design. De rullister som finns i ramen fungerar inte tillfredsställande. Den yta som finns tillgänglig för rullning är endast den totala skärmstorleken. När bojar placeras utanför ytan på skärmen, borde dessa bojar också bli tillgängliga. Rullytan borde vara: aktuell storlek på banan, och alla bojar.

6.4.2 Ritfunktion



Figur 17: Körning av simulering med Kalmanfilter.

- När inzoomning har skett, ritar den grafiska ritfunktionen upp hela det tidigare området fast mer detaljerat. Detta trots att endast en mindre del av det är synligt. Prestandaförlusten är stor, användaren tvingas sitta och vänta på att ritfunktionen skall rita upp något som ändå inte syns.
- Ordningsföljden av de objekt som ritas om bör förbättras, det skulle öka synligheten av systemets status.
- De grafiska objekt som hör till bojarna ritas om vid varje tidssteg. Detta stjäl prestanda av ritfunktionen.
- Grafikkomponenten för målet "Target" förstör annan grafik, därför bör dess utformning ifrågasättas. Ett hårkors med ring kan ritas med tunnare linjer och samtidigt ge en korrektare position för målet. Designen hade blivit mer minimalistisk.
- "Buoy-connection line" förstör annan grafik, eftersom det ritas upp i varje tidssteg. Detta får konsekvensen att den verkar ligga ovanför ellipserna. Ellipserna är viktigare än en förbindelselinje mellan boja och den position som målet befann sig vid tidpunkten.
- Användaren är intresserad av att veta hur långt målet hörs och kanske inte alltid behöver se vilken räckvidd bojarna har. Detta skulle öka användarens möjlighet att se systemets status.
- Uppritning av historieellipser bör göras på ett tydligare sätt, för att designen skall bli mer minimalistisk.

- Vid stor utzoomning grötar historieellipserna ihop sig. Följden blir att det är svårt att skilja ellipserna åt. Om t.ex. var 4:e ellips ritas, minimeras resursåtgången för uppritande och synligheten ökar.

6.4.3 Statusinformation

Statusfältet ligger till höger om menyraden, och är en samling av data som användaren kan behöva studera under körningen. Statusfältet visar information som användaren kan känna igen och därför inte behöver hålla i minnet. Först från vänster ligger tiden och simuleringssekvensens tillstånd, viktiga funktioner som är bra placerade i användarens blickfång. ”Score” är några svårtolkade värden på den kvalitet som bojkonfigurationen har erbjudit beräkningsfunktionen under den tid som simuleringen pågått. Finns endast för Kalmanfiltret, saknar mening för Partikelfiltermetoden. Slutligen ett mått på zoomningsfaktorn, som ingen utom en van användare kan tolka.

Det finns statusinformation i en dialogruta som heter ”Buoys”, och som återfinns under huvudmenyn ”View”. En dålig placering då detta är information som borde återfinnas som hjälp, information eller kanske i statusfältet. Dialogfönstret uppdateras inte automatiskt, det måste ske genom en knapptryckning.

6.4.4 Hjälp och dokumentation

Hjälpen är väldigt splittrad, den återfinns i hela menysystemet. Den har inte fokus på användarens uppgift.

6.5 Startsekvens

Vid start väljer användaren inställningar för att köra en simulering, ungefär fem val görs, som varierar beroende på vad demonstratören vill visa upp av SubTracks funktionalitet. Dessa val görs i den befintliga versionen i fem delmoment. Dessa moment upprepas gång på gång innan varje körning startas. Ingen återkoppling sker till användaren av vilka moment som gjorts, och vad som då valdes.

Ett alternativ till WIMP och dess metod att starta programmet genom inställningar och laddning av bana via interaktivt fönster, är att starta SubTrack från en kommandoprompt. Om användaren är van att arbeta med kommandofönster, så kan hon mata in alla de nödvändiga parametrarna som behövs vid en start som argument på kommandoraden. I detta fall bör även en startdialog ske vid programstart, så att användaren får en tydlig återkoppling och lätt kan ändra något av alternativen. Om användaren har matat in felaktiga argument eller om användaren ångrat sitt val, kan hon ändra i initieringsdialogen.

Startdialog hjälper användaren att minska minnesbelastningen, hon behöver inte komma ihåg vilken av inställningarna som gjorts och vad som är kvar att utföra. Allmänt anses kommando via en kommandorad öka en användares minnesbelastning. Så är dock inte alltid fallet, användaren kan ha sparat kommandon och parametrar från en tidigare körning i en fil.

6.6 Prestanda

De enda prestandabrister som kommer att föreslås förändrade är presentationen och uppritningen av de grafiska komponenterna. Se *Ritfunktion*, 6.4.2.

Förlorande av interaktionsmöjlighet vid utläggning av ny boj när Partikelfilter och automatisk bojutläggning används, kommer inte att behandlas utan lämnas till en eventuell senare implementation.

När simuleringen sker efter en splinebana, förloras interaktionen. Detta lämnas utan åtgärd för en eventuell senare implementation.

7 Intervjuer med demonstratörer

Detta kapitel behandlar det användardeltagande som skedde i utvecklingen av ett användarvänligare SubTrack. Syftet är att låta all den kunskap som finns hos användarna komma interaktionsdesignen till godo.

Innehållet i detta kapitel är en uppgiftsanalys för en demonstratör, 7.1. Därefter följer användardeltagande av användare A, 7.2, samt användare B, 7.3. Kapitlet avslutas med en beskrivning av startdialogen mellan användare och system, innan simuleringssekvensen startas.

Intervjuer har gjorts med två anställda på FOI. Användare A har varit anställd i 15 år. A har en erkänt god pedagogik och är den som demonstrerat SubTrack flest gånger. A har en god datorvana och har själv varit med om att utveckla delar av SubTrack. Både MS Windows på PC och X-Windowsmiljö under Unix, är plattformar som A är väl bekant med.

Användare B är nyanställd och har viss vana vid SubTrack. Den första arbetsuppgift som B tilldelades var att optimera den matematiska funktionen Partikelfilter, en av de algoritmer som kan nyttjas för beräkning i SubTrack. Omfattningen av användningen var inte så stor som i demonstratörens fall, men det resulterade i en grundläggande förståelse för programmet. B har, liksom A, god datorvana på de aktuella plattformarna.

7.1 Uppgiftsanalys

SubTrack används av ett flertal aktörer i olika roller. För dessa användare finns ett flertal olika scenarier, detta arbete skall dock endast behandla en av dessa, nämligen demonstratörens.

För att identifiera de scenarier som demonstratören använder, gjordes en intervju med den demonstratör på FOI som hade mest erfarenhet, användare A. Utöver detta besitter han, enligt de andra på avdelningen, en bra pedagogisk förmåga. Det scenario som identifierades under denna demonstration och intervju, ligger till mall för hur demonstratören skall utföra sin demonstration.

Användare A förevisade hur en demonstration av SubTrack kan se ut. Demonstrationen är pedagogisk, den börjar med att visa grundläggande funktionalitet, för att senare bli mer avancerad. Scenariot kommer att användas som mall vid fortsatt utveckling av SubTrack mot en mer användarvänlig version, med fokus på en demonstratörs behov. Demonstrationen består av sju moment, efter varje körning måste programmet startas om:

1. Följning av en ubåt. Inga grafiska objekt visas förutom de som finns som förval, nämligen "Buoys" och "Measurement ellipse". Förval för inställning av bojutläggning är "Manual" och "Kalman Filter" för beräkningsalgoritmen.
2. Frysa körningen, "Freeze". Simulering och experimentering med bojutläggningen, för att visa hur bojarna skall placeras för att optimera beräkningen av var ljudet kommer ifrån (målet befinner sig). Grafiska objekt är samma som ovan.
3. Prediktionsellips för att visa historien av var ubåten troligtvis har befunnit sig. Grafiska objekt som läggs till är "prediction ellipse history".
4. Automatiserad bojutläggning används för att visa när programmet beräknar hur bojarna skall placeras och läggas ut för att, så optimalt som möjligt, kunna förutsäga var ubåten befinner sig. Bojgenskaper sätts till "Automatic". Inga ytterligare grafiska objekt.
5. Visa målet för att bevisa att "prediction ellipse history" stämmer. Ytterligare grafiska komponenter är "target" och "path".
6. Visa hur Partikelfilter fungerar, vilket är den andra av de två beräkningsalgoritmerna. Bojgenskaper sätts till "Manual" och "Particle Filter". Ingen "measurement ellipse" kommer att visas som förval, men "particle cloud" adderas som grafisk komponent. Både "target" och "path" inaktiveras så att de inte ritas grafiskt.

7. Att visa hur Partikelfiltret fungerar med automatisk utläggning, tar alltför lång tid att köra för att normalt kunna demonstreras. Användare A har därför skapat ett videoklipp som tillåter hopp över de moment, då tung beräkning får processen att stå still. Om denna demonstration ändå körs eller visas, sätts bojegenskaperna till "Automatic". De grafiska komponenterna "target" och "path" läggs till för att visa att Partikelfiltret verkligen fungerar.

7.2 Användardeltagande i designförslag, användare A

7.2.1 Användartest

Vid varje start sker ett antal moment, som kan ses som en initiering innan en körning av en bana påbörjas:

1. Breder ut fönstret till full skärmstorlek.
2. Game – Load – ladda bana ur filvalsdialog.
3. View – Tracks – välj vilka grafiska objekt som skall visas
4. View – Zoom/Pan – zooma så att banan och bojar troligtvis kommer att synas under hela körningen.
5. Options – Tracking Method – välj den beräkningsmetod som passar och om bojutläggningen skall vara manuell eller automat.

Användare A demonstrerade det tillvägagångssätt en demonstratör bör arbeta efter, varför det inte är nödvändigt att redovisa tillvägagångssättet steg för steg. En sammanfattning görs istället direkt. De besvärliga situationer som uppkom under sekvensen beror på dålig användarvänlighet i programmets gränssnitt:

- Glömde att ställa tillbaka tillståndet "Zoom/Pan" till normaltillstånd, när en manuell körning startades. Detta resulterade i att inga bojar kunde placeras ut i början av simuleringen och A startade om programmet från början.
- Glömde att byta beräkningsmetod (manuell Kalmanfilter är förvald inställning). Startade om programmet.
- När "Zoom out by step" användes lade sig dialogruta i mitten på grafikfönstret. Detta fick till följd att när klick på "Normal mode" gjordes, och dialogrutan stängdes, så ritades inte det som varit under fönstret om. "View - Refresh" var nödvändigt att använda före att rita om grafikfönstret.
- "View – Zoom/Pan" gick inte att använda, detta för att ingen bana var laddad i SubTrack.

7.2.2 Användare A kommenterar SubTrack

En intervju med användare A, där A får tillfälle att påpeka och föreslå förändringar av gränssnittet.

Generellt, funktionalitet för programmet

- En startsekvens, som innehåller defaultvärden, borde starta hela simuleringen. En inspelningsfunktion som kan spara den simulering som körts. Detta i kombination med en logg som beskriver tidsaxel och variabler för systemet för denna tidpunkt (funktionaliteten är inte användbar för demonstratören, kanske för användarrollen som analytiker).
- När simuleringen är klar borde inte "Stop" aktiveras utan "Freeze", eftersom detta skulle möjliggöra modifiering av bojar efter banan är slut.
- Omstart skulle ske med samma bana eller en ny, "Restart"?
- Väsentliga systemparametrar borde vara synliga under körning.
- Funktion för att ångra är inte önskvärt, eftersom programmet är ett "spel".
- Implementeringsfel – när "Game – Freeze" är aktiverad slumpas inte "Buoy position disturbance" om, varför "measurement ellipse" inte räknas om. Detta kan vara en lösning

för att ellipsen skall ligga still och inte upplevas att den hoppar runt i grafikfönstret, men det ger inte en rättvisande bild av simuleringssekvensen.

- Implementeringsfel – när Partikelfilter körs har bojarna felaktig (för lång) räckvidd, vilket ger Partikelfiltret en felaktig förbättring vid jämförelse mot Kalmanfilter.

Menysystem och dialogrutor

- Kortkommandon till frekvent använda funktioner, vilka kortkommandona är, borde stå efter namnet.
- Dialogfönstrens storlek borde minskas och placeras lämpligare.

Game

Load	Ladda bana skall kunna göras även om en bana har valts redan vid start. Dålig dialogruta, om förstoring sker så förstoras endast den vänstra kolumnen.
Run	
Run until	Bör delas i en funktion för att stega till en tidpunkt och en för att ta ett förutbestämt steg framåt. ”Run until next” kan vara endast ”Run until”.
Step time	
Stop	Bör ersättas av ”Freeze”, som även kan avsluta programmet.
Freeze	
Exit	

View

Tracks	Hör ”Buoy range” hemma i denna dialogruta?
Files	Borde vara under en huvudmeny, Info?
Buoys	Info om buoys på två ställen. Borde vara under Info?
Path Edit	
Zoom/Pan	Anpassning till vedertagna metoder för att zooma.
Refresh	

Deploy Använder inte denna funktionalitet.

Help

Options

Tracking method En separering av kolumner, tag bort överflödigt text, detta skall kanske in i en initieringsdialog.

Presentation

- Ramen borde inte vara grön, sticker i ögonen.
- Rullisterna är inte bra. Det är bättre utan rullister, eller att rullisterna kan rulla ett stort område, både lodrät och vågrät.
- Dialogfönstret ”Buoy deployment” öppnas när användaren vänsterklickar på grafikfönstret, men endast då stopptillstånd inträtt för simuleringen. Detta borde ingå i initieringsdialogfönstret, så att användaren kan förprogrammera bojutläggningen för en hel körning.

7.3 Användardeltagande i designförslag, användare B

7.3.1 Användartestet

B fick utföra de uppgifter som identifierats när A demonstrerade programmet, se *Användardeltagande i design, användare A, Uppgiftsanalys, 7.2*. Nedan följer de moment som B fick i uppdrag att genomföra samt resultat och kommentarer på detta. Sist i varje moment kommer kommentarer av intervjuaren om vad som kan ha orsakat felaktigheter eller svårigheter:

1. Följning av en ubåt. Inga grafiska objekt visas förutom de som finns som förval, nämligen bojar och "measurement ellipse". Bojegendskaper är "Manual" för bojutläggning och "Kalman Filter" för beräkning.

B: Laddade bana. Startade körning och placerar ut bojar. B kommenterar att det är en konstig ellips och har svårt att följa målet.

Intervjuledare: det är svårt att placera bojarna korrekt så att en bra ellips skapas i initialskedet och utan kunskap om hur banan löper skapas lätt förvirring. Vidare så skulle en utzoomning ha förenklat förloppet, genom att delar av ellipser inte hade befunnit sig utanför fönstret.

2. Frysa körningen, "Freeze". Simulering och experimentering med bojutläggningen för att visa hur bojarna skall placeras, för att optimera beräkningen av var ljudet kommer ifrån (målet befinner sig). Grafiska objekt är som ovan.

B: Startar som tidigare men använder "Freeze". Skall ta reda på hur bojar tas bort, och testar "view buoys" utan att hitta någon ledtråd. "Hur hittar man hjälp till hur bojar tas bort?" frågar B. "Gillar inte hjälp, men jag måste väl in där ändå." Letar under hjälp, men hittar ingen. Efter tips att kolla under "Help" och "Mouse" hittar hon den hjälp hon behöver. Att modifiera bojpositioner gick efter detta utmärkt.

Intervjuledare: B visste att utläggning av bojar gjordes med vänsterklick, det var dock inte intuitivt att högerklick tog bort bojar. Att leta hjälp om bojar under "Mouse" var heller inte naturligt. Hjälp borde ligga samlad under ett menyalternativ, nämligen "Help".

3. Prediktionsellips för att visa var ubåten troligtvis befinner sig. Grafiska objekt som läggs till är "prediction ellipse history".

B: Laddar bana, och aktiverar "prediction ellipse history". B säger att hon ser dåligt och aktiverar "Freeze". Öppnar "Zoom/Pan" under huvudmenyn "View" och använder "Zoom out by stepanvändare". Glömmer att aktivera "Normal mode", varför bojutläggningen inte fungerar, utan zoomning istället sker. Efter detta zoomar B om, och slutför uppgiften.

Intervjuledare: Att hantera zoomfunktion är inte enkelt, inte ens för en van användare. Se *Användardeltagande i design, användare A, 7.2*.

4. Automatiserad bojutläggning används för att visa när programmet beräknar hur bojarna skall placeras och läggas ut för att optimalt kunna förutsäga var ubåten befinner sig. Bojegendskaper sätts till "Automatic". Inga ytterligare grafiska objekt läggs till.

B: Laddade banan, sätter "Kalmanfilter" och "Automatic mode". Innan körningen justerades zoomning med "Zoom out by stepanvändare", varefter "Normal mode" aktiveras. Allt korrekt. B konstaterar att ellipser skriver över och förstör varandra.

Intervjuledare: Instämmer med iakttagelsen avseende de grafiska funktionerna.

5. Visa målet för att bevisa att "prediction ellipse history" stämmer. Ytterligare grafiska komponenter är "target" och "path".

B: Laddar banan, Zoomar och sätter "Automatic" och "Kalman", Aktiverar de grafiska

komponenterna "Target", "Path" och "Prediction ellipse history".

Intervjuledare: Bra, inga kommentarer.

