



INGENJÖRSHÖGSKOLAN
HÖGSKOLAN I JÖNKÖPING

Användbarhet på Macromedia flash- baserade webbplatser

Fredrik Olsson

EXAMENSARBETE 2006
INTERNETTEKNIK



INGENJÖRSHÖGSKOLAN
HÖGSKOLAN I JÖNKÖPING

Användbarhet på Macromedia flash- baserade webbplatser

Usability at Macromedia flash-based websites

Fredrik Olsson

Detta examensarbete är utfört vid Ingenjörshögskolan i Jönköping inom ämnesområdet Internetteknik. Arbetet är ett led i magisterutbildningen. Författarna svarar själva för framförda åsikter, slutsatser och resultat.

Handledare: Ragnar Nohre

Omfattning: 10 poäng

Datum: 2006-08-10

Arkiveringsnummer:

Postadress:
Box 1026
551 11 Jönköping

Besöksadress:
Gjuterigatan 5

Telefon:
036-10 10 00 (vx)

Abstract

This report portrays an introduction to the sequence of work developers have to go through to produce user-friendly websites in Macromedia Flash. The paper's goal is to understand the importance of user-focused thinking in the development process and help to improve the usability of Macromedia Flash content in general.

Despite many critics say Macromedia Flash content doesn't have good usability this paper shows that it's highly possible to make flash pages usable for the visitor with technical solutions in Macromedia Flash's programming language actionscript.

This report also includes a web based template with all the technical solutions applied to it so developers can follow every step of the build up.

Sammanfattning

Detta examensarbete behandlar den rad av arbete som utvecklare måste gå igenom för att producera användbarhetsanpassade flash-baserade webbplatser. Rapportens mål är att få utvecklare att förstå vikten av att alltid tänka på besökaren i utvecklingsprocessen och på så sätt hjälpa till att förbättra flash-baserat innehåll generellt.

Trots att många kritiker anser att flash-baserade webbplatser inte är bra ur användbarhetssyfte visar denna rapport att det är högst möjligt att skapa användbarhetsanpassade flash-sidor för besökarna med hjälp av Macromedia Flashs eget programmeringsspråk actionscript.

Denna rapport innefattar också en webbaserad mall med alla de tekniska lösningarna som nämns i texten för att utvecklare enkelt skall kunna följa varje steg i utvecklingsprocessen.

Nyckelord

Användbarhet, Usability, Macromedia Flash, Actionscript.

Innehållsförteckning

1	Inledning	4
1.1	BAKGRUND	4
1.2	PROBLEMFÖRMULERING	5
1.3	SYFTE.....	5
1.4	AVGRÄNSNINGAR.....	5
1.5	DISPOSITION.....	5
1.6	TERMINOLOGI	6
2	Teoretisk bakgrund	7
2.1	DEFINITION AV ANVÄNDBARHET.....	7
2.2	MACROMEDIA FLASH.....	7
2.3	ANVÄNDBARHET I MACROMEDIA FLASH	9
2.4	TYPISKA ANVÄNDBARHETS FEL	10
2.4.1	Nerladdningstid.....	10
2.4.2	Rullningslistor	11
2.4.3	Markerbar text och utskriftsmöjligheter	12
2.4.4	Webbläsarens bakåtknapp	12
2.4.5	Ge besökare alternativ	13
2.4.6	Webbplatsens navigation	13
2.4.7	Lättscannad text	14
3	Genomförande.....	15
3.1	LADDNINGSIKRIKERE	15
3.2	KOMPONENTEN UI SCROLLBAR	18
3.3	UTSKRIFT AV TEXTINNEHÅLL.....	18
3.4	ANVÄNDA WEBBLÄSARENS BAKÅT- OCH FRAMÅTKNAPP	19
3.5	ANVÄNDBARHETSANPASSAD NAVIGATION	22
3.6	TESTER AV WEBBPLATS.....	23
3.6.1	Resultat testperson ett:.....	24
3.6.2	Resultat testperson två	24
4	Resultat.....	26
5	Slutsats och diskussion	27
6	Referenser	28
6.1	BÖCKER	28
6.2	WEBBPLATSER	28
6.3	FORUM OCH INSPIRATIONSKÄLLA	29
7	Sökord	30
8	Bilagor	31

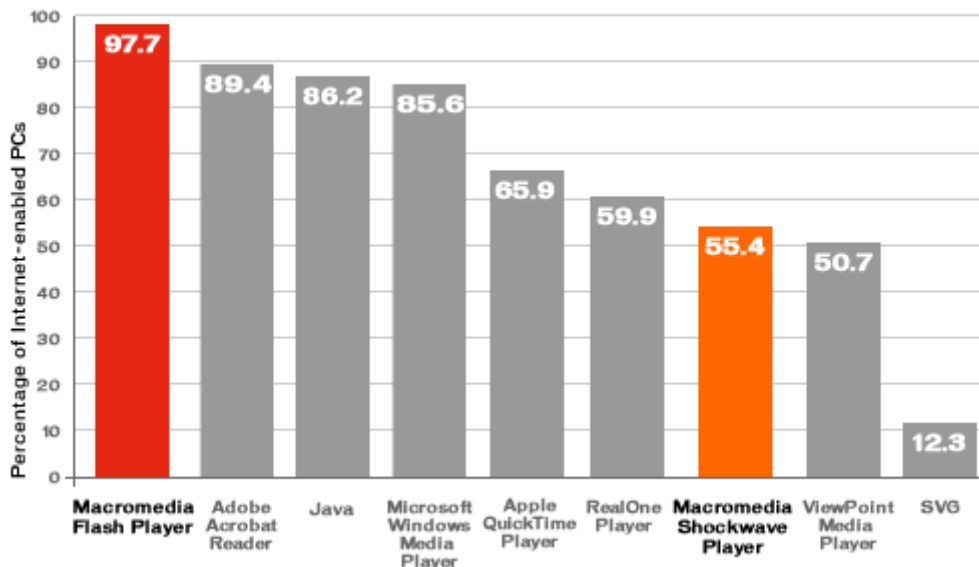
1 Inledning

Jag har valt att dela upp inledningen i bakgrund, problemformulering, syfte, avgränsningar och disposition.

1.1 Bakgrund

Användbarhet är ett begrepp som används mycket inom området MDI (människa-dator-interaktion). Användbarhet är ett viktigt begrepp för att produkter eller system ska kunna spara tid, pengar och ork. Många utvecklare av webbplatser utgår från komplexa tekniker när de utformar gränssnitt. Andra utgår från fräcka funktioner och snygga detaljer. Men egentligen är de ju användarnas behov som ska tillgodoses, inte utvecklarnas egna intressen. Vad har webbplatsen annars för mening om inte användarna kan förstå hur den fungerar och inte finner den information man letar efter. Användbarhet är det viktigaste på en webbplats för att målgruppen ska attraheras och stanna kvar.

Macromedia Flash är idag ett väldigt utbrett format för publicering av interaktiva webbplatser. Enligt undersökningar har 97,7 % av alla Internetanvändare Macromedia flashs insticksprogram installerat.



(http://www.macromedia.com/software/player_census/flashplayer)

Tyvärr så har generellt flash-baserade webbplatser en mycket lägre användbarhet i jämförelse till webbplatser skapade i standardiserad html. Webbplatserna som är flash-baserade koncentrerar sig istället på att skapa ett häftigt gränssnitt och glömmer istället bort användbarheten. Med hjälp av actionscript som är Macromedia Flash's egna kraftfulla programmeringsspråk finns det dock möjligheter att skapa den användbarhet som efterfrågas med lite uppfinningsrikedom och finesse.

1.2 Problemformulering

Den centrala frågeställningen i rapporten och det jag ska undersöka är hur man kan öka användbarheten för besökare på flash-baserade webbplatser genom att programmera och integrera tekniska lösningar. Min frågeställning ser ut enligt följande: ”Hur kan man öka användbarheten på Macromedia flash-baserade webbplatser?”

