

EXAMENSARBETE I DATAVETENSKAP, GRUNDNIVÅ 15 HP

SurfaLätt - .NET och COM baserad webbläsare för personer med vissa funktionsnedsättningar



SAMMANFATTNING

Personer med vissa kognitiva funktionsnedsättningar har ibland svårt att använda en vanlig webbläsare. Detta gör internet mindre tillgängligt för denna grupp. Med hjälp av en anpassad webbläsare med förenklat gränssnitt och funktionalitet finns förhoppning om att användarvänlighetsbarriärer för internetanvändning hos dessa personer skall minska och att användningen skall kunna vara mer självständig.

Som teknisk målplattform används Microsoft Windows med Internet Explorer och utvecklingen genomfördes inom .NET med C#. Ett mål med den tekniska implementationen var att till så stor del som möjligt utnyttja redan installerad programvara på målsystemen och därmed minimera behovet av programuppdateringar, vilket skulle göra programmet mer framtidssäkert, samt minimera mängden administration av programmet.

Tester genomförda på Västmanlands läns landstings Datatek på Handikappcentrum i Västerås gav positivt resultat. Användarna kunde fokusera på de webbsidor de besökte och personalen kunde snabbt ställa in programmet för användaren.

ABSTRACT

People with certain cognitive disabilities have sometimes difficulties using a standard web browser. This makes the Internet less accessible to this group. With the help of a custom web browser with simplified user interface and features there is hope that the usability barrier around internet usage can be lowered and allowing this group of user to use the Internet in a more independent manner.

Microsoft Windows with Internet Explorer has been used as technical platform and the development was carried out using .NET with C#. One of the aims of the technical implementation was to use already installed software on the target system at the largest degree possible, thus minimizing the need for software updates, which makes the application more future-proof, and also minimizing the amount of administration needed.

Tests conducted at the Västmanland County Council Datatek on Handikappcentrum in Västerås gave positive results. Users could focus on the websites they visited, and staff could quickly set up the application for the user.

Datum: 22 februari 2012

Utfört vid: Mälardalens högskola Västerås (MDH) och Datateket Handikappcentrum Västerås.

Handledare vid MDH: Federico Ciccozzi

Kontakt vid Datateket: Eva-Karin Alm

Examinator: Ivica Crnkovic

FÖRORD

Detta examensarbete har varit mycket intressant ur både teknisk och användarvänlighets synpunkt. Det hade dock varit ogörligt utan stöd och hjälp från ett antal personer. Jag vill därför tacka min handledare Federico Ciccozzi och min kontakt på Datateket, Eva-Karin Alm. Ett tack riktas också till testanvändarna på Datateket. En särskilt tack riktas till Shahbaz Tahmasebi för ovärderligt stöd.

Västerås, januari 2012

Martin Persson

INNEHÅLL

| | |
|---|-----------|
| Kapitel 1 INLEDNING | 1 |
| 1.1 Bakgrund..... | 1 |
| 1.2 Syfte | 1 |
| 1.3 Begrepp | 1 |
| 1.4 Befintliga lösningar..... | 2 |
| Kapitel 2 ANALYS | 3 |
| 2.1 Interaktion med programmet med hjälp av tryckkontakter..... | 3 |
| Kravanalys..... | 3 |
| Teknisk aspekt | 3 |
| 2.2 Visning av webbsidor | 4 |
| Kravanalys..... | 4 |
| Teknisk aspekt | 4 |
| 2.3 Navigation till och mellan webbsidor | 4 |
| Kravanalys..... | 4 |
| Teknisk aspekt | 5 |
| Kapitel 3 IMPLEMENTATION | 6 |
| 3.1 Val av plattform och språk..... | 6 |
| Operativsystem | 6 |
| Teknisk plattform | 6 |
| 3.2 Klassen SurfaLätt..... | 6 |
| 3.2.1. SurfaLätts gränssnitt..... | 7 |
| Gränssnittets beståndsdelar..... | 8 |
| 3.2.2. Rendering av webbsidor | 8 |
| 3.2.3. Navigation till och mellan webbsidor | 9 |
| Nya fönster | 9 |
| Stänga fönster | 10 |
| 3.2.4. Favoriter | 11 |
| 3.2.4.1. Lagring av favoriter | 11 |
| Datamedlemmar I Favorit..... | 11 |
| Exportera och importera favoriter | 11 |
| 3.2.4.2. Skapa favorit | 12 |
| Hämta favoritikon..... | 12 |
| 3.2.4.3. Visa och hantera favoriter | 13 |
| Visa favoriter..... | 13 |
| Hantera favoriter | 13 |
| 3.2.5. Utskrift | 14 |
| 3.2.6. Inställningar..... | 15 |
| Lagring av inställningar..... | 15 |
| Ändra inställningar..... | 15 |
| Kapitel 4 RESULTAT | 16 |
| 4.1 Prov med testgrupp..... | 16 |
| Fel upptäckta vid testning | 16 |
| Önskemål om nya funktioner..... | 16 |
| 4.2 Slutprodukt | 16 |