6. Visa hur Partikelfilter fungerar, vilket är en alternativ beräkningsmetod. Bojgenskaper sätts till "Manual" och "Particle Filter". Ingen "measurement ellipse" kommer att visas förvalt men "particle cloud" adderas som grafisk komponent. Både "target" och "path" inaktiveras så att de inte ritas grafiskt.

B: Laddar bana, sätter "Manual" och "Particle filter". Väljer den grafiska komponenten "Prediction ellipse history". Zoomar ut, men återställer inte tillståndet till "Normal mode".

Intervjuledare: Återigen ett misstag vid zoomning, funktionaliteten är besvärlig att hantera.

7. Att visa hur Partikelfiltret fungerar med automatisk utläggning tar idag alltför lång tid att köra för att kunna demonstrera, varför ett videoklipp har skapats som tillåter hopp över de moment då tunga beräkningsfunktioner får processen att stå still. Om denna demonstration ändå körs sätts bojgenskaperna till "Automatic". De grafiska komponenterna "target" och "path" läggs till för att visa att Partikelfiltret verkligen fungerar.

B: Laddar bana, väljer grafisk komponent "Particle cloud", "Target", "Path" och "Prediction ellipse history". Zoomar, men glömmer att välja "Automatic" och "Particle filter".

Intervjuledare: Återigen ett misstag i inställningar innan själva körningen har börjat.

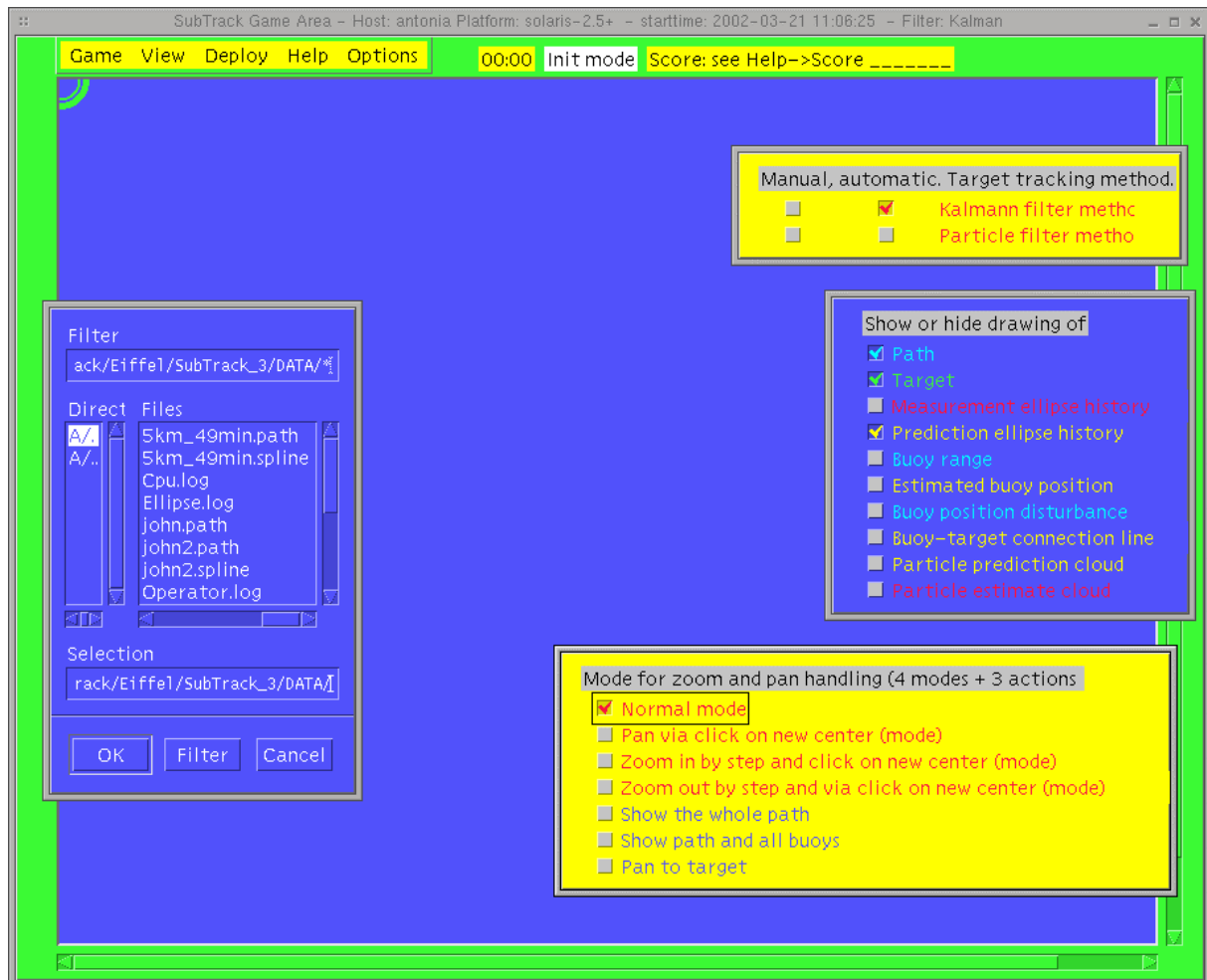
7.3.2 Användare B kommenterar SubTrack

I diskussionsform redogjorde B för tankar om interaktionen med SubTrack. Denna diskussion var helt ostrukturerad, varför upprepning kan förekomma från tidigare stycke.

- Det ger inte ett professionellt intryck, med grafik som förstör tidigare ritad grafik.
- Bättre zoom funktion efterlyses. Programmet måste frysas innan zoomfunktion används, eftersom den tar lång tid att använda (många steg).
- Ett eget verktygsfält, skulle vara bra för zoomning, grafikobjekt, refresh, stop och run. Dessa funktioner används så ofta att de snabbt måste kunna nås.
- Statusfältet uppe till höger innehåller tiden och tillståndet som B ansåg som intuitiva, medan "score" och "viewport" inte ansågs intuitiva. Som förklaring angav B att hon inte hittills haft användning av score. B hade sett antalet bojar som placerats ut som ett mått på om Bs optimering av Partikelfiltret givit ett bättre resultat eller ej. Istället för viewport har B använt det värde som stått längst upp på skärmen som ett mått på den zoomfaktor programmet visade.
- Det är allmänt för glada färger, detta ger inget seriöst intryck. En ram runt objekten i statusfältet skulle hålla ihop denna bättre.
- Menyradens huvudobjekt. "Options" är inte intuitivt, särskilt när den bara innehåller en undermeny. Deploy ansågs som mindre viktig än de övriga. "View" var bra.
- Vad är skillnaden mellan "Freeze" och "Stop"?
- "Run until" och "Run until next" är intuitiva, medan "Step time" kanske bara skulle heta "Step".
- "Graphic components" istället för "Tracks".
- Färger i "Tracks" lite tveksamt grupperade.
- Korta namn och hjälp skulle kanske vara bättre än endast långa namn.

- ”Deploy” känns som det hör samman med ”Tracking method”.

7.4 Startsekvens



Figur 18: De dialogfönster som demonstratören vanligtvis öppnar, när en simulering skall köras i det befintliga SubTrack programmet.

När användartesterna gjordes med användare A och B, framträdde ett mönster i initieringssekvensen. Detta mönster hade tidigare anats och var den största anledningen till att användartesterna utfördes. Vid start av programmet öppnade demonstratören vanligtvis fyra dialogfönster, för att välja fem moment av inställningar. Dessa borde samlas i en startdialog:

1. Beräkningsmetoden som skall användas: Kalman- eller Partikelfilter.
2. Utläggingsmetoden för bojar som skall sköta användas. Manuell (användaren) eller automat (programmet). Återfinns i samma dialogruta som ovan.
3. Vilka grafiska objekt som skall visas i ritytan.
4. Laddande av önskad bana (beskrivningen av målets väg).
5. Zoomfaktorn som simuleringen skall ha vid start av simuleringsekvensen.

Denna startsekvens används som mall för hur en användare antas starta en simulering.

8 Resultat

Resultatet av arbetet är ett designförslag till en utveckling av det befintliga SubTrack, för att skapa ett mer användarvänligt gränssnitt för en demonstratör. Detta kapitel innehåller dessa förslag till design. Syftet med kapitlet är att användaren skall få en beskrivning av de förändringar och tillägg som föreslås.

Kapitlet består av en beskrivning av yttre påverkan på SubTrack, som gör att möjliga designförslag begränsas. Första underkapitlet, 8.2, är förslag till förändringar i SubTracks funktionalitet. Andra underkapitlet, 8.3 beskriver användarens interaktion med systemet, och kapitel 8.4 beskriver systemets presentation till användaren. Det designförslag som föreslås för startdialogen beskrivs i kapitel 8.4. Avslutningsvis diskuteras prestanda i kapitel 8.5.

8.1.1 Översikt

Tre större tillägg till programmet föreslås:

- en startsekvens som skapas för att förenkla sekvensen innan start av en simulering. Detta för att initieringsfilen med programparametrar skall göras åtkomlig för användaren. Användaren skall enkelt kunna manipulera inställningarna i SubTrack, därför föreslås också direktmanipulation. Detta innebär att kortkommandon och en verktygsrad med den funktionalitet som ofta används skapas som komplement till menyvalet.
- Viss utökning av popupmenyer kommer att skapas, tillgänglig i ritytan.
- En uppsnygning av färg och val av layout, gällande både huvudfönster och dialogfönster.

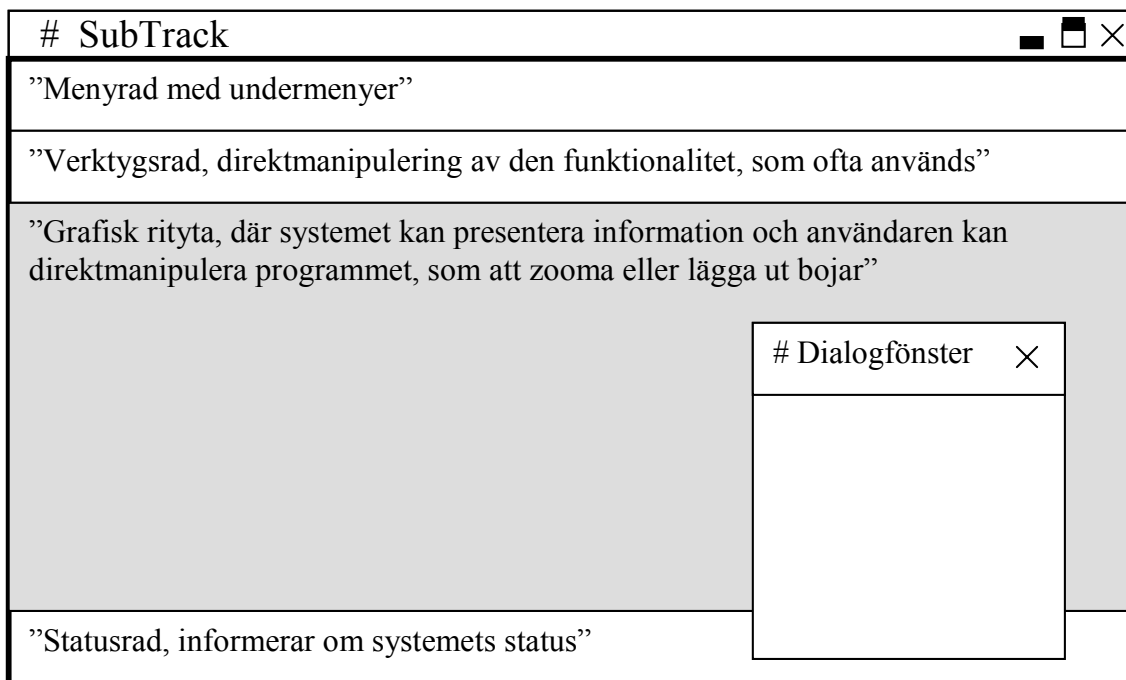
8.1.2 Begränsningar

Designförslaget för SubTrack bygger på expertutvärdering, intervjuer med potentiella demonstratörer och på uppgiftsanalys. Designförslaget tar upp förändringar som bör göras på nästa version av SubTrack. I varje stycke finns beskrivet nuvarande status, designförslag och en diskussion om anledningen till detta val av design.

SubTrack baseras på ett Unix- eller MS Windowsplattform med tillhörande skärm, tangentbord och mus. Programmet kommer att utformas som en fristående applikation och designförslaget kommer att hålla sig inom dessa ramar. Inga förslag eller diskussioner kommer att ske rörande lösningar som ligger utanför dessa ramar. Vidare är det viktigt att demonstratörer och publik av förevisningarna får en så tydlig bild av programmet som möjligt. Detta innebär att utseendet skall vara enkelt, och efterlikna de produkter som är vanliga på marknaden, i demonstratörens fall produkter som är baserade på MS Windows för PC eller X-Windows under Unix.

SubTrack har funnits en tid och flertalet användare som kan vara aktuella som demonstratörer har redan vant sig vid kommandoinitiering eller interaktion av typen WIMP. Ett grafikfönster presenterar simuleringen och i menyraden ger användaren de instruktioner som programmet skall utföra. Detta kommer inte att förändras, och inte heller användarens sätt att placera och ta bort bojar genom muskommandon eller att musen används för att hantera meny.

De krav som skall utrönas är vad demonstratören kräver, i form av interaktion mot programmet. Det är viktigt att gränssnittet struktureras och organiseras på ett sätt, som gör att användaren kan interagera med produkten på ett enkelt sätt, hjälpt av kunskap om liknande system. Krav på systemets prestanda kan t.ex. vara att svarstider från systemet inte skall vara otillbörligt långa. Användarorganisationens definition av syftet med verksamheten skall passa den produkt som utvecklas. De kommer också ha krav på tillgänglighet, prestanda och underhåll samt en fungerande hjälp. Det sätt de arbetar på måste också tillvaratas och skyddas.



Figur 19: Programmets huvudkomponenter.

8.1.3 Läsanvisningar

Kapitlet är indelat i fem huvudgrupper och deras underkapitel. En liten orientering om dess innehåll följer nedan:

SubTracks funktionalitet

Ingen avsevärd förändring av funktionaliteten kommer att föreslås för SubTrack. Funktionaliteten har sin källa i den forskning som SubTrack har till uppgift att visa.

Användarkontroll

- Menyn – de tillägg och förändringar som föreslås, syftar till att strukturera menyraden och dess undermenyer. Konsekvensen kommer att öka, och kopplingen mellan systemet och den verkliga världen görs starkare.
- Dialogfönster – förändringarna syftar till att förbättra överskådligheten och ge dialogfönstren ett konsekvent utseende.
- Körkontroll – kontroller som styr simuleringsssekvensen. Användarkontrollen skall göras tydligare.
- Zoom- och panoreringsfunktioner – de förändringar som görs syftar till att förenkla åtkomst till zoomning och att minimera nödvändiga operationer. Operationer på presentationen från systemet är en användarkontroll i form av direktmanipulation, men påverkan har så stor betydelse att det läggs som ett separat underkapitel. Effektiviteten av åtgärder under simuleringsssekvensen är väsentlig för användarvänligheten.
- Direktmanipulationen – tillägg och förändringar som föreslås, syftar till att snabba upp åtkomligheten av programfunktioner. Effektiviteten av åtgärder under simuleringsssekvensen är väsentlig för användarvänligheten.
- Initieringsparametrar – de parametrar som styr programmet, görs åtkomliga via en dialogfönster. Vissa parametrar kommer dock icke vara redigerbara, utan endast fungera som information från system till användare. Användarkontrollen ökar.

Presentation:

- Huvudfönster – programmets huvudfönster kapslar in de övriga komponenterna i systemet. Inga övergripande förändringar kommer att föreslås, men en något mer estetisk och minimalistisk design blir följd.
- Ritfunktionen – förändringar i informationen som presenteras grafiskt av programmet. Presentationen måste vara tydlig för att synligheten av systemets status skall vara god. Presentationen har även stora följdverkningar på prestanda.
- Statusinformation – programmets väsentliga dynamiska parametrar. Det är viktigt att användaren har den information lätt åtkomlig som är nödvändig för simuleringen. Detta minskar belastningen på korttidsminnet.
- Hjälp och dokumentation – de förändringar som görs på hjälpfunktionerna, är att strukturera hjälpfunktionerna och göra dessa konsekventa genom hela programmet. Hjälptexter kommer inte att beröras, utan lämnas för senare implementation.

Startsekvens:

Startsekvens är de moment som alltid görs före körning av en simulering. Detta föreslås samlat i en förvald startsekvens. Detta görs för att minimera de operationer som krävs vid varje körningstillfälle, användarkontrollen ökar.