1.3 Syfte

På Internet finns mängder med webbplatser som är uppbyggda i programmet Macromedia Flash. Generellt har dessa en mycket låg användbarhet i jämförelse till webbplatser konstruerade i vanliga html. Avsikten med rapporten är att definiera de vanligaste problemen ur användbarhetssyfte som dessa webbplatser har, samt i högsta möjliga mån försöka finna tekniska lösningar på dessa. Från den allmänna definitionen av ordet användbarhet enligt användbarhetsgurun Jakob Nielsen har jag arbetat fram de viktigaste kriterierna för användbarhet hos en webbplats.

Resultatet är också skapat som en typ av mall/template riktad till webbdesigners och programmerare på hur användbarhetsfunktioner kan appliceras för att skapa mervärde på flash-baserade webbplatser.

1.4 Avgränsningar

Rapporten kommer inte att omfatta tekniska lösningar på samtliga angivna användbarhetsproblem som generellt finns på flash-baserade webbplatser. Rapporten kommer inte heller omfatta samtliga användbarhetsproblem utan bara att behandla de mest betydelsefulla. Användbarhet ur vissa handikappades synvinkel såsom synskadade kommer inte heller att inkluderas i rapporten.

1.5 Disposition

Rapporten är konstruerad utifrån den mall som tillhandahålls av Ingenjörshögskolan i Jönköping. Den första delen inleds med en kort sammanfattning följt av arbetets bakgrund, dess syfte & mål samt vilka avgränsningar som har gjorts. Denna del går under namnet *inledning*.

Inledningen följs av en *teoretisk bakgrund* där jag behandlar de litterära källor som uppsatsen och det undersökande arbetet grundas på. Utifrån de litterära teorierna om användbarhet formulerar jag en sammanfattande definition kring de mest relevanta inom ämnet användbarhet på flash-baserade webbplatser. Här kommer också definitionen på de vanligaste felen som görs av utvecklare ur användbarhetssyfte.

Under *genomförandet* förklaras hur mallen/templatens har utvecklats ur teknisk synvinkel för att finna lösningar kring användbarhetsproblemen som allmänt finns på flash-baserade webbplatser. Här redovisas också en del väsentlig programmeringskod med tillhörande förklaringar.

Slutligen följer *resultat* och *diskussion* kring examensarbetet där resultaten utvärderas och granskas ur kritisk synvinkel för att säkerhetsställa nyttan av användbarhet på Internet.

1.6 Terminologi

Webbplats = En webbplats är en samling webbsidor som hålls samman av en gemensam rubrik och sköts av en och samma webbmaster/redaktör.

Hemsida = Med ordet hemsida menas den första webbsidan som besökaren möter på en och samma webbplats. Ordet hemsida är synonymt med förstasida.

GUI (Graphical User Interface) = GUI är ett standardiserat användargränssnitt baserat på fönster, ikoner och andra grafiska komponenter som används på dagens PC-plattformar.

Actionscript = Actionscript är Macromedia Flashs eget programmeringsspråk som bygger på skriptspråket JavaScript.

Vektorgrafik = Vektorgrafik består av geometriska primitiver som punkter, linjer, cirklar som tillsammans beskriver en bild. Detta till skillnad från rastergrafik som består av bildpunkter (pixlar). Eftersom alla geometriska primitiver kan förstöras eller förminsas hur mycket som helst kan man förstora vektorgrafik utan att bilden blir suddig eller tar större plats.

Bitmap/rastergrafik = Bitmap/rastergrafik är ett sätt att skapa digitala bilder. I bitmap/rastergrafik är bilder uppbyggda av rader med bildpunkter (pixlar), varje bildpunkt ges en färg från en ändlig palett av tillgängliga färger.

Flashfil = En flashfil med filändelsen (.fla) är datafilen som används vid utveckling av applikationer i programmet Macromedia Flash. En flashfil kan inte visa sitt innehåll på Internet, filen kan heller inte editeras utan utvecklingsprogrammet.

ShockwaveFlash (SWF-fil) = Det filformat som exporteras från en flashfil (.fla) i utvecklingsprogrammet Macromedia Flash till SWF-formatet. Det är detta filformat som används för att få sin applikation att fungera på bland annat Internet, CD-ROM och är ofta implementerad i en HTML-sida.

2 Teoretisk bakgrund

2.1 Definition av användbarhet

Den standard som användbarhet definieras av lyder enligt ISO 9241-11. ”*Den utsträckning till vilken en specificerad användare kan använda en produkt för att uppnå specifika mål, med ändamålsenlighet, effektivitet och tillfredsställelse, i ett givet användningssammanhang.*” [9]

Med användbarhet menas alltså att man skall göra en ordentlig kraftansträngning för att skapa lättanvändhet. Systemet/applikationen skall skapas så den känns lättanvänd från början och kan användas av besökaren omgående utan funderingar och tveksamheter. En förutsättning för att lyckas leverera kvalitet till användarna är att man fokuserar på användarnas behov genom hela projektprocessen.

Användbarhet handlar inte bara om att göra det enkelt för användarna utan kan också vara skillnaden mellan en lyckad och misslyckad Internetsatsning. En hög användbarhet kan också öka förtroendet för företaget och dess produkter vilket kan leda till ökad försäljning/lönsamhet. Besökare som inte förstår en webbplats struktur ger snabbt upp generellt och söker information på annat håll där det är mer lättillgängligt. [10]

2.2 Macromedia Flash

Flash gavs första gången ut 1996 och är ett webbutvecklingsprogram skapat av företaget Macromedia som också utvecklat program som Dreamweaver, Fireworks, Director, Freehand med flera. Numera ägs Macromedia av företaget Adobe som bland annat utvecklat program som Photoshop, Illustrator och InDesign.

Programmet växte snabbt till att bli ledande för skapandet av vektorbaserade webbplatser. Vektorgrafik bygger på att man ritat upp en bild med hjälp av matematiska formler, så kallade vektorer, där varje vektor representerar en linje eller en kant på en polygon. Vektorgrafik är förvisso mer resurskrävande för datorn på grund av alla olika matematiska beräkningar som görs i jämförelse till klassisk bitmapsgrafik, men i gengäld krävs det avsevärt mindre datautrymme och mindre nerladdningstider. Från början gick det inte att vare sig skapa interaktion mellan dator och användare eller ladda in extern data, men i takt med att programmet har förbättrats och genom utvecklandet av Flash's programmeringsspråk actionscript har detta möjliggjorts. En stor fördel med Flash är actionscript's kraftfullhet vilket ger en möjlighet att kommunicera med HTML och utomstående scriptspråk såsom PHP och ASP. Filnamnsuffixet som används är SWF och det är ett öppet format. Flash behöver nödvändigtvis inte bara användas till webbsidor, utan även fristående program kan utvecklas såsom reklam på DVD skivor eller menyer till spel och filmer etc.

Idag är Flash mycket mer än bara ett grafikprogram. Nuvarande Flashversion heter Flash 8 och används bland annat för att skapa så kallade Rich Internet Applications (RIA). RIA är en blandning av webb - och mjukvarubaserade applikationer för Internet vilka erbjuder större funktionalitet än vad DHTML kan erbjuda i form av ljud, video, typsnitt och animationer på Internet.

Objekt, grafik och animationer styrs i Flash genom programmets inbyggda programmeringsspråk Actionscript som är ett äkta objektorienterat programmeringsspråk baserat på JavaScript. I flash finns det tre element, även kallade symboler som bör nämnas och de är grafik, knappar och filmklipp. Dessa symboler kan tilldelas funktioner, det vill säga programmeringskod (Actionscript). Macromedia Flash använder sig också av en tidslinje (timeline) där element kan styras mer grafiskt.