| | |
|-----------------------------------|-----------|
| Kapitel 5 SLUTSATSER | 17 |
| 5.1 Datavetenskaplig aspekt | 17 |
| 5.2 Användarvänlighetsaspekt..... | 17 |
| Kapitel 6 REFERENSER | 18 |
| Bilaga A 19 | |

Kapitel 1

INLEDNING

1.1 Bakgrund

Personer med vissa fysiska eller kognitiva funktionsnedsättningar har ibland svårt att använda en vanlig webbläsare. Svårigheterna kan härledas både till webbsidors utformning och till webbläsarens gränssnitt. Webbsidors utformning kan dock enbart kontrolleras av den som publicerar webbsidan, den enda lösningen på att en webbsida är alltför svåränvänd är att en lättare version skapas. Svårigheter orsakade av själva webbläsarprogrammet kan dock minskas genom att ett alternativt program används.

Försvårande inslag kan vara:

- Rörigt gränssnitt med för många funktioner
- Små och otydliga symboler/ikoner
- Åtkomst till exempelvis favoriter måste gå via rullgardinsmenyer som kräver interaktion för att visas
- Pop-upp rutor och webbsidor som öppnas i nya fönster

Ett annat problem är att reklam på en webbsida kan leda användaren bort från den webbsida som avsett att användas till en helt annan webbsida med okänt innehåll.

1.2 Syfte

Syftet med arbetet är att utveckla ett webbläsarprogram som har lämpliga anpassningar som kan underlätta för personer med vissa funktionsnedsättningar att använda internet.

Arbetet utförs i samarbete med Datateket på Handikappcentrum, Västmanlands läns landsting.

1.3 Begrepp

.NET - ett programvarumiljö som består av två huvuddelar:

Common Language Runtime (CLR), en programvarumiljö där .NET program kan köras oberoende av systemplattform och programmeringsspråk

Framework Class Library (FCL), en samling komponenter som kan användas av .NET program och som underlättar utvecklingsarbetet.

Programvara utvecklad inom .NET har god kompatibilitet eftersom den exekveras inom CLR. De färdiga komponenterna i FCL ger god funktionalitet eftersom de är beprövade. De gör även utvecklingsarbetet enklare genom att tillhandahålla grundläggande funktionalitet [12].

C# - Objektorienterat programmeringsspråk med C-liknande syntax. Används för att utveckla .NET program [12].

COM - *Component Object Model*. En teknologi som gör det möjligt för mjukvarukomponenter under Microsoft Windows att kommunicera med varandra. Detta gör att funktionalitet från befintliga program kan användas i andra program[6].

Favorit - En sparad genväg för snabb åtkomst till en särskild webbsida. Kallas även i andra sammanhang för Bokmärke. Benämningen Favorit har valts efter som det är den termen som används i Internet Explorer, vars teknologi används i arbetet.

Kontroll - En grafisk komponent med särskild funktionalitet, både grafisk och annan [11].

URL - Uniform Resource Locator, en adress till ett webbdokument.

Knapp - I denna rapport syftar ordet knapp på en komponent i ett grafiskt användargränssnitt. För en fysisk knapp på tangentbordet används ordet tangent.

1.4 Befintliga lösningar

I samband med detta arbete har inte något program särskilt avsett för personer med funktionsnedsättning kunnat hittas. Tilläggsprogram för att förstora text och bild på webbsidor eller för att spärra åtkomst till andra webbsidor finns, dessa utgår dock ändå från en standardwebbläsare.