Prestanda:

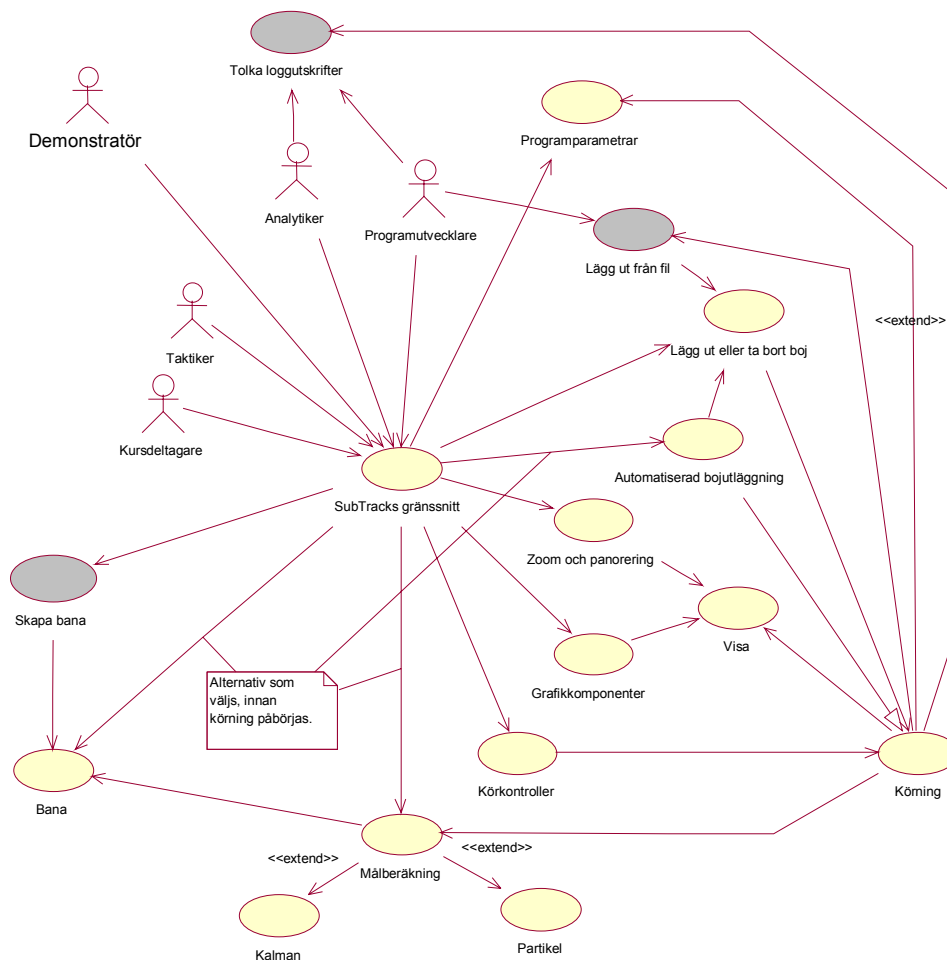
Större prestandaförbättringar kommer inte att vara aktuella, eftersom ingen implementation kommer att utföras. Detta gäller dock inte den grafiska ritfunktionen, som kommer att förbättras genom en strukturering av de objekt som ritas.

8.1.4 Roller och aktörer

Den enda roll som kommer att beskrivas i denna rapport, är demonstratörens. Demonstratörens roll skall göras så användarvänlig som möjligt, för att demonstratören inte skall behöva vara expert på att hantera systemet. Vidare skall den grupp som SubTrack demonstreras för, lätt känna igen sig från de program som finns i referenskorger och därigenom tämligen enkelt förstå programmet. Se *Roller och aktörer*, 2.1.2 och *Konventioner*, 6.1.4.

8.1.5 Användningsfall

Användningsfallen kommer inte att förändras nämnvärt. Som tidigare påpekats, berör denna rapport endast användarvänligheten och en större förändring av SubTracks funktionalitet är inte aktuell.



Figur 20: Användningsfall för designförslaget

Användningsfallet förändras i programparametrarna uppe till höger. De tillåts nu att redigeras av demonstratören, och därmed av alla användare.

8.2 SubTracks funktionalitet

Användaren föreslås kunna både backa och köra framåt i tidsaxeln för simuleringen. Detta kan komma att ske på två sätt, antingen genom att alla beräkningar behålls oförändrade och en återuppspelning kan ske av valt tidsintervall, eller så skall användaren kunna backa för att ångra en åtgärd och sedan fortsätta en ny simulering från detta tidssteg.

Från detta är inte steget så långt till att använda SubTrack som ett uppspelningsverktyg för redan körda simuleringar. Detta kommer dock inte att ingå i designförslaget. Vid en senare vidareutveckling av SubTrack kan detta vara ett intressant tillägg.

8.2.1 Simuleringen

Simuleringssekvensen är SubTracks ryggrad, och den kommer inte att förändras. SubTracks funktion är att visualisera forskning, och denna rapport behandlar endast designen av denna visualisering. De kontroller, som påverkar simuleringssekvensen kommer att förändras. Se *Körkontroller*, 8.3.3. Vissa av utläggningsmetoderna för bojar kommer att förändras, funktionaliteten kommer huvudsakligen att kännas igen. Se *Tools*, 8.3.1.2.5.

8.2.2 Beräkningsfunktion

Beräkningsfunktionen kommer inte heller att förändras, då rapportens syfte är att göra SubTrack användarvänligare för demonstratören.

8.2.3 Beräkningspresentation

Vad som presenteras, kommer till största delen att behållas oförändrat. Det sker ett tillägg, räckvidden av målets ljud, som ges ett eget grafiskt objekt. Se *Designförslag (ytterligare funktionalitet)*, 8.4.2.2.

8.2.4 Programparametrar

Programparametrar kommer inte att förändras, dock kommer en möjlighet att redigera dessa parametrar att erbjudas. Se *Initieringsparametrar*, 8.3.6.

8.2.5 Tillstånd

Tillstånden för simuleringen kommer att behållas oförändrade till funktion, dock kommer namnen att bytas ut. Körkontrollerna föreslås fungera lite annorlunda, det kommer t.ex. att vara möjligt att backa i simuleringen. Se *Körkontroller*, 8.3.3.

Mustillstånden kommer att förändras för zoom och panoreringsfunktioner, men framför allt de pekarsymboler som synliggör det tillstånd som musen intagit. Se *Zoom- och panoreringsfunktioner*, 8.3.4. Mustillståndet kommer också att förändras, när stopptillståndet för simuleringssekvensen har inträtt. Inget bojplaneringsfönster kommer längre att öppnas, användaren får själv sköta detta från menyraden. Se *Deploy Plan, under Tools*, 8.3.1.2.5.

Tangentbordstillstånd kommer att behållas oförändrade. Det kommer även fortsatt att vara nödvändigt att aktivera ett inmatningsfält, innan den kan ta emot inmatning från tangentbord.

8.3 Användarkontroll

Användarkontroll beskriver den inmatning av kommandon som användaren kan utföra för att styra simuleringssekvens, programinställningar, beräkningsfunktion eller utformningen av presentationen.

8.3.1 Menyrad

8.3.1.1 Nuvarande status

Menysystemet är inte konsekvent mot referenskorgen och har tillkommit ad hoc under programmets utveckling. Menysystemet har en otillräcklig gruppering av menyalternativ och ibland är dessa menyalternativ inte användarvänligt utformade. Det saknar konsekvens både i menyer och i dialogfönster. Menysystemet är svårt att känna igen eftersom inga av de konventioner, som andra program på windowsplattformar har, har följts. Se *Konventioner, 6.1.4*.

8.3.1.2 Designförslag

Vid design av menysystem, formulär och dialogrutor så skall strukturen och sekvensen av dialoger organiseras så att de anpassas efter användarens uppgift. Objekten skall vara distinkta och lättförståeliga.

Menyer är effektiva för att de erbjuder användaren ledtrådar för igenkännande, snarare än att användaren tvingas memorera kommandonamnet. Namn som är metaforer kan användas, så att en föraning ges om vad som menyalternativet resulterar i [Shneiderman].

Programmet förändras inte avseende menysystemets struktur med huvudmenyer och undermenyer. Inga expanderande menyer, undermenyer som öppnar nya undermenyer, har använts. En undermeny kan, liksom tidigare, öppnas upp till en dialogruta. Detta markeras, genom att två punkter avslutar menynamnet. Punkterna erbjuder användaren synlighet av systemets status, samt att det följer de konventioner som finns i referenskorgen. Se *Konventioner, 6.1.4*.

8.3.1.2.1 Huvudmenyn

Huvudmenyn har omformats, och består endast av fyra huvudmenyer. Detta för att designen skall vara så minimalistisk som möjligt, men främst beroende på att konventioner från referenskorgen skall följas. Huvudmenyns "File", "View" och "Help", finns i alla program. "Tools" är ett vanligt samlingsnamn för tillbehör till program. Huvudmenyerna är:

- File Finns i alla program. Filhantering och avsluta måste finnas med.
- View Alla förändringar i det grafiska utseendet återfinns här.
- Tools Här finns alla funktioner som på något sätt utökar den grundläggande funktionaliteten i programmet.
- Help Här finns all hjälp för programmet, förutom den som tillhör en dialogruta eller statusrad. Dock kan denna information vara så betydande att den finns både under menyn "Help", och hjälpen till dialogrutan.

File	View	Tools	Help
Load Path..	Toolbar	Game Controls..	User Manual..
Preferences..	Statusbar	Deploy Buoys..	Demonstrator..
Exit	Graphics..	Deploy Pattern..	System Comp..
	Zoom/Pan..	Path Editor..	About..
	Refresh		

Figur 21: Menysystem. Om namnet avslutas med två punkter, öppnas en dialogruta när menyvalet aktiveras.

8.3.1.2.2 Undermenyn

De meny namn som väljs, skall ge en bra och kortfattad beskrivning om vad åtgärden kommer att resultera i. Om undermenyn består av grupper som hör ihop, avgränsas dessa med en linje i undermenyn.

8.3.1.2.3 File

Finns i alla menybaserade program, längst till vänster.

- Load Path.. Laddning av ny bana bör även vara synlig när en bana har satts i kommandofönstret vid start. Möjligheten att sätta bana skall inaktiveras när en körning av simulering har påbörjats.
Se Dialogfönster, 8.3.2.
- Avgränsare Inläsning av fil och inställning av program hör inte ihop.
- Preferences.. Initieringsparametrar som kan redigeras eller läsas från en fil. Kan även nås från startdialogrutan. Få av dessa parametrar är redigeringsbara efter simuleringsstart.
Se Initieringsparametrar, 8.3.6, Dialogfönster, 8.3.2.
- Avgränsare Avslutande av program, hör inte ihop med ovanstående
- Exit Ingen förändring, behålls i nuvarande utformning.

8.3.1.2.4 View

Sköter alla förändringar av utseendet, både avseende programfönstret och programfunktionen grafikfönstret.

- Toolbar Ny funktionalitet som styr om verktygsfältet skall visas eller inte.
Se Designförslag (verktygsfält), 8.3.5.2.
- Statusbar Ny funktionalitet som styr om statusfält skall visas eller inte.
Se Statusfält, 8.4.3.
- Avgränsare Gruppen ovan sköter anpassningar av vilka huvudkomponenter, som skall visas i programmet. Den undre gruppen anpassar det funktionalitet av grafikfönstret.
Se Begränsningar, 8.1.2, för en förklaring av programmets huvudkomponenter.
- Graphic comp.. Tidigare ”Tracks”, dålig metafor. Grafiska komponenter som skall visas på grafikytan. Dess funktionalitet kan även nås via det nya verktygsfältet.
Se Dialogfönster, 8.3.2.
- Zoom/Pan.. Dialogrutans namn kortas och symboler skapas för varje funktion. Dess funktionalitet kan även nås via det nya verktygsfältet.
Se stycket Zoom och panoreringsfunktioner, 8.3.4, Dialogfönster, 8.3.2 och Designförslag (verktygsfält), 8.3.5.20.

- Refresh Ingen förändring, behålls i nuvarande utformning. Skall även kunna nå från verktygsfältet. Se *Designförslag (verktygsfält)*, 8.3.5.20.

8.3.1.2.5 Tools

- Game Controls.. De kontroller som behövs för att manövrera simuleringen i en dialogruta. Funktionaliteten kommer även att finnas i verktygsfältet. Se *Körkontroller*, 8.3.3, *Designförslag (verktygsfält)*, 8.3.5.20, och *Dialogfönster*, 8.3.2.
- Deploy Plan.. En funktion som användaren tidigare kom åt när programmets tillstånd var ”Stop”. Funktionen nås numera från menyraden, oavsett vilket tillstånd simuleringssekvensen befinner sig i. Funktionen ger användaren möjlighet att planera sin utläggning av bojarna, både positionen och tidssteget. Att göra funktionen ständigt tillgänglig, ökar känslan av kontroll, ökar konsekvensen och förhindrar fel. Se *Dialogfönster*, 8.3.2.
- Deploy File.. I denna meny placeras även inläsning av en fördefinierad fil, vars syfte är att placera ut bojar enligt position på ett visst tidssteg. Det dialogfönster som används för att lokalisera filen, är samma som används när en bana skall laddas. Se *Dialogfönster*, 8.3.2, *Load Path..*, 8.3.2.2
- Deploy Patterns.. Före detta ”Deploy” med hjälpen åtkomlig i dialogrutan. Liten användningsfrekvens, varför denna funktion flyttas till en mer undanskymd position. Se *Dialogfönster*, 8.3.2.
- Avgränsare De tre övre funktionerna påverkar spelet, den undre startar ett eget program.
- Path Editor.. Baneditorn. Ett fristående program, för skapande av den bana som simuleringssekvensen använder, för att bestämma målets position. Ingår ej i rapporten.

8.3.1.2.6 Help

I hjälpmenyerna öppnas alltid ett dialogfönster. Dialogfönstren kommer att ges ett litet format och om en större textmassa skall kunna läsas, används en rullist. Det är viktigt att formaten ser likadana ut och att ett lättläst typsnitt används.

- User Manual.. En manual som kortfattat beskriver programmets funktionalitet, men som innehåller referenser till övrig dokumentation. Se *Hjälp och dokumentation*. 8.4.4, *Dialogfönster*, 8.3.2.
- Demonstrator.. En beskrivning av de moment som ingår i ett typfall för en demonstrator. Se *Hjälp och dokumentation*. 8.4.4, *Dialogfönster*, 8.3.2.
- System Comp.. En beskrivning av komponenter och filer som ingår i programmet. Hjälptill någon som skall administrera systemet. Se *Hjälp och dokumentation*. 8.4.4, *Dialogfönster*, 8.3.2.
- Avgränsare Den undre menyn är information om utveckling och version.
- About.. Ingen förändring, behålls i nuvarande utformning. Se *Dialogfönster*, 8.3.2.

8.3.1.3 Diskussion och argumentation

Menyer är M:et, i interaktionsmetoden WIMP, Windows Icons, Menus och Pointers. För utförligare information om denna interaktionsmetod, se *Inmatning, användaren till systemet*, 4.5.

Huvudmenyerna har omformats för att bättre överensstämma med programkorgens standard. En strukturering har skett av huvudmenyer så att standardnamn som är vanliga i kommersiella program används, samt att undermenyer grupperas att passa dessa huvudmenyer. SubTrack är ett litet program sett till funktionalitet och har endast behov av fyra huvudmenygrupperingar.

”File” placeras längst till vänster och ”Help” placeras längst ut till höger i menyraden. Detta är en generell placering i de flesta program, såväl PC som Unix-baserade. Huvudmenynamn som också ofta förekommer är ”View” och ”Tools”, de passar även för de resterande undermenyerna i SubTrack. ”View” får hantera all förändring av grafik, såväl verktygsfält och statusfält, liksom de grafiska objekt som ritas för att visualisera simuleringen. En vanligt förekommande huvudrubrik som inte används, är ”Edit”. Den används dock nästan uteslutande för textredigering, något som inte förekommer i SubTrack.

Menyn består av menyer, och undermenyer, som i flera av fallen öppnar upp dialogfönster. Anledningen till att expanderande menyer inte används, är att vid en sådan lösning ges inte användaren möjlighet att lägga upp verktygspalletter. Simuleringen har ibland liknats vid ett spel, och i ett spel har användaren ingen tid att slösa på menyöppnande utan vill ha snabbt och lätt åtkomliga funktioner. Det finns komplement till dialogfönster, direktmanipulation genom verktygsfält och peka och klicka direkt i grafikfönstret. Se *Dialogfönster*, 8.3.2, *Körkontroller*, 8.3.3, *Zoom och panoreringsfunktion*, 8.3.4, *Direktmanipulation*, 8.3.5.

Expanderande menyer är en undermeny som kan öppnas upp i flera undermenyer. Det finns exempel på program från Adobe där undermenyer kan dras av från menyn, för att då bilda verktygspalletter. Denna metod är dock inte intuitiv och kräver tidigare erfarenhet av ett sådant program. Om tekniken har stöd i Eiffel, har heller inte studerats.

8.3.2 Dialogfönster

8.3.2.1 Nuvarande status

Paletter är dialogfönster, som innehåller funktioner från programmet. De är avsedda för envägsinformation från användaren till systemet, med syfte att öka igenkänning för användaren så att valmöjligheter blir synliga. Vidare så ger detta ofta en återkoppling till användaren om statusen för systemet. Utanför detta kapitel kommer ingen särskiljning göras av paletter, de kommer normalt att refereras till som dialogfönster.

Alla menyalternativ som avslutas med två punkter öppnar ett dialogfönster. Det finns även viss direktmanipulation som öppnar en dialogruta, och självklart också meddelanden som systemet initierar.