Namnet Macromedia Flash refererar både till utvecklingsverktyget och till insticksprogrammet Macromedia FlashPlayer. [12]

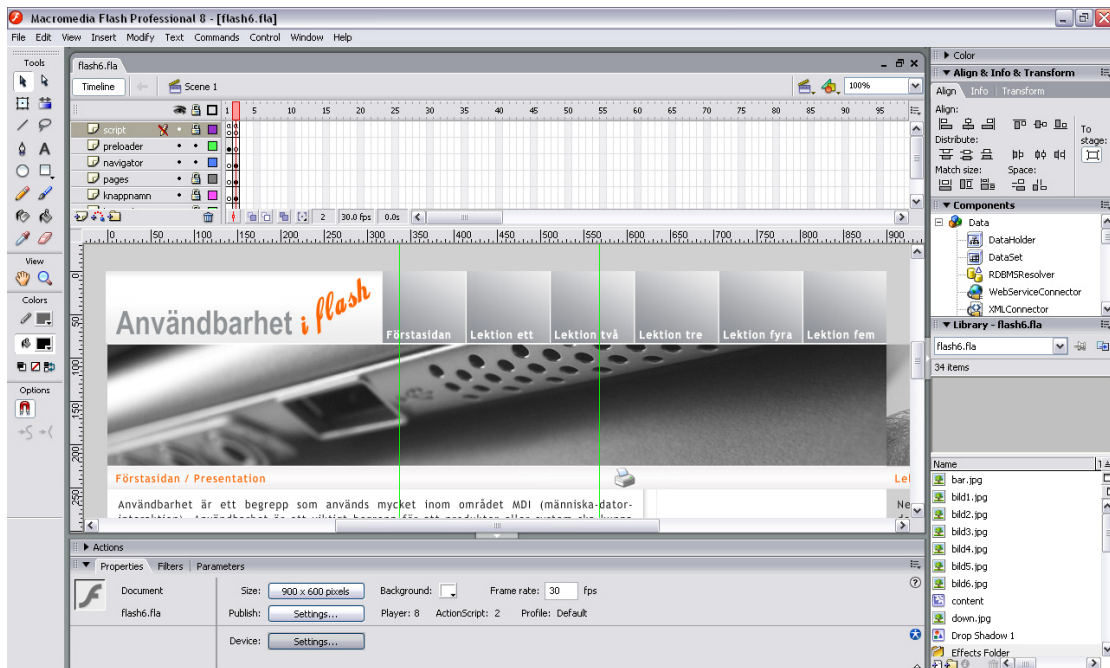
Fördelarna med vektorgrafik och Flash's SWF:

- Designmöjligheterna eftersom man enkelt kan animera, skala och förändra utan att kvalitén blir lidande eller information får förlorad.
- Kraftfullheten att på ett enkelt sätt genomföra olika transformationer på bilden eller delar av den, till exempel rotationer eller förvrängningar.
- Att tekniken tack vare sin matematiska natur fungerar utmärkt även i 3 dimensioner.
- Precis som CSS med HTML, Postscript, SCG och PDF kan Flash användas för att specificera en exakt positionering av objekt och text i en SWF-fil. Detta ger designern en stor frihet att bestämma bland annat hur ett användargränssnitt ska se ut.
- SWF stöder progressiv nedladdning och strömning som betyder att delar av filen kan ses utan att hela filen blivit nedladdad. Detta används i mallen/templatens till nerladdningsindikerare.
- SWF använder Unicode som är en textstandard som är språk- och teckenoberoende.

Nackdelarna med vektorgrafik och Flash's SWF:

- Att tekniken inte är speciellt bra till bilder med stor variation och detaljrikedom, som exempelvis fotografier.
- Att det är väldigt resurskrävande för datorn och att bilderna måste konverteras till rastergrafik innan den kan ritas upp på skärmen.
- En SWF-fil är designad att köras som en säker datorapplikation, med vilket menas att den inte kan få tillgång till vitala delar i användarens datorsystem, den har heller inte möjlighet att skriva till eller spara lokala filer på datorn.

- Innehållet i en SWF-fil är av binär typ och det är därför svårare för en sökmotor att tolka och indexera innehållet på en flash-baserade webbplats.



Utvecklingsmiljön i Macromedia Flash 8.

2.3 Användbarhet i Macromedia Flash

De tre vanligaste huvudargumenten som användbarhetskonsulter såsom Jacob Nielsen och Steve Krug anger för anledning till att flash-baserade webbplatser saknar en bra användbarhet är:

- Flash-baserade webbplatser innehåll är oftast störande och meningslösa.
- Flash-baserade webbplatser byggs ofta en gång och sen uppdateras de aldrig.
- Informationsinnehållet på flash-baserade webbplatser ofta är kortfattat och följer inte standarden för publicering av text och bild i form av lätläshet och överskådlighet.

Dessa argument stämmer väl överrens med majoriteten av webbplatserna på Internet. Anledningen till kritiken är oftast att webbdesignern koncentrerat sig helt på design och interaktivitet vid konstruktionen av webbplatsen och nonchalerad tanken på användbarhet. Kritikerna dömer Macromedia Flash efter dessa webbplatser istället för att se vad man egentligen kan göra med programmet.

För att lyckas med ett webbprojekt skall användbarhetsreglerna finnas i tanke redan vid planeringen och uppbyggandet. Att försöka applicera användbarhet efteråt skapar ofta en massa extra jobb och slutar sällan lyckat. [1] [6]

2.4 Typiska användbarhetsfel

Nedan följer en definition kring de mest relevanta fel som utvecklare gör vid skapandet av flash-baserade webbplatser ut användbarhetssyfte. Genom att ha en god förståelse för dessa användbarhetsproblem kommer utvecklare bli mer framgångsrika vid skapandet av användbarhetsanpassade flash-baserade webbplatser.

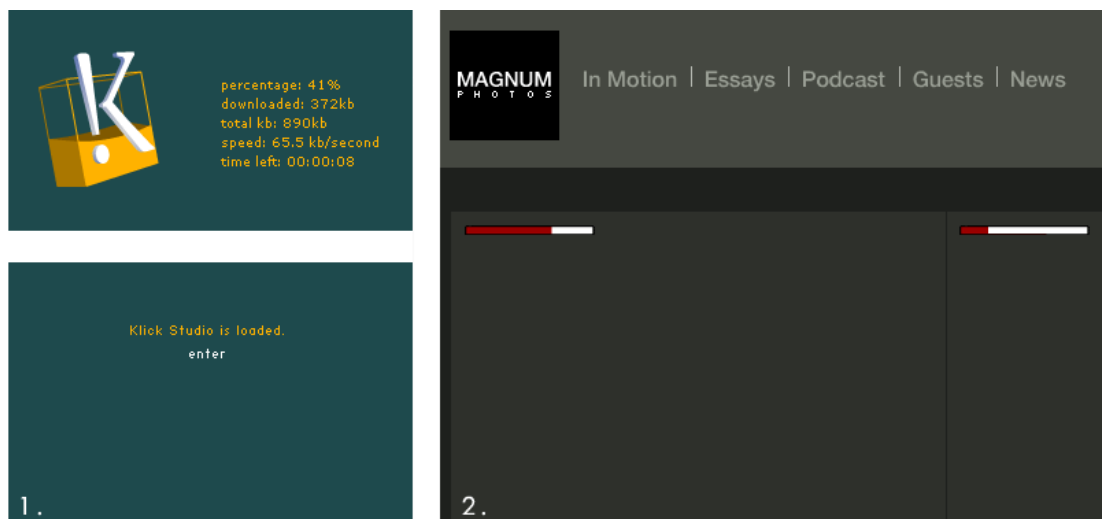
2.4.1 Nerladdningstid

Nerladdningstiden för hela hemsidan är alltid viktig ur användbarhetssyfte, oberoende om den är uppbyggd i Macromedia Flash eller vanlig statisk html. Webbplatsens besökare har inget tålamod, de vill se hemsidans helhet direkt. På vanliga statiska html-sidor laddas alltid text och struktur först för att sedan följas av mer bandbreddskrävande material som exempelvis bilder och filmer. På detta vis tillfredställs besökaren på html-sidor automatiskt eftersom det tydligt ges feedback på att bilderna är under nerladdning och visas efter hand de är klara.