Program med webbläsarfunktion särskilt anpassad för barn finns det dock ett antal. Några exempel på dessa är:

- KIDO'Z [5]
- KidZui [13]
- Magic Desktop [3]
- Zoodles [4]

Gemensamt för dessa är färgglatt gränssnitt med stor mängd rörelse och animationer samt ett stort antal ikoner. Detta sammantaget ger ett rörigt och distraherande intryck. Gränssnittets utformning och färgsättning är också barnslig, vilket inte passar alla användare.

Kapitel 2

ANALYS

2.1 Interaktion med programmet med hjälp av tryckkontakter

Kravanalys

Personer med vissa funktionsnedsättningar använder ibland särskilda s.k. tryckkontakter för att interagera med en dator. Dessa är betydligt större än vanliga tangenter och kan manipuleras med andra kroppsdelar än fingertopparna. Oftast används två tryckkontakter. Metodiken är att den ena tryckkontakten styr och den andra väljer, dvs. den första kontakten växlar mellan olika alternativ och den andra används för att bekräfta val. En implementation av styrmetodiken med tryckkontakter blir väldigt lik den som tillämpas av synskadade. Genom att använda TAB-tangenten för att hoppa mellan länkar och retur för att bekräfta val går det att använda webbsidor utan att använda mus. Denna metod används med framgång av många synskadade, den får dock anses ha ett flertal nackdelar både generellt och specifikt för SurfaLätts målgrupp:

- Moderna webbsidor innehåller ofta många länkar och detta innebär att det krävs många knapptryckningar för att komma till önskad länk.
- Med hjälp av tangentkombinationen "SHIFT+TAB" är det möjligt att hoppa bakåt bland länkarna. Detta är inte möjligt med enbart två tryckkontakter. Resultat blir att om användaren råkar hoppa för långt går det inte att gå tillbaka utan att stega igenom alla länkar tills man gått hela varvet runt.
- Aktivt innehåll som t.ex. Adobe Flash [1] saknar stöd för att med tangentbordet markera exempelvis länkar. Detta gör att det inte går att ersätta musstyrning för Flash-innehåll som inte är byggt för tangentbordsstyrning.

Detta sammantaget gör att SurfaLätt i sig kan underlätta väldigt lite när det handlar om användning av styrmetodiken med två tryckkontakter.

En alternativ metod är att använda tangentbordstryckningar till att styra muspekaren. Genom att använda fem tryckkontakter, en för varje riktning och en för vänsterklick, går det att ersätta musen med tryckkontakter. Detta är dock en helt annan styrmetodik än den med två tryckkontakter och är därför inte etablerad hos användarna, och måste därför endast ses som ett experimentellt komplement.

Teknisk aspekt

Tryckkontakter genererar tangentbordstryckningar, oftast "TAB" och "Retur". Dessa tryckningar måste därför hanteras i programmet för att styra det.

Varianten med styrning av muspekaren med hjälp av tangentbordstryckningar kräver att dessa omvandlas till förändringar av muspekarens position samt generering av musklick.

2.2 Visning av webbsidor

Kravanalys

Den viktigaste och mest grundläggande funktionen för programmet är att visa webbsidor. Webbsidor kan förekomma både i rent html-format eller med aktivt innehåll som t.ex. Adobe Flash.

Funktionaliteten för att visa webbsidor består dels i att en visningsfält måste finnas i programmets grafiska gränssnitt och funktionalitet för rendering av webbsidor och deras innehåll.

Teknisk aspekt

Att från grunden utveckla en renderingsmotor för webbsidor är både onödigt och mycket tidskrävande. Istället bör funktionalitet från en redan existerande webbläsare användas.

Eftersom programutvecklingen sker inom .Net finns Internet Explorers funktionalitet direkt tillgängligt i form av en kontroll. Att använda Internet Explorer ger ett flertal fördelar:

- Modern rendering av webbsidor
- Aktivt innehåll, t.ex. Adobe Flash, kan visas med hjälp av plug-in:er redan installerade för Internet Explorer
- Internet Explorer uppdateras kontinuerligt både med avseende på rendering av webbsidor och på säkerhet. Detta gör att Surfalätts livslängd förlängs avsevärt utan att uppgraderingar behöver göras.

Genom att använda en lista över tillåtna webbplatser som jämförs med aktuell Uniform Resource Locator (URL) innan en ny sida laddas kan spärr mot okända webbsidor uppnås.