Dialogfönstren är av många olika utseenden och dåligt avgränsade mellan olika slags funktionalitet. Dialogfönstrens storlek och placering är inte anpassade för användaren behov. Namngivningen på dialogfönstren är inte konsekvent, inte heller de objekt som finns i dialogfönstren har konsekvent utseende och placering. Exempel på detta är att inte textmassor visas i ett enhetligt format eller typsnitt, samt att namn på egenskaper eller objekt skrivs i färger.

8.3.2.2 Designförslag

Alla dialogrutor bör vara av en största storlek på ungefär 10 – 15 % av den aktuella skärmupplösningen. Dialogfönstret bör skymma så lite av bakgrunden som möjligt, men ändå rymma de knappar som behövs och tillåta ett tydligt läsbart teckensnitt. Om inte en textmassa rymms, skall en rullist erbjuda möjlighet att se all information.

Knapparna bör anpassas för att efterlikna de vanligaste programmen, ”Avbryt” längst till höger, ”OK” längst till vänster osv. Fönstren bör öppnas i ytterkanterna på programfönstren, för att så lite som möjligt störa bilden av ritytan. Namngivningen skall vara konsekvent med menyvalen för att inte förvillan användaren. Ingen färgsättning av dialogfönstren, allt skall ligga i gråskalan förutom eventuella ikoner och fönstrets namnlist. Denna bör färgsättas i en diskret blå färg.

En hjälpknapp och en knapp att stänga fönstret skall finnas i varje dialogfönster. Mer om hjälp finns under stycket Hjälp och dokumentation. En indelning i grupper av dialogfönster:

- Paletter, verktygsfönster som avses ligga tillgängliga under en simulering. De skall vara små och kompakta och stjåla så lite utrymme från grafikyten som möjligt. Paletter ökar användarens förmåga till igenkänning, hon ser vilka val som finns tillgängliga.
- Systemdialoger – information eller felmeddelande som initieras av systemet. Dessa skall erbjuda konstruktiv hjälp, så att användaren kan korrigera och fortsätta. Meddelanden skall helst inte börja med ord som ”Warning”, eftersom detta kan inverka negativt på användarens självförtroende.
- Interaktionsdialoger ger möjlighet att sätta parametrar i systemet. Dialogerna bör delas in i kategorier som är väl avgränsade, för att användaren skall känna igen sig. En koppling till den verkliga världen är önskvärd, men ibland får de grupperingar som finns i implementationen styra.
- Hjälpdialoger som ger information om systemet. Dessa finns i tre undergrupper, hjälp att köra och tolka programmet, information om systemets beståndsdelar och hjälp till en demonstratör. Dessa dialogrutor bör ha ett enhetligt utseende och ha en rullista för att se dokument större än textrutan. Hjälpen skall fokusera på användarens uppgift.

Dialogrutor som startas från menyraden:

Load Path..

Interaktionsdialog. Använder en standarddialog från Eiffel Vision. Användaren får möjlighet att söka den fil som önskas läsas in. Dialogfönstret är bra, men saknar möjlighet att öka storleken på en av de två kolumnerna, användaren tvingas använda rullisten. Det är den kolumn som visar filnamnet, och detta namn kan ibland vara ganska långt.

Preferences..

Interaktionsdialog. Ett formulärfönster, ”Form Fill”, där ett antal defaultvärden redan är ifyllda, men kan förändras innan ”OK” trycks. Efter att simuleringen har startat, är väldigt få av objekten i dialogen redigerbara, eftersom beräkningsfunktionen är beroende av dessa. Se *Initieringsparametrar*, 8.3.6.

Graphics..

Palett. Namnet ändras till en bättre metafor. Detta ger en bättre beskrivning av vad samlingen av funktioner kan tänkas utföra. Alla de grafiska komponenter som skall visas kan aktiveras genom att de bockas för. Komponenterna bör vara indelade i logiska grupper och vara korta för att lätt kunna läsas. Rutan för bocken skall vara markerad i samma färg som de grafiska objekten ritas i, men detta är endast ett komplement till den informationen som namnet och hjälpen ger.

# Graphics		X
<input checked="" type="checkbox"/>	Path	
<input checked="" type="checkbox"/>	Target	
<input checked="" type="checkbox"/>	Prediction	<input checked="" type="checkbox"/> Pred. History
<input checked="" type="checkbox"/>	Measured	<input type="checkbox"/> Meas. History
<input checked="" type="checkbox"/>	Sound Range	
<input type="checkbox"/>	Buoy Range	<input type="checkbox"/> Buoy Disturbance
<input type="checkbox"/>	Buoy Position	<input type="checkbox"/> Estimated Buoy
<input type="checkbox"/>	Buoy-Target	
<input type="button" value="Help"/> <input type="button" value="Close"/>		

Figur 22: Vilka grafiska komponenter som skall visas.

Zoom/Pan..

Palett. De funktioner som behövs för att anpassa användarens önskemål om förstoring eller panorering. Se *Zoom- och panoreringsfunktioner*, 8.3.4.

Game Controls..

Palett. Ny funktion, innehåller de funktioner som behövs för att manövrera simulering. Se *Körkontroller*, 8.3.3.

Deploy Buoys..

Lägger ut 10 stycken bojar på av användaren angiven position och tidssteg. Behålls enligt nuvarande utformning.

Deploy File..

Interaktionsdialog. Läser in en fil som lägger ut bojarna på förutbestämda punkter och vid förutbestämda tidpunkter. Användbar för en demonstrator som vill visa hur en automatisk bojutläggning med Partikelfilter borde se ut. Det är ingen riktig simulering, utan Partikelfilters långsamma utläggningsalgoritm har ersatts av en fil med förutbestämda positioner. Om demonstratören har förberett sig väl, stämmer positionerna ganska bra med var bojarna borde ha hamnat i en riktig beräkningssituation. Behålls enligt nuvarande utformning.

Deploy Patterns..

Verktygsfönster och interaktionsdialog. En nästan oanvänd funktion avsedd för manuell bojutläggning, placeras undanskynt i en undermeny. I dialogrutan väljs ett av de olika mönster som bojar kan placeras enligt, markeras med en bock. Detta ökar synligheten av systemets status.

Path Editor..

Interaktionsdialog. Startar ett program för skapande av bana som styr en simulering. Programmet är en fristående del av SubTrack och ligger utanför denna rapport.

User Manual..

Hjälpdialog. Här ges beskrivningar på hur programmet skall hanteras. Den hjälp som samlats från andra menyer:

- ”Score” som visas i statusfältet.
- ”Buoys”, informationsfönster om bojarna. Har legat som undermeny i ”View”, men visas numera i statusfältet.
- ”Mouse”, vilken funktionalitet som musen kan utföra. Skall även finnas som hjälp för direktmanipulation, då det är viktigt att hjälpen fokuserar på användarens uppgift.
- ”Files”, en undermeny från ”View” som förklarar vilka filer som används av programmet.

Demonstrator..

Hjälpdialog. En textbaserad hjälp för att utföra standardmomenten som en demonstrator vanligtvis vill förevisa.

System Comp..

Information. Till systemadministratör om vilka komponenter programmet består av, samt vilka filer som läses in.

About..

Hjälpdialog. Använder dialogfönster från Eiffel Vision.

8.3.2.3 Diskussion och argumentation

Ett konsekvent utseende är viktigt för att användaren skall känna igen sig och kunna fokusera på meningen med dialogfönster, istället för att förvirras av ett inkonsekvent utseende. Dock bör dialogfönster ges ett specifikt utseende, för att passa den grupp av dialogfönster som den tillhör. En estetisk och minimalistisk design är viktigt för att användaren skall trivas. Vidare förhindrar en konsekvent design uppkomsten av fel. Synlighet av systemets status blir tydligare.

Dialogfönster skall inte förstöra annan grafik. Det händer ofta i det befintliga SubTrack, bland annat när fönster som ligger ovanför det grafiska ritfönstret flyttas. Detta beror delvis på att utvecklaren som gjort det grafiska gränssnittet i SubTrack, inte har tillräcklig kunskap om hur grafikbiblioteket Eiffel Vision fungerar. Det finns en ny version av grafikbiblioteket, Vision 2, som kan vara enklare att hantera, men tillfälle har inte givits att prova detta bibliotek vid en implementation av SubTrack.

8.3.3 Körkontroll

8.3.3.1 Nuvarande status

Körkontrollerna förändrar det tillstånd som programmet befinner sig i. Det är viktigt att enkelt kunna förändra simuleringens tillstånd, då detta kan ses som ryggraden i programmet. I det befintliga SubTrack återfinns alla körkontrollerna i menyraden under "Game".

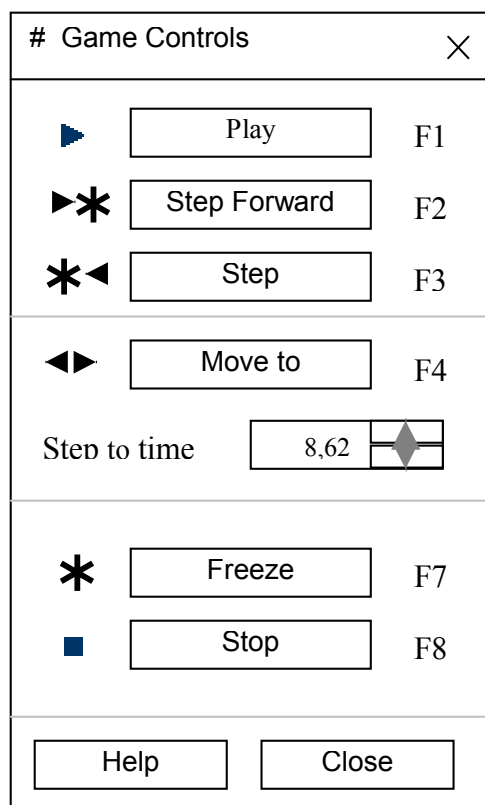
Körkontrollerna har inte en fullständig koppling till den verkliga världen. Det är t.ex. inte lätt att förstå vad funktionaliteten "Freeze" resulterar i.

Programmet bygger på en följd av händelser, varav några slumpmässiga faktorer. Detta omöjliggör hopp framåt i programmet utan att aktuellt tidssteg redan har passerats. Det finns ingen möjlighet att backa i programmet och återgå till ett redan passerat tidssteg.

När en simulering har kört klart, nått banans slut, måste programmet stängas ned och startas om igen för att kunna köra nästa simulering. Det finns ingen möjlighet att starta om utan att stänga programfönstret.

8.3.3.2 Designförslag

Funktionerna är endast till viss del intuitiva. Funktionen "Freeze" är svår att förstå, men behålls eftersom den är en speciell funktion för SubTrack och inte har någon motsvarighet i referenskorgen.



Figur 23: Kontrollpanel för körning av simulering.

Viss del av funktionaliteten finns även i verktygsfältet. Om användaren har valt att inte visa verktygsfältet, kan istället denna dialogruta läggas flytande på programfönstret. Den fullständiga uppsättningen av körkontroller:

- ”Run” – startar eller fortsätter simuleringen från frysläge.
- ”Step Forward” och ”Step Backward” – hette tidigare ”Step time” eller ”Step 4 seconds”. Kör simuleringen ett tidssteg framåt, tidssteget finns definierat i initieringsfilen. Efter tidssteget inträder programmet i ”Freeze”. Det går även att stega bakåt.
- ”Move To” – hette tidigare ”Run Until”, och föreslås numera även kunna backa. Kör programmet till det tidssteg som angivits i inmatningsfält, under funktionen i dialogrutan. Ett tillägg är att funktionen föreslås kunna backa tillbaka i spelsekvensen. En dialogruta skall aktiveras, när simuleringsssekvensen backar. Där skall användaren få tillfälle att ange om simuleringen ångrats och skall räknas om från detta tidssteg, eller om tillbakaflyttningen på tidsaxeln syftar till att kontrollera spelsekvensens historia, men behålla sekvensen oförändrad.
- ”Freeze” – beräkningsfunktioner körs fortfarande trots att målet är stoppat. Detta tillåter modifieringar på status för bojar så att modifiering kan göras för att optimering av bästa bojplacering kan uppnås. Inte förändrad jmf. med tidigare version.
- ”Stop” – både mål och beräkningsfunktion stoppas. Det är inte stor skillnad på ”Stop” och ”Freeze” eftersom det kan finnas tillfällen då användaren med säkerhet vill veta att ingen förändring kan ske. ”Freeze” behålls eftersom denna funktion känns lite tryggare och bidra till känsla av användarkontroll.

Funktionerna för att manövrera simuleringen är färre i verktygsfältet. Där finns endast de väsentligaste, de som måste nås direkt, exempelvis stopp. Se *Designförslag (verktygsfält)*, 8.3.5.20.

Möjlighet att backa programmet är viktigt för användarens frihet. Användaren kan tillåta sig att göra fel, som sedan kan korrigeras, genom att tiden backas tillbaka. Användaren kan också önska att studera simuleringsssekvensens historia.

För att en omstart av programmet skall kunna göras måste det ske en radering av den tidigare historien samt en återställning till defaultvärden av alla variabler som förändrats under körningen.

8.3.3.3 Diskussion och argumentation

En genomgång har gjorts av vilka funktioner som behövs. Viss inkonsekvens har tagits bort, och funktionalitet har lagts till eller tagits bort. Kvaliteten på den föreslagna utformningen avsågs att kunna verifieras genom användartester. Inga sådana tester kunde dock genomföras, varför en kognitiv genomgång har utförts på körkontrollerna.

För att användaren snabbt skall kunna nå körkontrollerna, och inte som tidigare behöva leta i menyer, föreslås att en dialogruta skapas som innehåller de nödvändiga funktionerna. Användare kan låta dialogfönstret med körkontroller ligga flytande på programfönstret. De väsentligaste av dessa kontroller skall även finnas i verktygspanelen, så att körkontrollfönstret inte måste vara öppnat. En användare behöver inte kunna direktmanipulera hopp på tidsaxeln. Användaren måste först stoppa simuleringsssekvensen, för att sedan gå till meny och öppna körkontrollerna. De föreslagna förändringarna ökar effektiviteten av programmets nyttjande.

En möjlighet att backa i programmet bidrar till att användarens kontroll och frihet ökar. Användaren behöver inte vara rädd att göra fel, eftersom allt går att rätta till i efterhand. Det är nödvändigt att, när användaren backar i simuleringen, skilja på om detta är en tillbakagång på tidsaxeln för att användaren har ångrat de val som gjorts, eller om backningen är för att studera simuleringens historia, utan att några förändringar sker i beräkningarna. Önskar användaren endast att studera historien, sker inga nya beräkningar.

8.3.4 Zoom- och panoreringsfunktion

8.3.4.1 Nuvarande status

Zoomfunktioner och panoreringsfunktioner styr inställningar av systemets presentation. Användaren kan sätta den grad av zoomning som skall ske och även vilken punkt som skall vara centrum i den grafiska presentationen.

Zoomfunktionen är svår att hitta och dialogrutan ligger när den väl öppnats inte alltid synlig. Funktionerna i dialogrutan är inte alltid så intuitiva, namnen är dessutom alltför långa. Vid de minimerade användartesterna av SubTrack med användare på institutionen, var vissa problem frekvent förekommande, se *Användardeltagande i design, 7.2 och 7.3*. De oftast förekommande problemen var att något moment i zoomningen glömdes bort, eller att användaren behövde sätta programmet i fryst tillstånd så att zoomning skulle hinnas med. Detta eftersom zoomoperationer var så omständliga att utföra, att användaren inte hann med andra nödvändiga operationer.

8.3.4.2 Designförslag

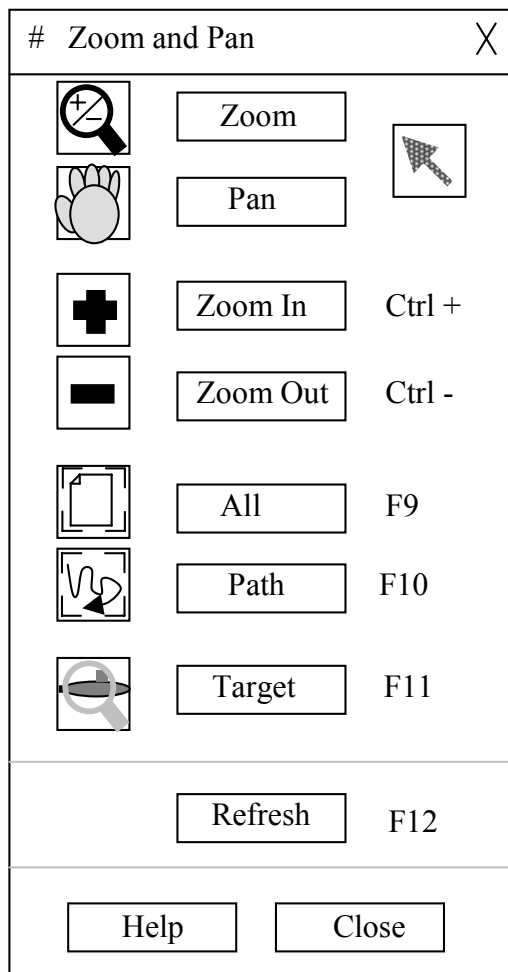
Zoomfunktionerna kan nås från menyraden eller från verktygsfältet. Från menyn öppnas en dialogruta med kommandona förstärkta av symboler och kortkommandon. Användaren väljer själv var dialogrutan passar bäst att placera. Alternativt, när verktygsfältet är synligt kan användaren snabbt och enkelt komma åt dessa funktioner, dock endast i form av en symbol.