I Macromedia Flash laddas innehållet på webbplatsen ner i en källfil med filnamnsuffixet SWF. Denna fil innehåller hela webbplatsen såsom text, designelement och fotografier kompilerat till vektorgrafik och kan därför bli tungladdad beroende på webbplatsens storlek.

Lösningen på detta problem är att skapa en laddningsindikerare (*eng; preloader*) för webbplatsen och på så sätt ge besökare feedback på att sidan håller på att laddas och hur mycket som återstår av nerladdningen. I exempel ett (1) nedan återfinns en laddningsindikerare som visar detaljerad information för besökaren under laddningen av webbplatsen.

Efter att filen är färdigladdad visas en entréknapp så besökare utan webbläsarfönstret aktivt inte ska missa introduktionen. Detta är speciellt viktigt eftersom besökare med låg bandbredd ofta tittar på andra webbplatser under laddningstiden.



Laddningsindikatorer från klickstudio.com och magnumphotos.com

En ännu bättre lösning vid större webbplatsprojekt där swf-filerna kan vara stora är att ladda in alla bilder/filmer dynamiskt från en extern källa efter att själva webbplatsen presenterats. På så sätt visas själva webbplatsen snabbare samt ger tydlig feedback på vad som håller på att laddas in för besökaren (bild två). [8] [11]

2.4.2 Rullningslistor

Trots att många utvecklare känner till riskerna av att skapa dåligt designade rullningslistor utbreder sig detta problem fortfarande i stor skala (bild ett). Resultatet blir ofta frustrerade användare som missar webbplatsens innehåll. Det viktigaste att tänka på när man ska skapa en bra rullningslist ur användbarhetssynvinkel är att den verkligen ser ut och uppfattas som en rullningslist enligt GUI standarden som är välkänd för Internetanvändare (bild två).

På flash-baserade webbplatser är det inte bara designen på rullningslisterna som ofta ställer till problem. Även själva funktionen är ofta bristfällig då inte rullningshjulet på musen fungerar som besökare är vana vid på html-sidor. Detta kan leda till att besökare missar viktig information då de antar att det inte finns mer text att läsa i en textruta eftersom inte rullningen fungerar från musens rullningshjul.

Det finns flertalet olika lösningar på problemet. Den vanligaste är att separera innehållet på de flash-baserade webbplatser i logiska delar som man istället kopplar till individuella html-sidor. På detta sett sker navigeringen istället externt till olika html-sidor när besökaren klickar på länkar på webbplatsen. Fördelen är att besökaren kan skapa bokmärken till specifik information då varje del sida ligger som individuella html-sidor. Nackdelen är fördröjningen när besökaren navigerar samt att animeringar och bildövergångar inte kan användas vid navigering mellan delsidorna.

En annan lösning som dock anses irriterande och dålig är att låta webbplatsen presenteras i ett popup-fönster utan verktygsfält så besökaren inte har tillgång till bakåtknappen.

Den mest användbara lösningen är att använda JavaScript i en dold ram som hela tiden kommunicerar med den flash-baserade webbplatsen genom actionscripts och ger feedback på var besökaren senast befann sig. Denna lösning gör att webbläsarens navigeringsknappar fungerar internt på den flash-baserade webbplatsen utan att ta bort möjligheten för bildövergångar eller andra animeringar mellan delsidorna. [6] [8] [14]

2.4.5 Ge besökare alternativ

För de få procent av besökarna som inte har insticksprogrammet Macromedia Flash installerat är det viktigt att erbjuda webbplatsens innehåll på ett vanligt html-alternativ. Detta alternativ passar även besökare med låg bandbredd exempelvis modem. Detta alternativ är också bra för registrering av webbplatsen till sökmotorer då deras robotar inte kan läsa innehållet i swf-filer. [12]

2.4.6 Webbplatsens navigation

Varje webbplats viktigaste element ur användbarhetssynvinkel är navigationen. Vare sig om det handlar om en vanlig html webbplats eller en flash-baserad webbplats. Innehållet bör vara välorganiserat då besökaren inte vill spendera lång tid på att försöka hitta önskvärd information på webbplatsen. En god struktur är också viktigt eftersom man inte kan visa alla möjliga destinationer på alla sidor.

Navigeringsdesignen är viktig för att besökaren hela tiden ska ha en möjlighet att veta var den befinner sig och var den är på väg. Länkarna bör även vara namngivna logiskt så man alltid vet vad som finns på varje del sida.

Länkarna bör dessutom följa Fitt's lag som lyder enligt följande: "The time to acquire a target is a function of the distance to and size of the target". Enkelt översatt menas detta att ju större länkarna är ju enklare är de att klicka på och det lättaste objektet att klicka på bildskärmen är det som är närmast musmarkören och störst på bildskärmen. [3] [13]

2.4.7 Lättscannad text

Information i form av text är svårare att läsa på en datorskärm än på papper. Av den anledningen skall text inte animeras då detta kan försvåra ytterligare. Flertalet flash-baserade webbplatser gör detta fel då de animerar texten så användaren inte hinner med i tempot som informationen visas. Många webbplatser väljer också att placera text över bilder vilket också leder till svår läsbarhet på grund av den dåliga kontrasten till bakgrunden.

Det är inte bara kontrasten som gör att flertalet webbplatser är svårlästa. Även teckenstorleken kan vara ett problem många gånger då mindre textstorlek ofta används för att det ser snyggare ut designmässigt. Praxis för skrift på papper är att bokstäver ska vara sex punkters höjd i jämförelse till 12 punkters höjd på en bildskärm. För att text skall bli så lättläst så möjligt på en datorskärm bör man även använda sanserif teckensnitt. Sanserif är teckensnitt som inte har klackar på bokstäverna, exempelvis Verdana, Trebuchet, Geneva och Georgia.

Textraderna bör varken vara för långa eller korta, ungefär 600 pixlars textbredd anses vara bra. För långa rader gör att man lätt hamnar fel vid radbrytningar och för korta rader ger en hackig lästrytm vilket ofta ger upphov till mindre fina hålrum i texten. Radmellanrummen ska också vara större än ordmellanrummen.
[3]

3 Genomförande

Genomförandet av detta projekt har efter den teoretiska bakgrunden bestått av att utveckla en mall/template som stödjer ovanstående väsentliga användbarhetsproblem som flash-baserade webbplatser har idag. Genom att utveckla tekniska lösningar vill jag bevisa att användbarhetsproblemen som kritikerna av Macromedia Flash anmärker på egentligen inte beror på produktens kapacitet utan på personerna som utvecklar och designar flash-baserade webbplatser.

Det praktiska resultatet kommer att ligga som en mall/template så utvecklare och designers kan ladda ner hela källfilen innehållande kommentarer på de viktigaste funktionerna. Anledningen till detta presentationssätt är att alla programmeringskod inte kan redovisas i rapporten. För utvecklare som använder Macromedia Flash är det även enklare att få ett helhetsintryck när man ser programmeringskoden på plats i programmet länkat till varje objekt.