2.3 Navigation till och mellan webbsidor

Kravanalys

I omgivningen kring personer med kognitiva funktionsnedsättningar kan det finnas personer, t.ex. i form av familjemedlemmar eller personal, som är oroliga för att personen, av misstag eller med vilja, skall ta del av "olämpliga" webbsidor, t.ex. med våldsamt eller pornografiskt innehåll. Även om oron kan vara helt obefogad kan den leda till att en person med funktionsnedsättning av omgivningen hindras från att använda internet [2]. Genom att implementera funktionalitet som förhindrar att andra webbplatser än på förhand tillåtna kan besökas finns förhoppning om att oron hos personer i omgivningen skall lindras, vilket leder till en mer tillåtande attityd jämt emot internetanvändning. En spärr mot okända webbsidor kan även förhindra att en användare av misstag navigerar till en helt annan webbplats än avsett, vilket kan verka förvirrande och ge svårighet att återvända till avsedd webbsida. Funktionalitet för att spärra okända webbsidor kräver att så fort en länk aktiveras måste mål URL:en analyseras. Funktionen måste dock vara valbar så att även full tillgång till alla webbplatser är möjlig.

Navigering till webbsidor måste ske på ett enkelt och tydligt sätt. Bakåt och framåt ikoner bör finnas samt ikoner för att gå till favoritsidor. Dessa ikoner bör vara valbara eller ha något med webbplatsen att göra. Möjlighet till att skriva in URL måste även finnas, både för att användare som kan skriva in URL:er själv skall kunna göra det och för att personer som hjälper till med administration av programmet skall kunna skapa favoriter. Ett textfält för inmatning av URL måste dock gå att gömma via inställningar, detta för att minimera

antalet detaljer i gränssnittet i de fall användaren inte har nytta av dem och enbart blir distraherad av dem.

Teknisk aspekt

Genom att jämföra en lista över tillåtna webbplatser med den URL som ska laddas kan otillåtna webbsidor spärras.

Tydligt gränssnitt med ikoner för favoritsidor kan antingen skapas statiskt eller dynamiskt. Ett statiskt gränssnitt är enklare att utveckla men blir mer begränsat i antalet snabbikoner till favoritsidor. Ett dynamiskt gränssnitt innebär däremot ingen begränsning av antalet snabbikoner till favoritsidor.

Varje snabbikon måste kunna föras med en bild att associera webbplatsen med. En metod för att automatiskt få en bild som har med webbplatsen att göra är att använda webbplatsens "favicon". Favicon är en ikonfil som webbplatser anger att de vill att webbläsaren ska använda för webbplatsens lokala bokmärke eller favoritnotering. Genom att låta programmet ladda ner faviconen och använda den för snabbikonen får man en lämplig bild utan att leta.

Favoritwebbplatser och deras bild måste lagras i konfigurationsfiler för programmet.

Kapitel 3

IMPLEMENTATION

Implementationen av programmet byggs på C#, .NET och COM med Microsoft Windows som målplattform. Utvecklingsarbetet skedde med hjälp av Microsoft Visual Studio 2008 [10].

3.1 Val av plattform och språk

Operativsystem

Valet Microsoft Windows som målplattform var enkelt. Majoriteten av Datatekets datorer använder Microsoft Windows XP. Windows är även det dominerande operativsystemet bland övriga datorer i världen och får anses vara dominerande bland potentiella användares privata datorer.

Alternativa persondatoroperativsystem som Linux och Mac OS X får anses vara i minoritet och Datatekets maskinpark, samt senare Windows-versioner, får anses vara utgångspunkt [14].