Zoomfunktionerna kan uppdelas i två kommandon som förändrar status för musen och i fyra funktioner som utför en zoomoperation. Tillståndsförändringarna visualiseras genom att förändring sker av pekarens utseende till ett förstoringsglas, detta måste senare inaktiveras för att kunna fortsätta att köra programmet normalt, vilket görs genom att klicka på symbolen för normal pekarpil. De två tillståndsförändringarna är zoomfunktion och/eller panoreringsfunktion:

1. **”Zoom”**. Zooma ut eller in, med markör som centrum. Ikonen på verktygsfält är ett förstoringsglas med ett plustecken och en med ett minustecken inuti, ett klick i grafikytan med vänster musknapp ger inzoomning, eller utzoomning med höger musknapp. Den position där markören stod är centrum av den nya uppritningen. Markören följer samma utseende som ikonerna och om funktionen inverteras sker även så med tecknet i förstoringsglas. Funktionen erbjuder ett snabbt arbetstempo, eftersom den kräver få operationer. Denna zoomfunktion förekommer hos referensprogrammen, och ersätter SubTracks ”Zoom in (eller ”out”) by step and via click on new center”.
2. **”Pan”**. Panorera, fokus på grafiska ytan flyttas. Pekaren förändras till en hand när markören finns över grafikytan. När vänster musknapp trycks ned, kan fokus på grafikytan dras åt de håll som användaren önskar. Detta är en viktig funktion när användaren försöker följa målet, eller snabbt behöver ändra fokus för att se något som ligger utanför ritytan. Förekommer ofta i bild- och textredigeringsprogram. Denna funktion ersätter den befintliga ”Pan via click on new center”.

De fyra exekverande kommandona förändrar zoomfaktor och/eller panorering men inte musens tillstånd. Flera av dessa funktioner, särskilt nummer fem och sex, är inte allmänt förekommande i andra program, de är specialfunktioner för SubTrack.

3. **”Zoom In”** och **”Zoom Out”**. Zooma ut och in stegvis i centrum. Symbolen är helt enkelt + och -. En funktion, som med en enda operation, snabbt kan ge användaren uppritning av objekt utanför den synliga ytan. Det går snabbt att komma tillbaka till tidigare zoomfaktor, när användaren så önskar. Denna funktion finns inte i SubTrack eller Acrobat Reader, men det är önskvärt att kunna zooma utan att behöva förändra musens tillstånd. Användaren behöver inte återställa programmets tillstånd till normalt.



Figur 24: Zoomning- panorering- och återupprättningsfunktioner.

4. ”**All**”. Zooma och panorera för att visa alla bojarna och hela banan. Begreppet ”All” skall få användaren att förstå att hon skall få all grafik uppritad. Denna funktion finns i exempelvis Adobes program, i form av en ikon som visar ett ark med ett förstoringsglas på. Här har dock valts symbolen av ett ark med en ram runt, för att symbolisera att allt kommer att ritas upp. Funktionen hette tidigare ”Show path and all buoys”.
5. ”**Path**”. Zooma och panorera för att visa hela banan. Denna symbol skall vara en ”spagettislinga”. Spagettislingan avser att föra användarens tankar till en bana omsluten av en ram. Finns ingen förekommande motsvarighet i referensprogrammen. Hette tidigare ”Show the whole path”.
6. ”**Target**”. Panorera med målet som centrum. Denna symbol skall vara en ubåt med ett förstoringsglas över. Bra om användaren har zoomat och panorerat så mycket, att hon till slut tappat bort sig. Snabb, eftersom alternativet skulle vara att zooma ut tills målet hittats och sedan zooma in igen. Finns ingen vanligt förekommande motsvarighet i andra program, vilket får till följd att den inte är helt intuitiv. Hette tidigare ”Pan to target”.

Funktionen ”Refresh” uppdaterar skärmen, när ritfunktionen inte fungerar som avsetts. Denna funktion tillhör inte denna grupp, men har likheter som gör att den ingår i fönstret.

Funktionaliteten delas upp i grupper, ”Zoom” och ”Pan” som är tillståndsförändringar (användaren initierar själv zoomningen eller panoreringen), ”Zoom In” och ”Zoom Out” som exekverar zoomning utan att panorera, ”All” och ”Path” som exekverar en zoomning och en

panorering, ”Target” som exekverar en panorering och slutligen en funktion som ritar om hela den grafiska ritytan.

Tillstånd för tangentbord och mus

För att ytterligare minska de antal moment som användaren behöver utföra vid zoomning och panorering, bör kombinationer av tangentbords- och muskommandon kunna utföra en sekvens muskommandon. Denna sekvens skulle kunna vara att zooma in ett steg, sedan fortsätta boj-utläggning. I det befintliga SubTrack skall förstoringsglasat aktiveras för att förändra musens tillstånd. Därefter klickas den punkt i grafikfönstret som användaren önskar zooma. När zoomning har utförts, måste musens tillstånd ändras tillbaka till normalläge, för att bojar skall kunna läggas till eller tas bort. Sammanlagt skall tre kommandon utföras.

Om mustillstånden kan förändras genom en nedtryckning av en tangent, kan sekvensen ovan kortas ned. Användaren håller ned skiftkommandot på tangentbordet, detta aktiverar zoom-funktionen, varefter användaren klickar på den punkt som önskas att zooma. När zoomning utförts så släpper användaren skifttangenta, då mustillståndet återgår till normalläge.

De två funktionerna ”Zoom” och ”Pan” innebär en tillståndsförändring av musen, och bör implementeras med denna kombination av tangentbord och muskommando. Panorering får förslagsvis ”Alt”-knappen som tillståndsförändrare.

Detta sätt att aktivera tillståndsförändringar, måste användaren läsa sig till i hjälp, eller bli tipsad av en annan användare om. Tillståndsförändringen, visas genom att muspekaren förändras till aktuell symbol.

8.3.4.3 Diskussion och argumentation

En ritfunktion som inte direkt ingår i zoom och panoreringsfunktionerna, men ligger i gränstrakten, är omritning av hela skärmen, ”Refresh”. Denna funktion skulle i ett optimalt program vara onödig. Alla ritfunktioner skulle vara optimerade, inga fel eller brister skulle existera. Detta får ses som en utopi för en prototyp som SubTrack. Användaren får helt enkelt själv uppdatera skärmen, när programmet fallerar.

Zoomfunktionen behöver synliggöras och funktionaliteten måste omformas för att efterlikna de funktioner som går att hitta i vanliga kommersiella program, såsom Adobe Acrobat Reader och Microsofts Photo Editor. Metaforer som korta namn, tillsammans med symboler för funktionerna bidrar till en minimalistisk och överskådlig design. Om detta kombineras med hjälp i dialogfönstret, kan även ovana användare snabbt lära sig systemet.

Effektiviteten hos systemet kan ökas genom att knappar finns i ett verktygsfält, som alltid finns tillgängliga. Det skall fortfarande vara möjligt att nå funktionerna genom menyn, så att ovana användare skall kunna förstå funktionernas innebörd genom namnen i menyerna.

Pekarsymboler bidrar till att visualisera tillståndet för direktmanipulationen genom muskommandon. Återställning av mustillstånd till normaltillstånd, skall tydliggöras med en ikon som visar en normal pil. Att användaren uppmärksammas på systemets status är viktigt, för att användaren skall känna att hon har kontrollen över systemet.

Rullisten tas bort, eftersom den stjälar utrymme. Det går också snabbare att panorera genom att dra ritfönstret med panoreringsfunktionen, än att hitta rullisten och sedan manövrera denna.

8.3.5 Direktmanipulation

I WIMP står P:et för Pointers. Pekaren, i förlängningen musen, är den del av interaktionsmetoden som sköter direktmanipulationen.

Demonstratören syften uppfylls i detta fall väl av direktmanipulation, eftersom användaren snabbt med få handgrepp kan demonstrera programmets funktionalitet. Att en optimal och exakt inmatningsmetod uppnås är av sekundärt intresse.

8.3.5.1 Nuvarande status

Direktmanipulation har beskrivits som ”känslan av direkt deltagande i en värld av objekt, snarare än kommunikation med något som uppträder som en mellanhand” [Hutchins].

Direktmanipulationen i det befintliga SubTrack består av:

- Bojutläggning, bojar som läggs ut och tas bort.
- Zoom och panoreringsfunktioner, anpassning av den grafiska presentationen från programmet.

All direktmanipulation sker mot grafikfönstret.

8.3.5.2 Designförslag

Designförslaget om direktmanipulation, delas nedan upp i tre grupper.

Bojutläggning i grafikfönster

De befintliga funktionerna bibehålls med ett tillägg. Om pekaren förs över en punkt på banan, visas statusinformation i aktuellt läge efter en viss tids fördröjning. Detta sker endast om musen har ett normaltillstånd, alltså inte har ett zoom- eller panoreringstillstånd.

Informationen som kommer upp består i:

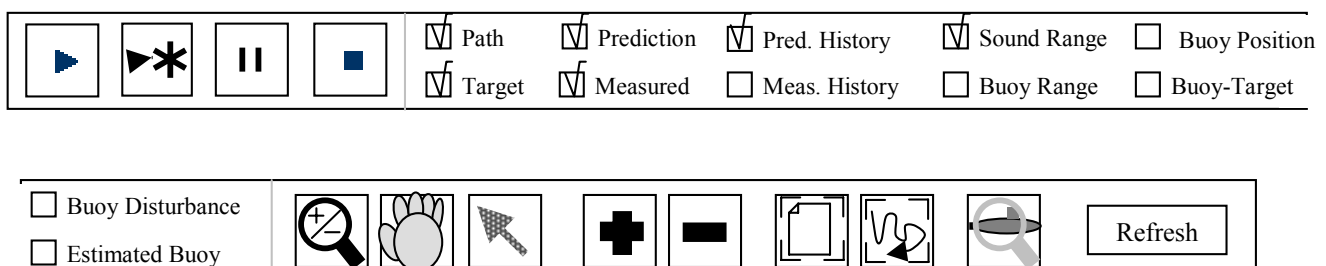
- position
- aktuellt tidssteg
- information om den aktuella beräkningsfunktionen.

Zoom- och panoreringsfunktioner i grafikfönster

Zoom och panorering finns i separat kapitel. Se *Zoom- och panoreringsfunktioner*, 8.3.4.

Verktygsfält

De operationer som kräver snabb åtkomst för användaren samlas i verktygsfältet. De är från fyra grupper av funktionalitet som verktygsfältets knappar kommer. Först kommer kontrollerna för att styra körningen av simuleringen, se *Körkontroller*, 8.3.3, näst efter detta kommer de grafikobjekt, som kan visas i grafikfönstret, se *Graphic Comp.*, 8.3.2.2, zoom- och panorering, samt den fristående funktionen som ritar om grafikytan, se *Zoom- och panoreringsfunktioner*, 8.3.4.



Figur 25: Verktygsfältet. I detta dokument är verktygsfältet uppdelat i två delar p.g.a. utrymmesskäl. Innehåller de mest väsentliga funktionerna för körning av simulering, undantaget bojutläggning vilket sker direkt med mus.

8.3.5.3 Diskussion och argumentation

Bojutläggning i grafikfönster

Bojutläggningen sker med höger och bojborttagning med vänster musknapp. Manipulationen skulle lika gärna kunna ske med samma knapp, eftersom det inte är önskvärt att lägga ut två bojar på samma position. Högerknappen kunde då användas för popupmenyer, som brukligt är på PC-plattformar.

Den nuvarande hanteringen av muskommandon i grafikfönstret behålls eftersom programmet använts av personer under en längre tid och det skulle skapa problem för dessa att ändra sitt användande. Popupmenyer aktiveras med en kombination av ”Shift”+ musklick, med höger knapp. Detta är en genväg som tyvärr inte är intuitiv. Användaren måste läsa sig till denna information, under ”Help”.

Den från expertutvärderingen föreslagna zoomfunktionen att en ruta dras ut med muspekaren, för att bilda en gräns för vad som skall ritas i grafikfönstret tas inte med i designförslaget. Skälet till detta är att funktionerna redan blivit många, och en kombination av de i designförslaget framtagna zoom- och panoreringsfunktionerna kan utföra samma uppgift.

Problemet med att programmets huvudfönster ibland tar tillbaka fokus när tillståndet är fryst läge från dialogfönster, lämnas till eventuell senare analys och implementation. En ny implementation med ett nytt programbibliotek, kanske automatiskt löser detta problem.

Zoom och panoreringsfunktioner

Diskussion finns i zoom och panoreringskapitlet, se *Zoom- och panoreringsfunktioner*, 8.3.4.

Verktygsfält

Fyra grupper ingår i verktygsfältet. Dessa har gemensamt att de används under körningen av en simulering. Under en simulering sker allt snabbt och användaren behöver snabbt ändra fokus när något intressant har dykt upp. Snabb åtkomst och full kontroll av simuleringen är i detta läge mycket viktigt. Funktionerna för körkontroll start, stopp, och frys är självskrivna. Om simuleringen snabbt kan bli fryst, kan komplicerade och tidskrävande funktioner utföras utan att användaren förlorar ett enda tidssteg av simuleringen.

De grafiska komponenterna är viktiga att kunna förändra snabbt, något objekt kan skymma ett annat eller under simuleringen kan användaren bli intresserad att närmare studera bojarnas räckvidd. Zoomfunktioner är naturligtvis med i verktygsfältet eftersom fokus förändras hela tiden under en simulering, simuleringen är i högsta grad dynamisk. Alla funktioner är inte optimerade ur rithänseende och vissa dialogrutor får grafikfältet att uppträda onormalt. Därför skall användaren enkelt kunna rita om grafikytan.

Alla funktioner i dessa grupper är inte inkluderade i verktygslådan, eftersom de inte krävs under den direkta körningen av simuleringen. Ett exempel är att köra fram till angivet tidssteg, ”Move To”. Denna funktion används efter att ”Freeze” aktiverats och simuleringen redan står still.

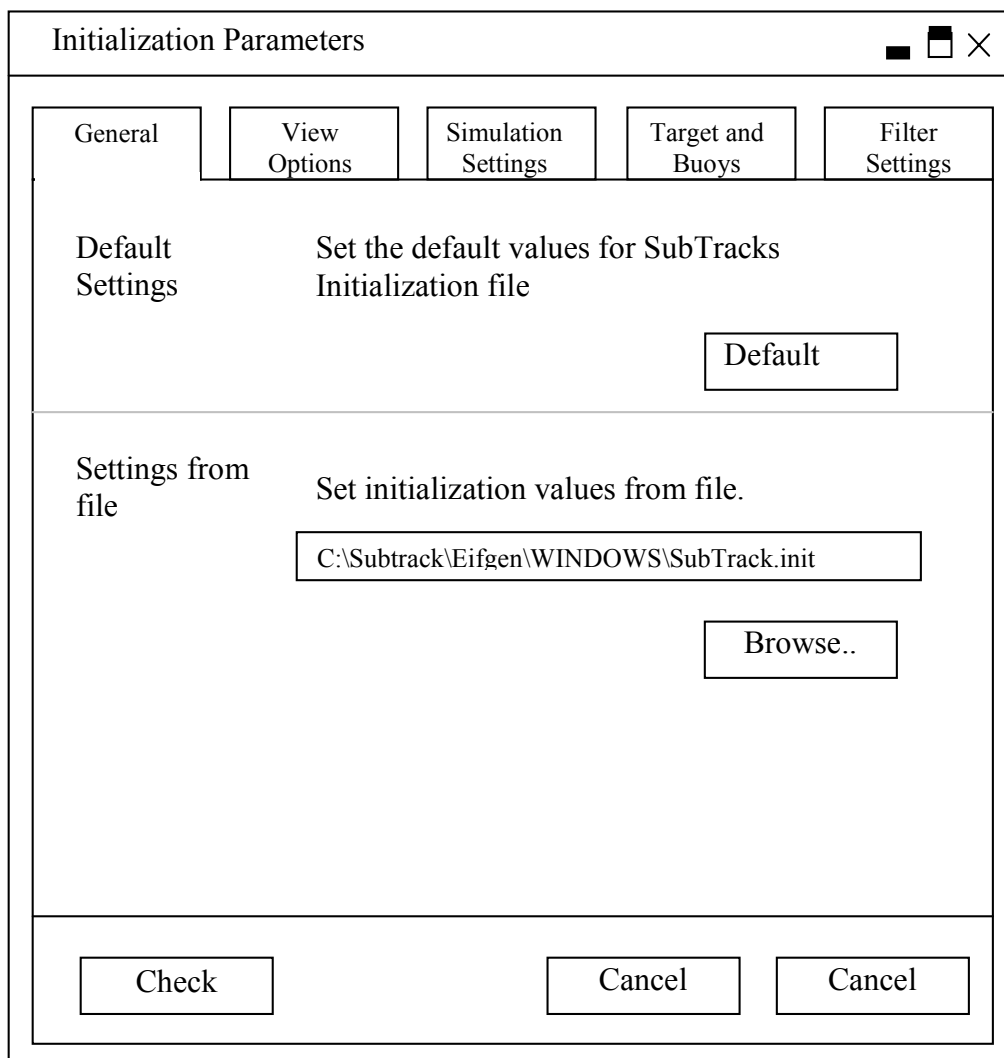
8.3.6 Initieringsparametrar

8.3.6.1 Nuvarande status

Initieringsparametrarna ligger i en fil som läses in vid start och är osynliga och opåverkbara. Vissa av dessa parametrar kan demonstratören ha intresse av att redigera, för att påvisa olika beteenden och egenskaper hos SubTrack. Andra är däremot endast läsbara, för att de är grundläggande fysikaliska egenskaper såsom ljudets hastighet, eller för att de är av alltför komplex natur för att enkelt redigeras.