För att sedan utvärdera den färdiga mallen/template kommer jag göra ett användartest utifrån Bruce Tognazzini's metod i boken "TOG on interface" där personer från olika ålders- och samhällsgrupper med olika stor datorvana får prova resultatet för att mäta om användbarhetsgraden verkligen har ökat ur besökarens synvinkel. Testpersonerna kommer även att få utföra samma operationer på en flash-baserad webbplats som inte är användbarhetsanpassad för att kunna jämföra de olika resultaten. Feedbacken från användartesten kommer att ligga som grund för den avslutande diskussionen där jag analyserar om det verkligen finns behov av användbarhet på flash-baserade webbplatser.

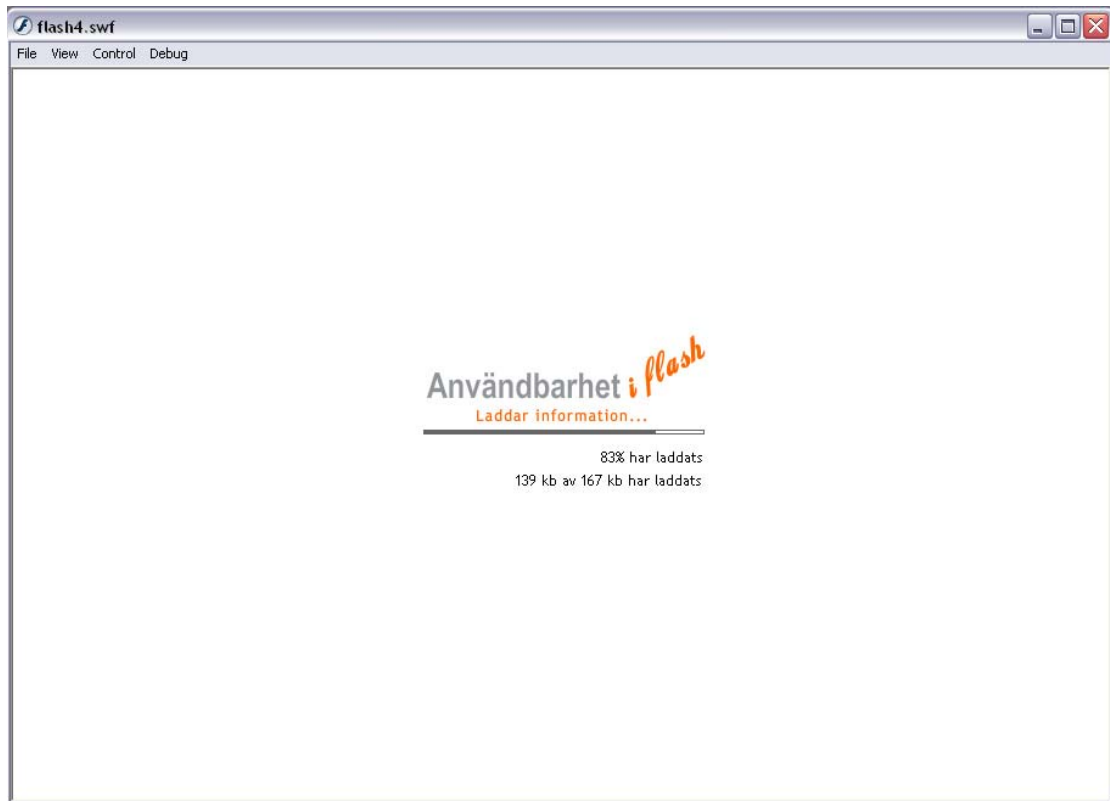
3.1 Laddningsindikerare

Som nämnts ovan så är nerladdningstiden för hela hemsidan mycket viktigt för besökaren. Internetanvändare idag har inget tålamod och vill ha information om vad webbplatsen handlar om inom fyra sekunder annars ger de upp och söker sig vidare på annat håll på Internet. En laddningsindikerare som i exemplet nedan ger användaren den feedback som behövs för att få honom/henne att vänta.

- 1. Skapa utrymme** - Det första man måste göra är att bereda plats för sin laddningsindikerare genom att flytta allt innehåll på tidslinjen ett steg så att nyckelruta ett blir ledig. Detta görs för att ge feedback till besökaren att något håller på att laddas direkt när de kommer in på webbplatsen.

2. **Stoppa swf-filmen** – Meningen är att swf-filmen ska stanna på nyckelruta ett för att börja ladda och sedan gå vidare när laddningen är klar. Därför måste man skapa ett stopp på nyckelruta ett genom att markera den och ange `stop()`; i actionscriptfönstret. Bygger ens swf-film på att varje del sida har en nyckelruta på tidslinjen i roten så får man inte glömma att uppdatera länkarna då dessa har blivit förskjutna en nyckelruta.
3. **Skapa grafiken** – Man kan välja själv hur ens laddningsindikerare ska se ut. Det enda man ska tänka på är att den ska vara lättladdad. Först bör man dock skriva text som säger att filen laddar och eventuellt skriva några ord som återspeglar vad besökaren kan vänta sig på webbplatsen. Alternativt kan man komplettera med en logotyp.
4. **Ge feedback** – Feedback skall ges till användaren för den skall veta hur fort nerladdningen går och hur mycket som är kvar att ladda. I exemplet används ett enkelt streck som laddningsindikerare och sköts med programmering i actionscript, men man måste rita den för hand först. Man får inte glömma att göra laddningsindikeraren som ett movieclip. Vidare bör man även ge feedback i form av text (procent etc.) genom att skapa informationsrutor med variabelnamn som också ska kopplas till programmeringen.
5. **Programmering** – Först markerar man sin laddningsindikerare och sen anger man följande kod i actionscript fönstret.

```
onClipEvent(enterFrame) {  
  
total=_root.getBytesTotal(); //hämtar information om filen  
totalkb=Math.round(total/1000);  
laddat=_root.getBytesLoaded();  
laddatkb=Math.round(laddat/1000);  
procent=Math.round(laddatkb/(totalkb/100)); //räknar procent  
_root.text1=procent + "% har laddats" //skriver ut procent  
this._xscale=procent; //förflyttar stecket på laddningen  
//skriver ut mängden kb som är laddat av total mängd  
_root.text2=laddatkb + " kb av " +totalkb + " kb har laddats";  
//Kollar om laddningen är klar, i så fall går den vidare  
if(laddat==total){  
_root.gotoAndStop(2);}}
```



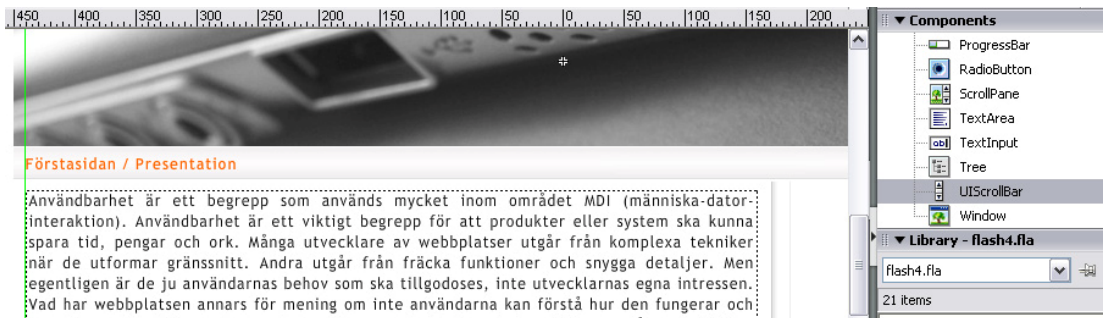
Färdig laddningsindikerare som ger feedback till besökaren under laddningen.