Tekniskplattform

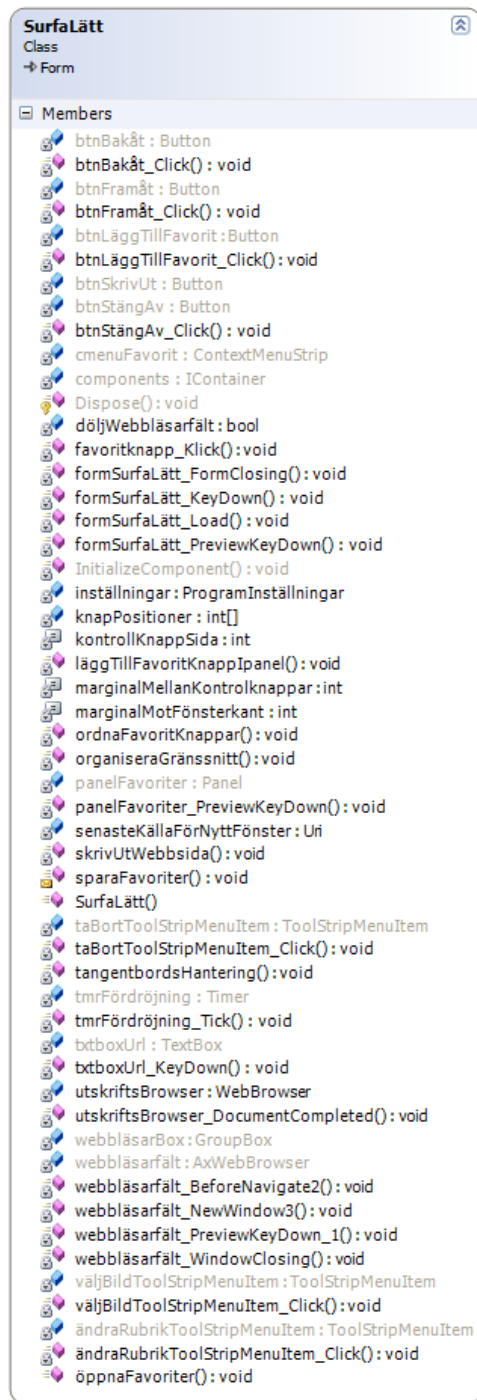
Valet av .NET som ramverk för programmet grundar sig på förhoppning om god kompatibilitet och snabb programutveckling. Eftersom .NET försätter att utvecklas efter att utvecklingsarbetet av SurfaLätt avslutats så kommer framtida förbättringar inom .NET även SurfaLätt till fördel.

Valet av C# är i sig inte särskilt viktigt. C# är dock ett av de mest dominerande och utvecklade språken inom .NET, vilket gör det till ett naturligt val.

Ursprungligen avsågs att enbart använda komponenter ur .NET. Under utvecklingsarbetet upptäcktes dock en brist i funktionaliteten (se 3.2.3) hos .NET-kontrollen WebBrowser. Istället användes Microsoft Web Browser som är en COM-komponent definierad i systemfilen shdocvw.dll. Microsoft Web Browser utgör en grundsten i webbläsaren Internet Explorer där den tillhandahåller både rendering av webbsidor och navigation. Denna funktionalitet kan importeras till andra applikationer via COM. I SurfaLätt görs detta i form av en `AxSHDocvW.AxWebBrowser` kontroll [7].

3.2 Klassen SurfaLätt

Den klass som utgör huvuddelen av programmet är klassen SurfaLätt, som ärver av standardklassen Form. En instans av SurfaLätt startas när programmet exekveras och majoriteten av användarupplevelsen sker inom SurfaLätt:s grafiska del.