8.3.6.2 Designförslag

Parametrarna som styr programmet görs åtkomliga i en dialogruta under "File" och "Setup..". Vissa av dem kommer inte att vara redigeringsbara utan endast kunna läsas medan de redigerbara parametrarna kommer att kunna redigeras via ett textfält med ett förvalt värde. Användaren kan välja att läsa in den senaste körningens parametrar, läsa en egendefinierad fil, eller läsa in en fördefinierad fil.



Figur 26: Initieringsparametrar. Generella inställningar.

Felkontrollen kommer att bestå i att alla textfält med namninmatning har ett piltextfält som innehåller en lista med förval. Parametrar med heltalsvärden kan matas in via ett +/- textfält, men med möjlighet att själv mata in värden inom givet intervall. Inga värden skall kunna sättas, som var för sig är tillåtna men som skapar en konflikt sinsemellan. Användaren kommer att kunna köra en felkontroll, genom att använda funktionen "Check". För att återhämta sig från ett eventuellt fel kan dock användaren alltid välja att ladda in programmets fördefinierade inställningar.

Nedan listas de programparametrar som finns i SubTracks initieringsfil, och som föreslås kunna ses, eller redigeras, i initieringsdialogen. Dialogfönstret finns under menyn "Settings..", i huvudmenyn "File". Parametrarna i initieringsfilen, har följande egenskaper:

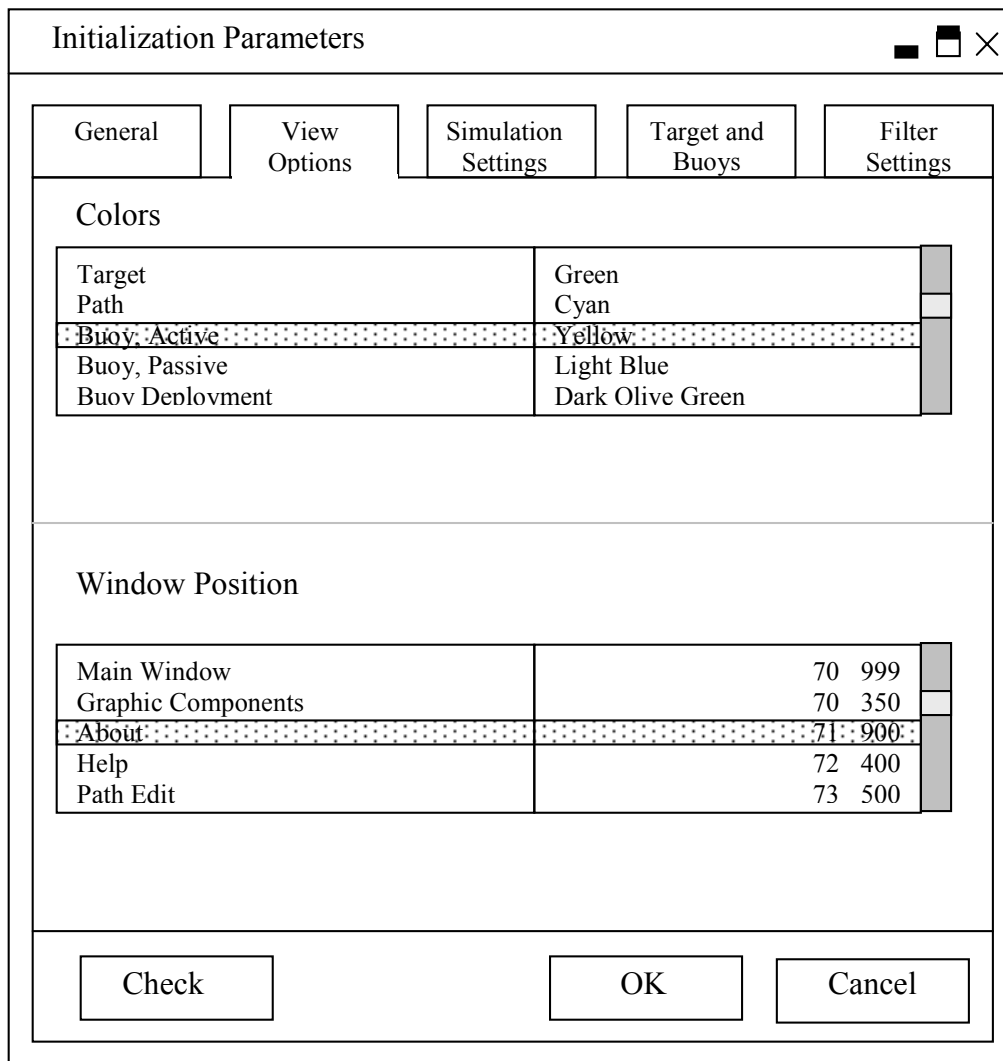
- Simulation Parameters – med vilken sannolikhet målet finns inom prediktionsellipsen("Probability") och hur långt ett tidssteg är("Simulation Step").
- Sonar EQ – *Icke redigeringsbara parametrar*, DI, TL_K, DT_REF (Direction Threshold), NL (Noise Level), SL (Source Level)
- Buoy Parameters – "Standard Deviation in m", "Standard Deviation in seconds", "Activation Time in seconds"
- Optimization Parameters – *Icke redigeringsbara parametrar*, "Measurement tolerance in m", "Fitness of the angle optimization (number of sectors)", "Tolerance for buoy position in m".
- Target Motion Parameters – *Icke redigeringsbara parametrar*, "Linear Acceleration", "Centripetal Acceleration".
- Particle Filter Parameters – antal partiklar i molnet, antal vägkandidater, antal försök vid ny bojutläggning, tid som kan förutsägas, antal sekunder för utläggningspaus, genomsnittlig gräns för uppskattning, positionsstörning för bojförslag.
- Colors – färger för programmets fönster.
- Window Position – Var dialogfönstren bör placeras.

Om simuleringen redan har startats, kommer endast ett fåtal parametrar kunna redigeras. Det kan exempelvis inte tillåtas, att ljudstyrkan på målet plötsligt ökas, utan att hastigheten har förändrats. Detta skulle innebära att målet plötsligt förändrats, från en typ av ubåt, till en annan.

Grupperingarna av initieringsparametrar, är:

- Vilken fil som skall läsas in, ("General").
- Färg och positioner på fönster, ("View Options").
- Inställningar för simuleringsssekvensen, ("Simulation Settings").
- Inställningar för målet och bojarna, ("Target and Buoys").
- Inställningar för beräkningsfunktionen, ("Filter Settings").

Endast två exempel av de fem flikarna kommer att visas. Detta eftersom de sista fyra grupperingarna är väldigt lika, och inte skulle bidra till en tydligare bild.



Figur 27: Initieringsparametrar. Inställningar för ickedynamiska grafikkomponenter.

8.3.6.3 Diskussion och argumentation

Initieringsparametrar förändrar drastiskt simuleringen. Därför är det endast erfarna demonstratörer som bör korrigerar initieringsparametrarna. Att skapa olika profiler för hur avancerad demonstratören är kan förhindra att felaktigheter uppkommer, men det är upp till demonstratören att inse sina begränsningar. Användaren har stöd av en inmatningskontroll, inga felaktiga värden kan matas in. Vidare kan användaren kontrollera att de val som gjorts inte är i konflikt med varandra genom att köra en kontroll. Detta till trots kan en felkontroll aldrig bli fullständig, varför ett antal säkra parametrar kan läsas in.

Parametrarna i dialogfönstret är redan inlästa från initieringsfilen, för att användaren snabbt skall kunna starta programmet om användaren anser att dessa variabler är korrekta. Om redigering skall göras i dagsläget, måste detta ske i initieringsfilen. Detta omöjliggör all form av felkontroll.

Det är viktigt att felkontroll skapas så att användaren känner kontroll över programmet. Om fel ändå uppkommer, skall användaren kunna återhämta sig från dessa fel.

De variabler som tagits med i initieringsfilen har diskuterats med programutvecklaren och han har angivit vilka variabler som endast är av intresse att läsa, och vilka som även bör kunna redigeras.

8.4 Presentation

SubTrack har två huvudgrupperingar som skall presentera information, den statiska presentationen samt den dynamiska presentationen. Den statiska presentationen kan vara menynamn och komponenter i statusfältet, men även delar som kan anpassas om användaren så önskar. Exempel på sådana anpassningsbara komponenter, är verktygsfältet och färger på fönster.

Den dynamiska presentationen är vad som visas i grafikfönstret. Det är komponenter som varierar beroende på simuleringens fortskridande, men också beroende på den anpassning som användaren har gjort och vilka grafiska objekt som skall visas.

8.4.1 Huvudfönster

Huvudfönstret är en del av skärmytan och innehåller programmets alla komponenter, såsom menyrad, verktygsfält, grafikfönster och statusfält.

8.4.1.1 Nuvarande status

Tjockleken på ramen stjälar värdefull rityta från grafiken. Ramens färgsättning är gräll, vilket ger ett oseriöst intryck.

8.4.1.2 Designförslag

Ramen bör göras tunn, ingen rullista skall finnas i ramen, panorering av grafikytan får istället ske genom någon av funktionerna i dialogrutan eller i verktygsfältet. Färgen på ramen bör vara ljusgrå. Namnlisten på huvudfönstret är, förutom ikoner och själva grafikfältet, det enda som bör vara färgsatt med en diskret färg, lämpligen blå. Designen skall vara estetisk och minimalistisk.

8.4.1.3 Diskussion och argumentation

Att ramen bör vara tunn och utan rullist, beror på att den skall stjäla så lite yta från grafikfältet som möjligt. Den ljusgrå färgen på ramen är vald för att färgen på ikonerna skall synas så tydligt som möjligt och att användaren, på grund av ikonens färgsättning, lätt skall uppmärksamma dem.

8.4.2 Ritfunktionen

Ritfunktionen är den uppritning av grafik, som programmet gör av simuleringsskvensen i grafikfönstret.

8.4.2.1 Nuvarande status

Den grafiska ritfunktionen i den befintliga versionen av SubTrack bygger på grafikbiblioteket Eiffel Vision från Eiffel Bench 4.5. Ett nytt utvecklingsverktyg, Eiffel Studio 5.1, innehåller ett utvecklat grafikbibliotek, Eiffel Vision 2. I detta arbete kunde ingen implementation ske av föreslagna förändringar. Om fortsatt utveckling skall ske, bör implementationen av gränssnitt ske med Eiffel Vision 2 för att kunna implementera de föreslagna förändringarna.

8.4.2.2 Designförslag

Prestandaproblem kan ha en mängd orsaker, i fallet med den grafiska ritfunktionen är det okunskap om hur den grafiska ritkomponenten skall utnyttjas. Orsaken står att finna djupt inne i koden, så den bästa lösningen kan vara den mest drastiska, att helt bygga om den grafiska ritfunktionen från början med ett verktyg med bättre grafikbibliotek.

Den utvecklare som har skapat det grafiska gränssnittet, har inte haft full kunskap om det grafikbibliotek som använts till implementationen, Eiffel Vision. Detta har berott på två orsaker, nämligen:

- Dokumentationsbrist – en dålig, näst intill obefintlig, dokumentation har erbjudits från ISE, företaget som säljer utvecklingsverktyget Eiffel Bench.
- Resursbrist – för lite tid har kunnat avsättas för att kunna botanisera i det programbibliotek som använts.

Uppritningen under en körning av SubTrack är därför inte helt optimerad, viss uppritning förstor tidigare uppritning och annan ritning är inte effektiv. Detta sänker prestanda på programmet. Nedan listas de förändringar som bör göras.

Bibehållen funktionalitet

Ritfunktionen i SubTrack innehåller endast en funktion som användaren utnyttjar direkt, utan mellanfunktioner. Denna funktion är ”Refresh”, som ritar om alla objekt i hela grafikfönstret. Funktionen finns för att återuppritningen inte kan göras tillräckligt snabbt. Det finns för många beroenden mellan olika komponenter för att kunna lösa alla problem. Därför får användaren själv kalla på en funktion som ritar om grafikfältet, när uppritningsfunktionen fallerar.

Funktionen att rita om kan nå genom menyraden under ”View”, eller dialogrutan för zoom och panorering eller i verktygsfältet. Se *Menyrad, 8.3.1, Zoom- och panoreringsfunktioner, 8.3.4, och Diskussion och argumentation (verktygsfält), 8.3.5.3.*

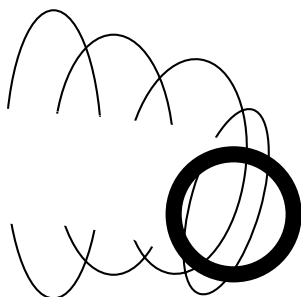
Ytterligare funktionalitet

”Sound Range” läggs till som grafisk komponent, samt ett visst tillägg i funktionalitet. Funktionaliteten visar räckvidden för det ljud som målet alstrar. Detta åskådliggörs lämpligen med en skuggad, fylld cirkel och inte med en ring. Om denna funktion används istället för ”Buoy Range” ritas endast en cirkel istället för en för varje boj. Förslaget bidrar till minimalistisk design.

Koordinering mellan ritobjekt

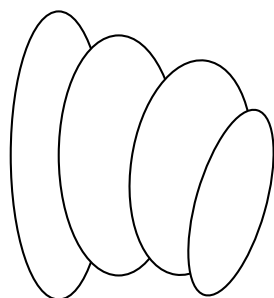
De grafikobjekt som ritas koordineras inte med varandra. Vissa grafikobjekt ritas över andra, eftersom vissa ritas varje tidssteg och andra endast ritas då de skapas. Omritning kan också ske när användaren byter de grafiska objekt som skall visas, eller när återuppritning begärs.

- ”Target” suddar ut ” Measurement ellipse history” och ” Prediction ellipse history”. Detta sker eftersom målet ritas vid varje tidssteg, medan historieellipser bara ritas om en gång. Några av dessa ellipshistorier måste ritas om, de som ligger under ”Target”.



Figur 28: Målets grafikobjekt suddar ut de tidigare ritade ellipserna.

- För att inte rita mer än nödvändigt bör inga ellipser av samma typ, ” Measurement ellipse history” eller ” Prediction ellipse history”, rita över den förutvarande. Detta innebär att när en ny ellips ritas, kontrolleras om den korsar någon av de tidigare. Om så är fallet, räknas den tidigare om och i den länkade listan som beskriver ellipsernas historia, byts den gamla cirkelbågen ut mot en korrigerad. Detta skulle innebära att den bakre delen av ellipsen ritas som en båge. Förslaget skulle göra ellipshistorikerna tydligare för betraktaren. Dock skulle troligtvis ingen prestandavinst göras, eftersom den vinst som görs av att rita mindre, tas ut av att ytterligare en beräkning måste göras.



Figur 29: Mål och prediktionsellipser som endast ritar väsentlig information.

- Historieellipserna ritas över av: mål, boj-mållinjen, beräknad bojposition, bojens avvikelse och bojräckvidd. Att dessa grafikkomponenter ständigt ritas om, skapar svårigheter att se de viktigare ellipserna. Det bidrar också till prestandaproblem, se *Designförslag (prestanda)*, 8.4.2.2
- Om ett grafiskt objekt väljs bort under körning, skall de objekt som ligger under detta ritas om. Om ”Path” väljs bort, kommer den att ritas med bakgrundsfärg, vilket får till följd att alla ellipser som den korsar, bör ritas om. Förslaget innebär en mer estetisk design.
- Om användaren har valt att se hela banan, kan historieellipserna vara så små att de knappt syns. När ellipserna blir för små räcker det att endast varannan, eller kanske var fjärde, ritas ut. Synbarheten av systemets status ökar när presentationen blir tydligare.

Prestanda

De föreslagna förändringarna nedan, kommer förhoppningsvis ha en positiv inverkan på programmets prestanda:

- Endast de grafikobjekt, eller delar av dessa, som finns synliga i grafikfönstret skall ritas om. Inget som inte är synligt skall ritas om. Skälet till att detta sker idag, kan vara att utvecklaren inte har tillräcklig kunskap om grafikbiblioteket, eller brister i detta bibliotek.
- Linjen boj-mål och cirklarna för bojpositionsstörning respektive höravstånd, ritas ut varje tidssteg. Detta leder till onödigt slöseri av beräkningsresurser. Dessa grafikkomponenter skall endast ritas ut om de har förstörts av något fönster eller av ett annat grafikobjekt.
- Boj-mållinje ritas ut med bakgrundsfärg varje tidssteg, trots att användaren valt att de inte skall visas. Uppritning av ett grafikobjekt med bakgrundsfärg skall endast ske om användaren begär att ett objekt skall döljas. Det är resursslöseri med grafikprestanda att rita upp linjer med bakgrundsfärg om det inte finns ett syfte. Detta fel beror på undermålig implementationen och inte på det dåliga grafikbiblioteket.