En alternativ lösning till denna laddningsindikerare är att sköta inladdningen av själva huvudfilen från en liten extern swf-fil istället för att göra det i samma fil enligt ovan. Detta kan vara en fördel på större webbplatser där mycket tung grafik skall laddas in då laddningsindikerare som ligger i samma fil oftast har en viss fördröjning. En lösning på detta problem liknar ovanstående lösning men man kallar istället på den externa filen enligt följande:

```
myMovie = "flash.swf"; //vilken fil som skall laddas
mclip = this.createEmptyMovieClip("themovie", setDepth());
// skapar nytt movieclip för den externa filen
mclip.loadMovie(myMovie, setDepth()); //applicerar filen
preloadFirst(mclip);
//kallar på samma funktion för inladdning som ovanstående
lösning
```

3.2 Komponenten UI Scrollbar

Det ligger ingen hemlighet bakom skapandet av en bra standardiserad rullningslist i Macromedia Flash även då man oftast stöter på egenkonstruerade med dålig användbarhet. Under komponentbiblioteket ligger klassen *UI ScrollBar* som kan appliceras på dynamiska textfält utan programmering. Även designen på rullningslistan är väldigt enkel, proper och neutral.



Klassen UI ScrollBar i komponentbiblioteket kan appliceras till textfält.

3.3 Utskrift av textinnehåll

För att besökare skall kunna markera text på flash-baserade webbplatser krävs att textfälten är inställda på markerbarhet. Möjligheten finns då att kopiera text för att sedan klistra in i exempelvis Microsoft Word om man vill göra utskrift utan att få med den grafiska layouten. Dock finns det ingen möjlighet att få utskrifter direkt från den flash-baserade webbplatsen utan actionscript programmering. För att skapa skrivarutskrifter av innehållet på flash-baserade webbplatser används klassen *printjob*. Klassen *printjob* kan användas för både utskrift av hela movieclip och för enstaka textfält som visas i exemplet nedan.

Då Macromedia Flash skriver ut innehållet i textfält som en bitmap krävs att man först skalar om själva textfälten till att få plats på ett a4 papper för annars beskär skrivaren texten automatiskt och information förloras. En lösningen är att skala om textfälten i utskriftsögonblicket och sedan återställa storleken efter utskriften är godkänd. Denna procedur sker så snabbt att det är omärkbart för besökaren.

Vidare i utskriftsbearbetningen kan man även definiera ytan på pappret där man vill att informationen skall hamna så marginalerna blir enligt eget önskemål. Detta definieras med hjälp av att ange koordinaterna (*xMin*, *xMax*, *yMin*, *yMax*) för pappersformatet, exempelvis a4.

```
on (release) {
    var skrivut = new PrintJob();
    //skapar en instans av printjob klassen
    var lyckad = skrivut.start();
}
```

```
//kollar om dialogrutan för printning har startat
var pageAdded = false;
if (lyckad) {
    //lägger till vilken textbox som skall printas, men skalar
    först textboxen för att passa på ett a4-papper

    tempwidth = _root.content.pages.page1.textruta1._width;
    _root.content.pages.page1.textruta1._width = 550;

    tempheight =
    _root.content.pages.page1.textruta1._height;
    _root.content.pages.page1.textruta1._height = 800;

    pageAdded =
    skrivut.addPage(_root.content.pages.page1.textruta1);
    }

    if (pageAdded) {
        skrivut.send();
        //skickar vidare till skrivaren
    }

    delete skrivut;

    _root.content.pages.page1.textruta1._width =
    tempwidth;

    _root.content.pages.page1.textruta1._height =
    tempheight;

    //rensar cacheminnet och återställer storleken på textfältet
}
}
```

Programmeringskod för att skapa en utskrift av textruta1 när man klickar på den tillagda utskriftsikonen.

3.4 Använda webbläsarens bakåt- och framåtknapp

Att Macromedia Flash inte stödjer användandet av webbläsarens bakåtknapp är den enskilt största anledning till kritiken av formatet ur användbarhetssynvinkel. Lösning på problemet görs genom att applicera de flash-baserade delsidorna i webbläsarens historiefunktion så att bak- och framåtknappen inte gör att besökaren lämnar webbplatsen.

För att lösningen skall fungera måste ett dold html ram inkluderas på webbplatsen där webbläsarens historiefunktion genereras. För varje flash-baserad del sida som besöks laddas en html-sida i den dolda ramen och uppdaterar webbläsarens historia. Html-sidan som laddas för varje del sida på den flash-baserade webbplatsen returnerar en variabel genom JavaScript till swf-filmen. På detta sätt får swf-filmen reda på var den skall navigera då variabeln innehåller information om vad besökaren har klickat på. En förenklad förklaring i tre steg kan se ut enligt följande:

1. Besökaren klickar på en knapp i menyn på den flash-baserade webbplatsen.
2. En html-sida laddas i den dolda ramen där ett JavaScript returnerar en variabel som innehåller information om vad besökaren klickat på. Detta leder även till att webbläsarens historiefunktion sparar den nya html-sidan som laddats. Varje del sida på den flash-baserade webbplatsen har en dold html-sida refererad till sig som laddas vid klickning på länkar. Dessa är naturligt döpta till 1.htm, 2.htm och så vidare.
3. Swf-filmen tar emot den externa variabeln från html-sidan som innehåller information om vad besökaren har klickat på för att sedan navigera dit.

Implementeringen av denna funktion kan vid första anblick se väldigt komplex ut då webbplatsen skall kompletteras med programmering på både html- och swf-filerna. Html-sidorna skall kompletteras med JavaScript som sen skall kommunicera med swf-filmen med hjälp av actionscript programmering. För bättre förståelse finns källfilen att ladda ner på webbplatsen. Källfilen kan då användas som mall/template då alla väsentliga skript innehåller förklarande kommentarer.

1. Implementering av JavaScript:

```
<script language="JavaScript"> //denna fil heter 1.htm
    parent.flashframe.setPage(1);
</script>
```

De dolda delsidorna som har laddats kommunicerar med html-sidan där swf-filmen finns genom JavaScript där den skickar en variabel och anger att besökaren har tryckt på en länk och anropar sida *ett* (i detta fallet) i swf-filmen.

```
<script language="JavaScript">
function setPage(newPage) {
    if (window.document.mymovie) {
        window.document.mymovie.SetVariable("page", newPage);
    }
}
```

```
}  
}  
</script>
```

SetPage funktionen anropas från en av de dolda delsidorna och sätter *page* variabeln i swf-filmens tidslinje till värdet som från den dolda delsidan. Det leder till att swf-filmen får reda vilken länk besökaren har klickat på.

2. Implementering i flash:

```
_root.page = 1;  
Stop();  
  
onClipEvent (enterFrame) {  
    this.checkPage();  
}
```

Första gången en besökare kommer in på webbplatsen sätts variabeln *page* till ett och stannar på första nyckelrutan. Vidare körs funktionen *onClipEvent* hela tiden under besökarens vistelse på webbplatsen för att kontrollera om man har klickat på någon länk.

```
function checkPage() {  
    newPage = _root.page;  
    if (oldPage != newPage) {  
        _root.content.pages.gotoPage(newPage);  
    }  
    oldPage = newPage;  
}
```

Funktionen *checkPage* hämtar *page* variabeln från tidslinjen och kontrollerar om den har ändrat sig. I så fall navigerar den vidare till den nya delsidan. Efter navigering har startat skrivs *newPage* variabeln över av *oldPage* så denna funktion fungerar igen nästa gång besökaren klickar på en länk. Funktionen *gotoPage* har egentligen inget med bakåtknappens användning att göra, utan i detta fallet gör den bara att delsidorna flyttar sig i sidled på webbplatsen.

```

on (release) {
    getURL ("2.htm", "bottom");
    _root.page = 2;
}

```

Händelse applicerad på navigationslänkarna/knapparna på webbplatsen som öppnar den dolda html-sidan (i detta fall 2.htm) för att historiefunktionen skall uppdateras. Detta gör i sitt fall att webbplatsen navigeras till delsidan två.

Nu är grundkoden till bakåtknappens funktion klar. Det man avslutningsvis måste göra är att modifiera den efter sitt specifika gränssnitt och addera programmeringskoden till rätt objekt, samt förändra sökvägarna.

3.5 Användbarhetsanpassad navigation

Att skapa en användbarhetsanpassad navigation innehåller inga svårigheter tekniskt utan handlar huvudsakligen om att följa designregler och ge feedback till besökaren på var man befinner sig just nu och var man navigera vidare.

Mallen/templatens knappar är skapade enligt Fitt's lag för att besökaren enkelt skall kunna klicka på dem utan precision. Knapparna ger även feedback genom att lysa orange då man befinner sig på den delsidan. Utvecklar man en flash-baserad webbplats där varje delsidan ligger på en ny nyckelruta i tidslinjen skapas feedback lättast genom att ange att knappen skall lysa exempelvis orange på just den nyckelrutan.

Användbarhet *i flash*

Förstasidan Lektion ett **Lektion två** Lektion tre Lektion fyra Lektion fem

Lektion två / Rullningslistor

användare som missar webbplatsens innehåll. Det viktigaste att tänka på när man ska skapa en bra rullningslist ur användbarhetssynvinkel är att den verkligen ser ut och uppfattas som en rullningslist enligt GUI standarden som är välkänd för Internetanvändare.

På flash-baserade webbplatser är det inte bara designen på rullningslisterna som ofta ställer till problem. Även själva funktionen är ofta bristfällig då inte rullningshjulet på musen fungerar som besökare är vana vid på html-sidor. Detta kan leda till att besökare missar viktig information då de antar att det inte finns mer text att läsa i en textruta eftersom inte rullningen fungerar från musens rullningshjul.

Att lösa problemet med dåligt designade rullningslistor på flash-baserade webbplatser är väldigt enkelt eftersom programmet har en inbyggd komponent som lätt appliceras till textrutor utan att behöva programmeras. Dock verkar många utvecklare inte känna till denna komponent då de ofta använder egenprogrammerade rullningslistor med sämre funktion och design. Macromedia Flashs standardiserade rullningslistor stöder även användandet av rullningshjulet på musen som få flash-baserade webbplatser med

Ladda ner källfilen

Navigationsmenyn ger tydlig feedback till besökaren att man befinner under kategorin Lektion två.

3.6 Tester av webbplats

För att evaluera resultatet av den användbarhetsanpassade mallen/templatens har ett användartest gjorts. Feedback och tips har samlats in med hjälp av scenario och implementerats vid möjlighet för att förbättra webbplatsen.

Testningarna har utförts enligt Bruce Tognazzini's metod i boken "TOG on interface" på följande sätt:

1. Jag presenteras mig.
2. Jag beskriver vilket syfte jag har med användartestet. Det är viktigt att testpersonen förstår att användartestet är ett hjälpmedel för utvecklingen av en webbplats. Jag förklarar tydligt att det är webbplatsen som testas och inte testpersonen.
3. Jag förklarar för testpersonen att den får sluta när som helst.
4. Jag förklarar syftet med den utrustning jag använder.
5. Jag förklarar att testpersonen ska tänka "högt" under testning.
6. Jag förklarar att jag inte kommer att erbjuda någon hjälp när testet väl har börjat.
7. Jag presenterar produkten och beskriver testpersonens uppgift.
8. Jag frågar om testpersonen har några frågor innan testet börjar.
9. Jag genomför testet med testpersonen.
10. Jag analyserar resultatet.

De två användartest som gjordes skedde hemma hos författaren under kvällstid med två olika utomstående testpersoner med grundläggande datorvana. Webbplatserna som testades var denna rapporters mall/template samt reklambyrå Maraccas webbplats (www.maraccas.se). Det korta scenariot som testpersonerna skulle gå igenom såg ut enligt följande:

1. Testpersonen skulle titta på strukturen av webbplatserna och ange antalet delsidor som fanns på var och en. Testpersonen skulle också ange på vilken delsida man befann sig på vid tillfället.
2. Testpersonen skulle sedan navigera till tredje delsidan på webbplatsen och skriva ut textinnehållet utan att få med grafiska element.

3. Slutligen skulle testpersonen navigera till femte delsidan för att sedan ”backa” tillbaka till förstasidan som testpersonen först möttes av när testet startades.

3.6.1 Resultat testperson ett:

Testpersonen hade inga problem att vare sig ange antalet delsidor eller berätta var man befann sig just nu på denna rapportens webbplats.

På den andra webbplatsen angav testpersonen fel antal delsidor som fanns då förstasidan inte fanns med som alternativ i navigationsmenyn utan endast genom att klicka på företagets logotyp. Detta ledde också till att testpersonen angav fel delsida som han befann sig på då han trodde att han var på första länken i navigationsmenyn.

Vidare navigerade testpersonen till tredje delsida på rapportens webbplats utan problem men misslyckades med utskriften han istället använde webbläsarens utskriftsknapp, vilket ledde till att han även fick med de grafiska elementen i utskriften.

På den andra webbplatsen begicks samma misstag då testpersonen navigerade till tredje delsidan och även där skrev ut textinnehållet med de grafiska elementen.

Vidare lyckades testpersonen med sista operationen på rapportens webbplats då han först klickade sig fram till femte delsidan för att sedan med hjälp av webbläsarens bakåtknapp navigera sig tillbaka genom delsidorna tills menyens ”förstasidan” knapp började lysa orange.

Den avslutande operationen på den andra webbplatsen fick ett abrupt slut då testpersonen använde webbläsarens bakåtknapp och hamnade på googles hemsida.

3.6.2 Resultat testperson två

Testperson två hade inte heller några problem att ange antalet synliga delsidor samt berätta var man befann sig just nu på rapportens webbplats.

På Maraccas webbplats uppstod det samma fel som för testperson ett då antalet delsidor inte angavs rätt. Testpersonen var inte heller säker på vilken delsida han befanns sig för tillfället och valde att hoppa över den frågan.

Vidare lyckades testpersonen med både navigeringen och utskriften på rapportens webbplats då han använde sig av utskriftsikonen som är placerad intill textfältet.

Även utskriften misslyckades för testperson två på Maraccas webbplatsen då han först markerade texten med musen för att sedan använda webbläsarens utskriftsknapp. Testperson två märkte då att man inte kunde klicka i utskrift av endast markerat innehåll och valde då att avbryta operationen.

Resultatet för testperson två vid sista operationen blev samma som för testperson ett då även han misslyckades att navigera tillbaka på Maraccas webbplats men lyckades på denna rapports webbplats.

4 Resultat

Undersökningarna i denna rapport, både teoretiska och de praktiska har resulterat i ett hjälpmedel som väl uppnått de målsättningar jag satt upp från starten. Tanken att kunna använda rapporten med tillhörande mall/template som ett underlag för utvecklare och webbdesignerns vid skapandet av användbarhetsanpassade flash-baserade webbplatser tycker jag har infriats.

Alla väsentliga användbarhetsfel som generellt finns på flash-baserade webbplatser har lösts och implementerats i mallen/template med löpande kommentarer så att målgruppen enkelt skall kunna följa och förstå uppbyggnaden. Även då användartestet är gjort i liten skala visar det tydligt att användbarhet på flash-baserade webbplatser både förbättrar och förenklar för besökaren.

Att kritiker som exempelvis användbarhetsgurun Jacob Nielsen inte gillar flash-baserade webbplatser är förståeligt då dessa generellt koncentrerar sig på häftiga animationer och glömmer bort att det är användarnas behov som ska tillgodoses, inte utvecklarnas egna intressen. Dock skall egentligen inte flash-formatet kritiseras utan utvecklarna då det är högst möjligt att skapa bra användbarhetsanpassade webbplatser utan större ansträngning om bara tankarna finns där vid planeringen och uppbyggnaden.