Dialogfönsters inverkan

Den negativa inverkan, som dialogfönster har på grafikfönstret, beror antingen på att utvecklaren har begränsad kunskap om programbibliotekets funktioner, eller på brister i grafikbiblioteket. Dialogfönstrens brister:

- När en dialogruta flyttas skall alla eventuella grafikobjekt som blir synliga ritas om. Detta medför att ellipser suddas ut, eftersom de inte ritas om i varje tidssteg.
- Ett informationsfönster för dialogfönstrets x och y pixelkoordinater visas mitt i ritytan varje gång som en dialogruta flyttas. Funktionen fyller inget behov och bör tas bort.

Återuppritningsfunktion

Omritningen av hela skärmen sköts av funktionen ”Refresh”. Denna funktion skulle i ett optimalt program vara onödig. Alla ritfunktioner skulle vara optimerade, inga fel eller brister skulle existera. Detta får ses som en utopi, som den prototyp som SubTrack är, aldrig kommer att komma i närheten av. Användaren får helt enkelt själv uppdatera skärmen om programmet fallerar.

8.4.2.3 Diskussion och argumentation

Det är inte lätt att skapa ett grafiskt gränssnitt med dåliga verktyg. Eiffel är ett programspråk som har liten spridning i programmeringsvärlden. Detta resulterar naturligt i små resurser att skapa bra verktyg. De utvecklare som varit ansvariga för implementationen av gränssnittet, har inte haft ett lätt arbete att, med dålig dokumentation, försöka lära sig använda grafikbiblioteket. Det grafikbibliotek som använts, Eiffel Vision, är gammalt och erbjuder inte de funktioner som ett motsvarande modernt bibliotek i Java eller C++ hade gjort.

Det är, trots allt, inte enbart problem med grafikbiblioteket som gjort att grafikfönstret inte fungerar särskilt väl. Om bättre planering sker av vilka objekt som ritas upp i vilken följd, hade prestanda på programmet ökat väsentligt. I vilken följd och hur ofta objekten ritas upp, beror på vilka grafikkomponent som är viktigast. Den grafikkomponent som ritas upp senast kommer att skymma andra objekt, varför den kommer att verka ligga överst.

1. Bojar, estimering och prediktion för målet bör ligga ovan allt, därför skall de ritas om vid varje tidssteg efter alla andra komponenter har ritats.
2. Bana och mål skall ligga ovanför ellipser och räckvidder. Komponenterna skall alltså ritas om före grupp 1.
3. Ellipserna ritas lika ofta som bojar och målets ljud. De skall ritas när nummer 4 redan har ritats, så att de verkar ligga ovanpå. Ellipserna behöver endast ritas om vid enstaka tillfällen, som när annan grafik förstört dem, användaren aktiverar återuppritning, eller när användaren inaktiverar denna grafiska komponent.
4. Boj- och målljudsräckvidd skall verka ligga längst ned för användaren. De skall alltså ritas tidigast. Uppritningstillfällen är lika som hos grupp 3.

8.4.3 Statusinformation

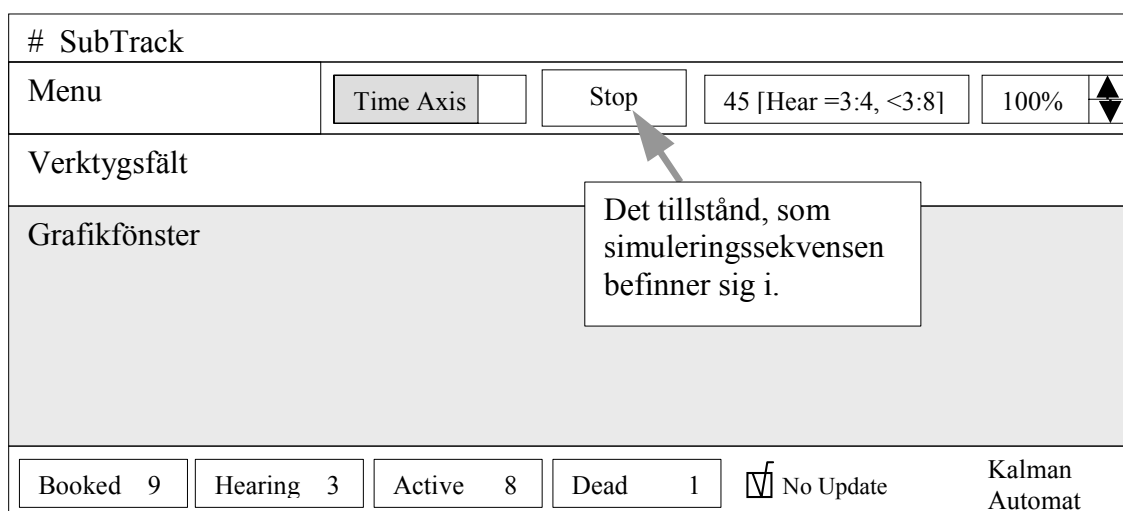
Statusinformation är information från systemet som användaren kan komma att behöva. Statusen kan förändras av användaren själv, eller vara en följd av simuleringsssekvensen.

8.4.3.1 Nuvarande status

Statusinformationen återfinns på två ställen, i statusfönstret till höger om menyraden och i menyval "Buoys", under "View". Statusfält innehåller information om aktuellt tidssteg, programmets tillstånd, ett mått på vilken kvalitet som finns och har funnits för målföljningen, slutligen ett mått på grafikfönstrets förstöringsgrad. Menyvalet "Buoys" innehåller information om antal tillgängliga bojar, antal använda, antal utplacerade, antal planerade men inte utplacerade, antal hörande och antal inaktiva.

8.4.3.2 Designförslag

Statusinformationen föreslås bestå av två statusfält, ett till höger om menyraden, samt ett längst ned i programfönstret.



Figur 30: Statusrader. Den övre, till höger om menyn. Den undre, längst ned i programfönstret.

Det övre statusfältet föreslås innehålla information som anses vara en väsentlig del av simuleringsssekvensen och är simuleringsssekvensens tillstånd och bojkvalité. Tidssteget föreslås förändrat till en liggande stapel, som visar hur lång tid som förflutit av simuleringen, och om användaren klickar på dialogrutan, förändras visningen till timmar, minuter och sekunder. Grafikfönstrets zoomfaktor ersätts med en zoomfaktor i procent av skärmyta, som användaren interaktivt kan ställa om till önskad zoomfaktor.

Den information som dialogfönstret "Buoys" innehåller, skall placeras längst ned i programmet. Innehållet föreslås vara information om bojar, antal utplacerade, antal hörande, antal inaktiva och antal planerade utläggningar av bojar. Dock skall inte information visas om antalet tillgängliga eller antalet använda.

Utvecklare A använde dialogfönstret "Buoys" för att under körning jämföra detta dialogfönster med statusfältets information. Det är därför olämpligt att ta bort denna möjliga tillämpning. Tillämpningen är beroende av att ingen uppdatering sker utan att användaren klickar på knappen för uppdatering. En förändring bör ske av funktionens namn, som bör ändras till "No Update". Denna namngivning är förhoppningsvis tydlig och användaren förstår att hon kan hindra dialogfönstret från att uppdateras.

Alla objekt i statusfältet bör ha en popuphjälp som visar objektnamnet när användaren för muspekaren över objektet. Vidare hjälp och information om statusfält och dess objekt skall användaren kunna hitta i menyval "User Manual", under "Help".

Utseendet runt objekten i statusfältet skulle hålla ihop bättre om en ram omslöt dessa. Det skulle även ge ett tydligare utseende.

8.4.3.3 Diskussion

Informationen i statusfälten syftar till att ge användaren möjlighet att hålla sig underrättad om systemets status. Användaren skall inte behöva belasta sitt korttidsminne med information, allt skall finnas tillgängligt när användaren behöver det. Det är viktigt att användaren får informationen presenterad för sig enligt den presentationsmetod som lämpar sig bäst. Exempel är grafik (tårtbitar och staplar), numerisk och textinformation. Statusinformation i form av text är oftast namn på tillstånd.

Tillstånden för simuleringssekvensen skulle ha kunnat ges symboler, men begrepp som "Stop" och "Run" är invanda. Läsning av text är en automatiserad funktion för det mänskliga sinnet och är nästan lika lätt att uppfatta som symboler.

8.4.4 Hjälp och dokumentation

8.4.4.1 Nuvarande status

Hjälpen är inte grundligt genomtänkt ur en användares perspektiv. Den har uppkommit ad hoc under programmets utveckling och ur utvecklarens synvinkel. All hjälp är inte samlad i huvudmenyn hjälp och det är svårt att urskilja vad som är hjälp till en användare och vad som är information till en utvecklare. Hjälp skall vara enkel att söka och skall fokusera på användarens uppgift. Dokumentation skall lista konkreta steg som skall göras, kort och koncist.

8.4.4.2 Designförslag

Utseende och placering av hjälpfunktioner föreslås i designförslaget. Dock kommer ingen hjälp att skrivas om, detta lämnas för senare implementation. Hjälp föreslås uppdelad i fem kategorier:

- En beskrivning av hur SubTrack fungerar, hur det hanteras och hur resultaten presenteras. Det bör även finnas referenser till utförligare dokumentation.
- En beskrivning för en demonstratör om lämpligt tillvägagångssätt under "Help".
- Information om hur programmet arbetar på den specifika plattformen, hur katalogstruktur ser ut och interaktionsverktyg såsom mus och tangentbord. Dess syfte är främst att felsökning underlättas, om programmet inte uppträder normalt.
- Hjälp i varje dialogfönster. Hjälp bör finnas tillgänglig direkt i detta fönster genom en knapp.
- Popuphjälp, för alla knappar och statusfält.

Hjälpen i de tre första kategorierna finns under huvudmenyn "Help". Alla dessa öppnar dialogfönster, se *Dialogfönster*, 8.3.2.

8.4.4.3 Diskussion och argumentation

Hjälp skall ses helt ur användarens perspektiv och användaren är vanligen mer intresserad av information om programmets funktionalitet, än av hur systemets komponenter fungerar. Detta är anledningen till att hjälp delats in i ovanstående grupper.

I moderna program finns ofta en hjälp som innehåller flikar med indexerad sökning och en innehållslista i form av ett träd. En såpass avancerad hjälp kan vara aktuell för en användarroll som kursdeltagare. Demonstratören skall dock ha en så pass gedigen kunskap om SubTrack, att en enklare variant av hjälp räcker.

I statusfältet skall en popuphjälp dyka upp, när musmarkören förs över objektet. Popuphjälp skall vara av en enkel art, tillräcklig för att friska upp minnet hos en användare som en gång har besittit kunskapen. Den oerfarna demonstratören skall med hjälp av denna information hitta den saknade kunskapen under huvudhjälpen.

8.5 Startsekvens

Detta är en ny funktion i SubTrack som skall sköta inställningarna av programmet för en körning av en simulering.

8.5.1 Nuvarande status

Vid start väljer användaren inställningar för att köra en simulering, ungefär fem val görs, som varierar beroende på vad demonstratören vill visa upp av SubTracks funktionalitet. Dessa val görs i den befintliga versionen i fem delmoment, och momenten upprepas gång på gång innan varje körning startas. Ingen återkoppling sker till användaren av vilka moment som gjorts och vad som då valdes.

När användaren använder det vanliga gränssnittet (WIMP) för att mata in simuleringens parametrar och ladda dess bana, sker detta via flera interaktiva fönster. Ett alternativ till detta förfaringssätt är att starta SubTrack från en kommandoprompt. Om användaren arbetar i kommandofönster, så kan alla de nödvändiga parametrarna som behövs vid en simuleringstart matas in som argument på kommandoraden. Även i detta fall kommer en startdialog att öppnas, så att användaren får en tydlig återkoppling av vad som matats in och om någon av dessa skulle ha visat sig vara fel, kan användaren förändra valen.

8.5.2 Designförslag

Vid start av programmet, öppnas ett dialogfönster som innehåller de fem inställningar som kan vara aktuella för användaren att förändra. Inställningarna är:

- Vilken beräkningsmetod som skall användas, Kalman- eller Partikelfilter.
- Vilken bojutläggningsmetod som skall användas, manuell (användaren) eller automatisk (programmet).
- Vilka grafiska objekt som skall visas i ritytan.
- Vilken bana, beskrivning av målets väg, som skall laddas.
- Vilken zoomfaktor som simuleringen har vid start av simuleringen.

Alternativet med kommandoprompt, då användaren skriver in kommando och argument vid start, kan utan konflikt kombineras med ett WIMP-alternativ och på så sätt användas som ett komplement. Detta erbjuder användare, vana vid Unix- eller DOS-miljö, ett snabbt och effektivt sätt för att starta SubTrack.

Dialogfönstret utformas för att hantera de vanligaste inställningar som en användare kan vilja justera innan start. Det kan finnas demonstratörer som vill göra ytterligare inställningar, de väljer då lämpligen "OK-Freeze" för att starta. Detta försätter programmet i frysläge innan användaren, när hon eller han känner det lämpligt, kan starta simuleringen. Användare som anser att alla inställningar är gjorda använder "OK-Run", varvid programmet omedelbart startar simuleringen.

Förval av värden lagras i filen "filnamn.su", där ändelsen står för "Start Up". Det finns minst två filer, en från den tidigare körning, och en skrivskyddad som programutvecklaren har skapat. Användare kan även skapa egna filer, optimerade för demonstrationstillfället. I detta fall rekommenderas dock handledning av en erfaren utvecklare eller annan användare. Separering mellan dessa förvalda värdena sker genom att dialogrutan delas i flikar och startas genom knapparna "OK-Freeze" eller "OK-Run", beroende på vilket tillstånd som önskas till simuleringen.

Figur 31: Startdialog.

8.5.3 Diskussion och argumentation

Uppreningen av moment i starten före en simuleringssekvens är en källa till misstag i hanteringen av programmet. I de minimerade användartesterna uppdagades några fel som beror på onödigt komplicerad inmatning. Se *Användardeltagande i design, användare A, 7.2 och användare B, 7.3*. Något av momenten kan med lätthet missas, vilket leder till att körningen får ett önskat förlopp och resultat.

Tre alternativ för att starta en körning valdes som kandidater för WIMP-lösning och ett alternativ baserat på kommandobaserad interaktion. De förslag som föregick designförslaget var:

Användarkontroll genom kommandoprompt

- Start med argument (-p för val av bana), efter kommandot run_subtrack, som startat SubTrack från kommandofönster.

Användarkontroll genom WIMP

1. **Start med förvalsvärden.** Förvalsvärden initierar programmet och denna information visas i statusfältet. Om dessa värden skall förändras av användaren, sker detta innan simuleringen startas. För att förändra värdena öppnas en inmatningsdialog i menyraden, alternativt görs en knapptryckning i verktygsfältet.
2. **Formulär.** Vid start öppnas automatiskt ett formulär. I detta finns förvalsvärdena, som kan ändras innan användaren trycker på "OK-Start" eller "OK-Freeze" knappen för start av simuleringen. Under körning finns simuleringsinställningarna i statusfältet. Fördelen med en startdialog är att statusfältets utformning kan göras mer kryptisk, i form av ikoner med en möjlighet att få mera information i en popuphjälp. Se Hjälp och dokumentation, 8.4.4.
3. **Frågestyrd dialog.** När programmet startas, körs en frågestyrd dialog som hjälper användaren att sätta de parametrar som behövs vid start. Förvalsvärden finns satta i varje steg, så om ingen förändring skall göras, så klickar användaren på nästa. När alla val har gjorts, så visas startinställningarna i statusfältet på samma sätt som i alternativ 2. Se *Statusinformation*, 8.4.3. Metoden med en frågestyrd dialog ("wizard") kan anses som omständlig, men är något mer pedagogisk. Både användaren, och de som förevisas demonstrationen kan lättare förstå de olika kategorier som SubTrack beror av.

Den mest erfarna demonstratören (utvecklare C, som även är användare A) var av åsikten att det första alternativet var bäst. De förvalda värdena för start, skall vara de värden som användes vid den senaste körningen. Utvecklare A tyckte att statusfält är i vägen, men kunde tänka sig ett kryptiskt statusfält som tar lite plats från ritytan. Möjligheten att sätta parametrar vid start från en kommandoprompt fick absolut inte försvinna. Projektledaren propagerade för alternativet med formulär, och föredrog ett statusfält som tog så lite yta från det grafiska fältet som möjligt. Ytterligare information skulle dock vara tillgänglig om komponenterna i statusfältet, kanske genom popuphjälp.

Zoomfaktorns resultat är obekant för användaren de första gångerna, men efter en tids nyttjande får användaren erfarenhet av vilken inställning som hon eller han föredrar. Det går även att justera zoomfaktorn med de vanliga zoom- och panoreringsfunktionerna, efter att simuleringen har startat.

Förslagen till en startdialog diskuterades med personal på avdelningen som alla har varit demonstratörer vid något tillfälle. De är också möjligt att de i fortsättningen använder SubTrack. Det är därför viktigt att låta dem delta i designprocessen. Därmed skapas acceptans för de förändringar som föreslås.

8.6 Prestanda

Prestanda har inte kunnat förbättras nämnvärt, endast till viss del genom en optimering av uppritningen. Källan till problemet är att SubTrack är en prototyp och inte var avsett att bli stort vid starten av projektet. En uppdelning av programmet i skilda trådar eller processer hade kunnat lösa de flesta prestandaproblem, dock inte den beräkningsfunktion som planerar utläggning av bojar när Partikelfiltret används. Se *Designförslag (prestanda)*, 8.4.2.2.