5 Slutsats och diskussion

Jag anser att det i högsta grad finns möjlighet att skapa användbarhetsanpassade webbplatser i Macromedia Flash då formatet är både kraftfullt och lättarbetat. Att 97,7 % av alla Internetanvändare har detta välkända insticksprogram installerat ser jag också som en fördel då bra konstruerade flash-baserade webbplatser verkligen kan skapa mervärde för besökaren i jämförelse till vanliga statiska html-alternativ. En annan stor fördel är att formatet är plattformsoberoende samt att resultatet blir likadant i alla webbläsare.

Själv anser jag dock att användandet av Macromedia Flash på många standardiserade informativa webbplatser kan kännas omotiverat. Jag tycker att formatet egentligen lämpar sig bäst när man verkligen vill skapa mervärde för besökaren i form av exempelvis reklamkampanjer där tyngdpunkten ligger animeringar och avancerad grafik.

Sammanfattningsvis tycker jag att Macromedia Flash är ett kompetent och kraftfullt format för Internet ur många synvinklar, inte bara i tanken på användbarhet. Emellertid skall beslutet om att använda formatet vara väl genomtänkt så att det verkligen är motiverat och skapar mervärde för besökaren.

Vidareutveckling kommer att ske kring bakåtknappens funktion i webbläsaren då den skall styras av endast en dold html-sida som sköter uppdateringen av historiefunktionen med hjälp av att sidnumret istället skickas som parameter i länken. Detta har dock inte färdigställts i mallen/templatet på grund av tidsbrist.

6 Referenser

6.1 Böcker

- [1] Nielsen, Jacob. 2001
Användbar Webbdesign.
Liber AB, Sverige, ISBN 91-47-03612-5

- [2] Krug, Steve. 2000
Don't make me think!
New Riders, USA, ISBN 0-7897-2310-7

- [3] Tognazzini, Bruce. 1992
Tog on Interface.
Addison-Wesley Professional, USA, ISBN 0-2016-0842-1

- [4] Bergström, Bo. 2004
Effektiv visuell kommunikation.
Carlsson Bokförlag, Sverige, ISBN 91-7203-681-8

6.2 Webbplatser

- [5] Macromedia Flash - ActionScript Dictionary : Contents of the dictionary
www.adobe.com/support/flash/action_scripts/actionscript_dictionary/
(Acc. 2006-08-11)

- [6] Flash: 99% Bad (Alertbox)
<http://www.useit.com/alertbox/20001029.html> (Acc. 2006-08-11)

- [7] Scrolling and Scrollbars (Alertbox)
<http://www.useit.com/alertbox/20050711.html> (Acc. 2006-08-11)

- [8] Flash 99 % Good
<http://www.flash99good.com/> (Acc. 2006-08-11)

- [9] ISO 9241-11, standarden för användbarhet
<http://www.santai.nu/artiklar/iso.htm> (Acc. 2006-08-11)

- [10] Vad är användbarhet?
<http://www.interakt.nu/> (Acc. 2006-08-11)

- [11] It May Flash, but It Doesn't Streak
<http://www.clickz.com/showPage.html?page=838701> (Acc. 2006-08-11)

- [12] Adobe Flash & SWF
http://en.wikipedia.org/wiki/Macromedia_Flash (Acc. 2006-08-21)
<http://en.wikipedia.org/wiki/Swf> (Acc. 2006-08-21)
- [13] A Quiz Designed to Give You Fitts
<http://www.asktog.com/columns/022DesignedToGiveFitts.html> (Acc. 2006-08-21)

6.3 Forum och inspirationskälla

- [14] Ultrashock
<http://www.ultrashock.com> (Acc. 2006-08-22)
- Orange Project – Sandra Weimar
<http://www.orange-project.com> (Acc. 2006-08-22)

7 Sökord

A

Actionscript.....2, 6, 8
 användartest15, 23, 24
 Användbar28
 Användbarhet1, 2, 3, 4, 5, 7, 9
 användbarhetsfel3, 10, 26

B

bakåtknapp3, 12, 20, 24, 25
 Bitmap6
 Bruce Tognazzini's15, 23

F

feedback ..10, 11, 13, 15, 16, 17, 22,
 23
 Flash ..1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12,
 13, 15, 18, 20, 27, 28, 29
 flash-baserade1, 2, 4, 5, 6, 9, 10, 11,
 12, 13, 14, 15, 18, 20, 26, 27

G

grafik8, 17, 27
 gränssnitt4, 22
 GUI.....6, 11, 12

H

HTML6, 7, 8
 html-sidor10, 11, 12, 13

J

Jakob Nielsen5
 JavaScript6, 8, 13, 20, 21

L

laddningsindikerare ... 10, 15, 16, 17

M

Macromedia.. 1, 2, 1, 2, 3, 4, 5, 6, 7,
 8, 9, 10, 12, 13, 15, 18, 20, 27,
 28, 29

MDI 4

Microsoft 18

N

navigering..... 13, 22

Nerladdningstid 3, 10

P

preloader 10

Programmering 16

R

rullningslister..... 11, 12

S

ShockwaveFlash..... 6

skriva ut 12, 24

Steve Krug..... 9

U

UIScrollBar 18

Utvecklingsmiljön 9

V,W

Vektorgrafik 6, 7

8 Bilagor

```
function checkPage() {  
    newPage = _root.page;  
    //hämtar variabeln från tidslinjen  
    //om variabeln page är ändrad, navigera till den  
sidan  
    if (oldPage != newPage) {  
        _root.meny[oldPage].gotoAndStop(1);  
        _root.meny[newPage].gotoAndStop(2);  
        _root.content.pages.gotoPage(newPage);  
    }  
    oldPage = newPage;  
}
```

```
onClipEvent (load) {  
    pageWidth = 900; //Varje delside är 900 pixlar  
}  
onClipEvent (enterFrame) {  
    this.checkPosition(); //Funktion innuti movieclip  
}
```

```
on (release) { //Funktion till knapp 3 på webbplatsen  
    //laddar html-sida i den gömda ramen.  
    getUrl("3.htm", "bottom");  
    //sätter variabel för nästa sektion.  
    _root.meny[_root.page].gotoAndStop(1);  
    _root.page = 3;  
    this.gotoAndStop(2);  
}
```

```
on (rollOver) {
    this.gotoAndStop(2);
}
on (rollOut) {
    if (_root.page != 3) {
        this.gotoAndStop(1);
    }
}
```

```
//Kollar positionen på varje delside och flyttar till rätt
position
function checkPosition() {
    distance = this.newPosition - _x; //Avstånd kvar att
flytta

    if (Math.abs(distance) > .1) {
        _x += distance / 5;
    }
}

//Funktion som mottager en extern signal för att flytta till
ny delside
function gotoPage (page) {
    this.newPosition = -1 * (page-1) * this.pageWidth;
}

//Exempelvis: om delside = 1, nya _x positionen blir 0
//Om delside = 2, nya _x position blir -1 * pageWidth (900
pixlar i detta fallet)
```

```
on (release) { //Utskrift av textfält på delside 6
    var skrivut = new PrintJob();
    //skapar en instans av printjob klassen
```

```
var lyckad = skrivut.start();

//kollar om dialogrutan för printning har startat
var pageAdded = false;

if (lyckad) {
    //lägger till vilken textbox som skall
    printas, men skalar först textboxen för att passa på ett a4
    tempwidth =
    _root.content.pages.page6.textruta6._width;
    _root.content.pages.page6.textruta6._width
= 550;
    tempheight =
    _root.content.pages.page6.textruta6._height;
    _root.content.pages.page6.textruta6._height
= 800;
    pageAdded =
    skrivut.addPage(_root.content.pages.page6.textruta6);
}
if (pageAdded) {
    skrivut.send();
    //skickar vidare till skrivaren
}
delete skrivut;
_root.content.pages.page6.textruta6._width =
tempwidth;
_root.content.pages.page6.textruta6._height =
tempheight;
//rensar cacheminnet
}
```