9 Sammanfattning

9.1 Mål

Ett designförslag skall utarbetas för SubTrack genom att omstrukturera och förändra interaktionen. Designförslaget skall syfta till att förbättra användarvänligheten för användarrollen som demonstratör. Lärbarhet, flexibilitet och ett robust system är de övergripande mål som SubTrack skall försöka nå.

Området människa-datorinteraktion och användarvänlighet för SubTrack har tidigare inte prioriterats. Detta eftersom utvecklingen har skett evolutionärt, nya mål har skapats när resultaten av de tidigare målen har analyserats.

Arbetet resulterar i ett designförslag med beskrivningar av de olika delmomenten och delkomponenterna. Ingen implementation av designförslaget utfördes, varken i SubTrack eller som en enklare gränssnittsprototyp som simulerar SubTracks funktionalitet.

Rapporten syftar till att fungera som ett stöd vid fortsatt utveckling eller som ett exempel och stöd i arbetet med att göra en liknande datorbaserad prototyp användarvänlig. Framst avses det arbete som sker inom avdelningen Ledningssystem.

9.2 Lösningssmetod

För att göra programmet SubTrack mer användarvänligt för rollen som en demonstratör utfördes en expertutvärdering i form av heuristisk utvärdering och kognitiv genomgång. Vidare genomfördes användardeltagande för att ta vara på användarnas domänkunskap och för att skapa en uppgiftsanalys för en demonstratör.

Användarna besitter mycket kunskap om hur programmet bör fungera. Det är därför viktigt att skapa en känsla av delaktighet hos användarna, som förhoppningsvis även leder till en större acceptans för den förändrade designen av gränssnittet. I användardeltagandet identifierades uppgiftsbeskrivningen för hur demonstratören antas arbeta. Användardeltagandet innebar också att användartester gjordes på det befintliga systemet för att användarna skulle kunna identifiera brister som inte uppdagats i expertutvärderingen.

Både heuristisk utvärdering och kognitiv genomgång användes för att utföra expertutvärderingen. Det krävs en utförlig beskrivning av systemet för att kunna utföra den kognitiva genomgången, detta skedde med textbeskrivning och användningsfallsdiagram.

Resultatet från den kognitiva genomgången och från användartesterna kompletterade den heuristiska utvärderingen. Dessa brister sammanfattades i en expertutvärdering. Vilken brist som uppdagades i vilket moment beskrivs inte, det anses viktigare att användaren får en samlad bild av bristerna i SubTrack. Dubbel identifiering av brister förvirrar läsaren.

Att utföra den kognitiva genomgången var inte planerat. Arbetet avsågs från början att leda till en implementation av en ny version av SubTrack. Detta var dock inte genomförbart p.g.a. svårigheter med det verktyg som skulle användas för implementationen, Eiffel Studio, och med portering av SubTrack till Microsoft Windowsmiljö. Det kom samtidigt besked från projektledaren, att utvecklingen av SubTrack skulle läggas ned. Detta minskade de tillgängliga resurserna, bl.a. möjligheten att uppta tid för den andra omgången av användartesterna med personer anställda på institutionen. Att genomföra användartester med en enklare gränssnittsprototyp som simulerade SubTracks funktionalitet var inte heller lämpligt, då det på institutionen inte fanns någon tillgänglig kunskap om dessa verktyg.

Det var för att uppväga att den andra omgången av användartester inte kunde utföras som expertutvärderingen utökades med den kognitiva genomgången. Detta ansågs vara den resurssnålaste lösningen för att fortfarande hålla en hög kvalitet på designförslaget.

9.3 Resultat

SubTracks brister ur en demonstratörs synvinkel är sammanställda i expertutvärderingen. Sammanfattningen av resultatet finns i två versioner och två olika underkapitel. Den första ger en beskrivning enligt de heuristiska punkterna och lämpar sig för läsare som inte har kunskap om hur SubTrack ser ut och fungerar. Den andra sammanfattningen är gjord enligt den indelning som återfinns i rapportens kapitel 2, 6 och 8. Den andra resultatsammanfattningen lämpar sig bäst för personer som vill ha mer detaljerad kunskap om designförslaget.

Förslagen till förändringar har inte provats praktiskt med användartester, eftersom ingen implementation har gjorts. Den kognitiva genomgången baseras på en metodisk genomgång av programmet, varför resultatet blir en likvärdig förbättring av designförslaget.

9.3.1 Resultat enligt de heuristiska punkterna

De heuristiska punkterna skapades av Jacob Nielsen och Rolf Morlich, 4.3.3.1.2, för att underlätta utvärderingen av ett interaktivt system. De tar inte hänsyn till prestandakriterier. Nedan finns en sammanfattning av förändringar som föreslås enligt de heuristiska punkterna:

- *Synlighet av systemets status* är beroende på hur uppritningen sker. Denna förändrades främst genom en systematisering av uppritningen av de grafiska komponenterna, men också genom att statusinformationen föreslås bli tydligare i programmet.
- *Kopplingen mellan systemet och den verkliga världen* var redan bra, dock har vissa justeringar gjorts beträffande namn och symboler för körkontrollerna.
- *Användarkontroll och frihet* önskas förbättrade, genom att en möjlighet att backa i tidsstegen för simuleringen läggs in i programmet. Vidare kan direktmanipulationen indirekt påverka användarkontrollen. Detta genom att direktmanipulation tillåter användaren att omedelbart korrigera ett kommando som inte givit avsedd effekt, med dess motsats. Startdialogen bidrar till att användaren får ökad kontroll.
- *Konsekvens och standarder* har ökat, framför allt genom att namngivning på menyval förändras och att dialogfönstren föreslås få en enhetlig och konsekvent utformning.
- *Förhindrande av fel* resulterar flera av de föreslagna förändringarna i. Exempel på sådana är: symboler till körkontroller, möjlighet att i en kontrollerad form ändra variabler i filen för initieringsparametrar, men främst genom skapandet av en startdialog.
- *Igenkänning, snarare än memorering*, förbättras om en initieringsdialog skapas som låter användaren sköta alla inställningar före starten av simuleringen, i ett och samma steg. Vidare minskar kraven på memorering om statusfältet fylls med de variabler som förändras under simuleringens gång.
- *Flexibilitet och effektivitet* ökar om användaren erbjuds snabbare metoder för kontroll av simulering och presentation, genom direktmanipulation. Ingen större grad av personlig anpassning erbjuds, dock kan användaren ställa in färger och placeringar av dialogfönster.
- *Estetisk och minimalistisk design* är ett subjektivt kriterium, som till stor del beror av den miljö där produkten skall fungera. Vissa förändringar har föreslagits, bl.a. förändring i färgval samt att storleken på dialogfönster minskas.
- *Att användaren skall inse, diagnostisera och återhämta sig från fel* har inte prioriterats, varken under tidigare utveckling, eller i designförslaget. De tänkbara demonstratörerna bör ha någon form av erfarenhet genom arbete med SubTrack och får förlita sig till hjälp och dokumentation när problem uppkommer.
- *Hjälp och dokumentation* har föreslagits fokusera mer på användarens uppgifter än, som idag, på den funktionalitet eller på de komponenter som systemet består av. En hjälp som återfinns direkt i dialogrutan eller i anslutning till den grafiska komponenten är önskvärd.

9.3.2 Resultat enligt rapportens kapitelindelning

Indelningen av SubTrack har tagit fasta på Normans modell om interaktion, 4.2, uppdelning sker mellan användarkontroll och systemets presentation. Detta är endast interaktion på användarens sida, eftersom ingen implementation utfördes. Vissa delar ligger utanför interaktionen i Normans modell, som funktionaliteten och prestanda. Startsekvens är en så pass viktig del i användarkontrollen, att den har fått ett eget kapitel.

SubTracks funktionalitet:

Det har endast föreslagits smärre förändringar i funktionaliteten, främst att användaren skall kunna backa i tidsstegen i simuleringsssekvensen för att öka användarkontrollen.

Användarkontroll:

- Menyerna har omformats, både vad beträffar huvud- och undermenyer. Namn har bytts, omgrupperingar har gjorts och layouten har förändrats, för att förhindra fel och öka kopplingen till den verkliga världen.
- Dialogfönster har grupperats, givits ett enhetligt utseende och en konsekvent namngivning.
- Körkontroll har skapats för att öka effektiviteten för användarens kontroll av simuleringen.
- Zoom- och panoreringsfunktioner har förändrats för att förebygga fel och effektivisera användarens handhavande.
- Direktmanipulation är ett lämpligt sätt för en demonstratör att arbeta. Detta har utvidgats och förändrats för att öka effektivitet och känslan av användarkontroll. Direktmanipulation är inte fristående, utan är förbundet med zoom- och panoreringsfunktioner, verktygsfältet, samt bojutläggning.
- Initieringsparametrar har gjorts tillgängliga för manipulation av en användare. Detta ökar användarens känsla av kontroll samt att det, om felkontroll finns, förebygger fel.

Presentation:

- Designen av huvudfönstret har förändrats för att uppnå en estetisk och minimalistisk design.
- Uppritning av grafikkomponenter har systematiserats för att öka synbarhet av systemets status, samt för att minska den prestandaförlust som uppkommer under ritsekvensen.
- Statusinformationen har strukturerats om och har gjorts om till en komponent av huvudprogrammet, istället för att vara en dialogruta.
- Hjälpen har strukturerats om för att fokusera på användarens uppgift.

Startsekvens:

En startdialog har skapats så att en användare skall finna all den information som är nödvändig för att köra en simulering, på samma ställe. Detta bidrar till att öka användarens känsla av kontroll, förhindra fel, och minska belastning av korttidsminnet.

Prestanda:

Prestanda har inte kunnat förbättras nämnvärt, endast till viss del genom en optimering av uppritningen. Källan till problemet är att SubTrack är en prototyp och inte var avsett att nå denna storlek när projektet startades. Vissa funktioner tar mycket datorkraft eftersom det är tunga beräkningar. En uppdelning av programmet i skilda trådar eller processer hade kunnat lösa de flesta interaktionsproblem och tillåtit användaren att ha en ständig dialog med systemet, men inte gjort någon skillnad för själva beräkningsprocessen.

10 Referenser

Referenser nedan är uppdelade i statistiska referenser som kommer att finnas kvar, samt referenser som till internetsidor, som endast kommer att finnas tillgång till så länge som sidans ansvarige så önskar.

[Allwood] Allwood, C. M. (1998). "Människa-datorinteraktion. Ett psykologiskt perspektiv". Lund, Sverige: Studentlitteratur

[Card, Moran, Newell_2] "The keystroke-level model for user performances with interactive systems", S. K. Card, T. P. Moran, A Newell, 1980

[Card, Moran, Newell_1] "the Psychology of Human Computer Interaction", S. K. Card, T. P. Moran, A Newell, 1983

[Dix, Finlay, Abowd, Beale] Alan Dix, Janet Finlay, Gregory Abowd och Russel Beale, 1997 "Human-Computer Interaction", andra utgåvan, Prentice Hall.

[Fowler] Martin Fowler och Kendall Scott, 1999 "UML distilled", andra utgåvan, Addison Wesley.

[Hutchins] Hutchins et al, "User Centered System Design: New Perspectives on Human-Computer Interaction", 1986, Lawrance Earlbaum Associates, Hillsdale.

[JS 97] Johansson Kristian, Svensson Per 1997 "Submarine Tracking by Means of Passive Sonobuoys - Position estimation and buoy deployment planning", , FOA-R--97-00440-505--SE Methodology report.

[Mills] Mills, H. D., O'Neill, D., Linger, R. C., Dyer, M., och Quinnan, R. E. (1980). "The Management of Software Engineering". IBM sys.

[Shneiderman] Ben Shneiderman, 1998 "Designing the User Interface", tredje utgåvan, Addison Wesley Longman.

[Sommeville] " Ian Sommerville, 1995 Software Engineering", femte upplagan, Addison Wesley.

[WFJS 96] Williams Mark, Svensson Per, Jöred Karsten 1996 "Multi-sensor tracking by particle-filter-based predictiv sensor deployment", Journal of Information

Internetlänkar

[Linköpings Universitet] kursansvarig är Mikael Kindborg, titel " Hur uppnår man användbarhet?" publicerat 1998-09-10

[http://www.ida.liu.se/~mikki/previouscourses/tddb34_ht98/forelasning2/sld018.htm] hämtat 2002-02-20

[Umeå Universitet] Lars Andersson, titel "Metoder och tekniker..." publiceringsdatum okänt [www.informatik.umu.se/gru/kurssidor/2001/vt/infa06/desi/ske/Methoderteknikerdesignstod.ppt] hämtat 2002-02-20

11 Ordlista och akronymer

Ordlistan ger förklaringar till ovanliga ord, specialtermer eller ord som kan ha flera betydelser. Ett avsnitt med akronymer finns som ett underkapitel.

11.1 Ordlista

Ad hoc – för detta särskilda ändamål (tillfälligt påkommen lösning).

Akronym – förkortning som består av initialer och uppfattas som ett namn eller ord.

Användarstudier – intervjuer eller observationer, där en användare nyttjar en prototyp eller befintlig produkt. Användarens handhavande och uppfattning om systemet, utgör ett mått på systemets användbarhet.

Användningsfall – (jfr. engelskans use case) Användningsfall är en inventering av de behov och krav som olika användare (kunder, operatörer, reparatörer, andra anslutna apparater) har på ett system. Det är den första fasen i systemutveckling, enligt det synsätt som lanserades av Ivar Jacobson i hans "Objectory"-metod och som har anammats i objektorienterad systemutveckling, till exempel i UML. (UML Destilled, 1999, Martin Fowler with Kendall Scott)

Batch – en process körs till den är helt klar, utan att avbryta eller temporärt lämna över exekveringsmöjligheten till en annan process eller tråd.

Designprocess – som arbetsmodell, de olika steg som passeras för att nå målet. Ett välkänt exempel är vattenfallsmetoden.

Domän – område (SAOL). Den organisation och miljö där ett system fyller sin funktion. Det behöver inte vara ett datoriserat system, utan det kan röra ett manuellt system.

Eiffel. – objektorienterat programmeringsspråk liknande C++. Eiffel använder kontraktskrivning, vilket innebär att klasser kan kräva att vissa krav skall uppfyllas innan en annan klass tillåts att använda dess funktionalitet. Eiffel kan även använda vissa efterkrav som skall vara uppfyllda. Detta kan ses som en felkontroll, att funktionen har utfört det som förväntats.

Granskning – (jmf. engelskans review). En expert på människa-datorinteraktion studerar redan gjorda experiment och försöker applicera dessa på det aktuella systemet.

Heuristik – metod syftande till att låta någon vinna kunskap steg för steg genom egen tankeverksamhet. (SAOL)

Händelse – en händelse som inträffar i ett system, systemet exekverar en uppgift eller inträder ett nytt tillstånd.

Interaktion – den dialog som sker mellan människa och dator. Människan interagerar, exempelvis genom tangentbord och mus, och datorn redovisar sitt agerande, exempelvis genom information på skärmen.

Modell – ”Av definition, en teori, taxonomi, eller en modell är en abstraktion av verkligheten varför de måste vara ofullständiga. Dock skall en god teori åtminstone vara förstålig, ge likvärdigt resultat för alla som använder den och hjälpa att lösa specifika praktiska problem”. [Shneiderman]

Piltexkfält – ett textfält för inmatning som har en pil längst till höger. Den ger användaren tillgång till ytterligare förval, utöver det som redan står i fönstret.

Prototyp – kan vara allt från enkla pappersprodukter, till funktionsduglig implementation av kod. Prototyper för ett datorprogram, kan vara gjorda med olika tekniker, ”story boards”, skisser, ”mock-up”, pappersprototyp, körbar prototyp eller ”Wizard of Oz”. Prototyper brukar vanligen delas upp i kategorierna ”throw-away”, evolutionär eller inkrementell.

Reaktiv planering– reaktiv planering innebär här, att aktuell sensorinformation utnyttjas för att förbättra den fortsatta utplaceringen av bojar.

Referenskorg – de tre program och den riktlinje som har använts för att en generell standard skall kunna skapas. Dessa konventioner är de som designförslaget för SubTrack har försökt att följa. Se *Konventioner, 6.1.4.*

Riktlinje – en anvisning om hur något skall utföras, se ut och hur det skall uppföra sig.

Sonarboj – i denna rapport, avses en boj som kan lyssna efter ljud. Bojen är av för enkel teknik, för att kunna avgöra, från vilken riktning ljudet kommer.

Standard – normaltyp eller normalmått. (SAOL)

Uppgift – (jmf. engelskans task) en uppgift som skall utföras, av system eller användare.

Åtgärd – (jmf. engelskans action) en manipulation, som en användare gör på systemet.

11.2 Akronymer

ISO International Standardisation Organisation.

KTH Kungliga Tekniska Högskolan.

PC IBM kompatibel PC, Personal Computer.

UML Unified Modelling Language, Grady Booch, Ivar Jacobson, James Rumbaugh.

WIMP Windows Icons Menus Pointers, alternativt Windows Icons Mouse Pull-Down Menus. Se *Inmatning, användaren till systemet, 4.5.*