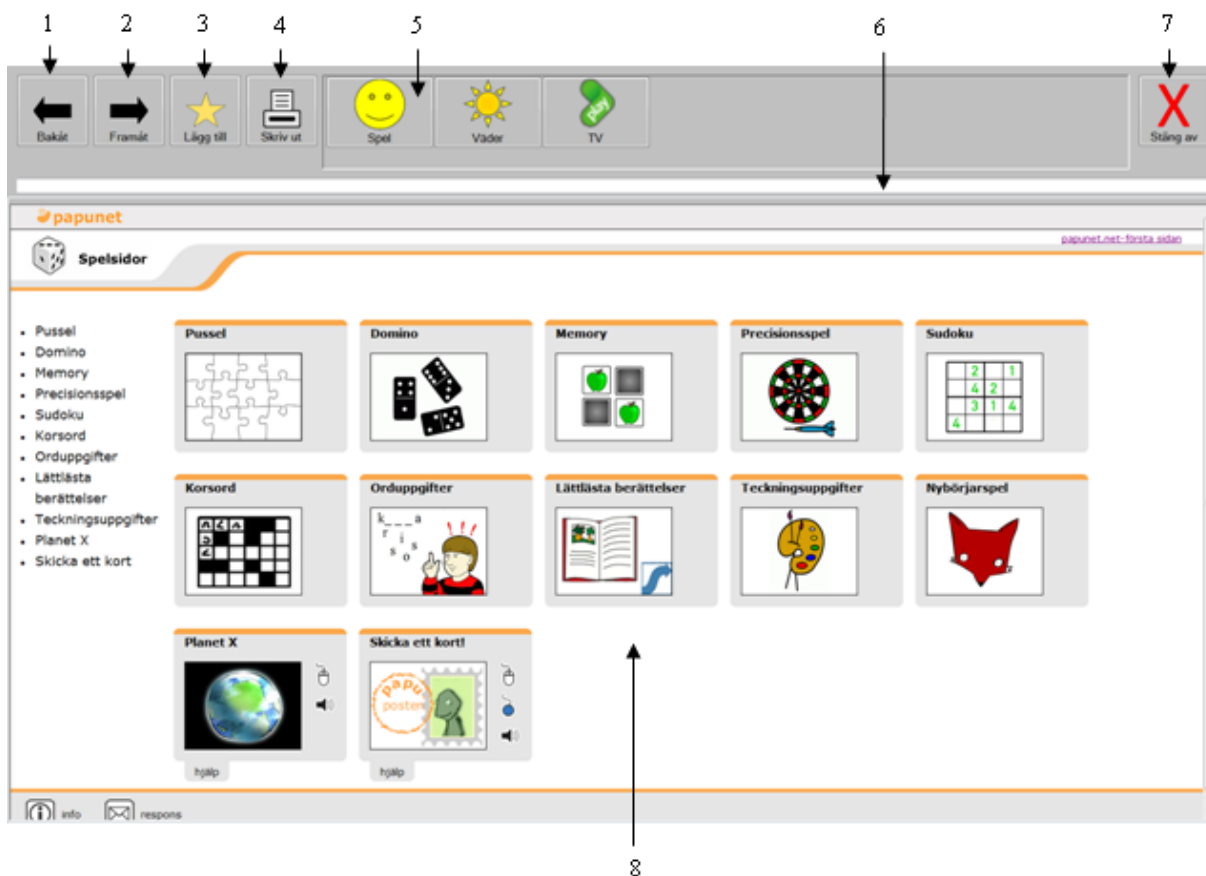
Figur 1. Klassdiagram för klassen SurfaLätt

3.2.1. SurfaLätts gränssnitt

Det grafiska gränssnittet SurfaLätt bygger på Windows Forms [8] som är en del av .NET. Gränssnittets design har formgivits med fokus på tydlighet och enkelhet. Programmets kontrollknappar är försedda både med text och symboler. Gränssnittets utseende anpassar sig automatiskt efter skärmupplösningen på den miljö det körs genom att metoden `organiseraGränssnitt` anropas vid programstart. Med hänsyn till de grafiska komponenternas storlek bör dock 800*600 pixlars bildskärmsupplösning vara att betrakta som minimiupplösning för att gränssnittet skall behålla sin avsedda utformning. Praktiskt

test vid just denna bildskärmsupplösning visar att gränssnittet vid sådana förhållanden behåller sin huvudsakliga utformning. För göra användandet av programmet mer fokuserad så täcker SurfaLätt:s gränssnitt hela bildskärmen. Detta uppnås genom att sätta klassens ärvda egenskaper `FormBorderStyle` till `None` och `WindowState` till `Maximized`.

Gränssnittets beståndsdelar



Figur 2 Gränssnitt, med detaljer, för SurfaLätt

1. Bakåt, knapp för att navigera bakåt till förra webbdokumentet. Button `btnBakåt`
2. Framåt, knapp för att navigera framåt till förra webbdokumentet. Button `btnFramåt`
3. Lägg till, knapp för att lägga till ny favorit. Button `btnLäggTillFavorit`
4. Skriv ut, knapp för att skriva ut aktuellt webbdokument. Button `btnSkrivUt`
5. Favoritfält, panel innehållande knappar för favoriter. Panel `panelFavoriter`
6. Adressfält, textfält för manuell inmatning av URL. `TextBox txtboxUrl`
7. Stäng av, knapp för att stänga programmet. Button `btnStängAv`
8. Webbält, fält för visning av webbsida. `AxSHDocvW.AxWebBrowser` `webbläsarfält`

3.2.2. Rendering av webbsidor

För att visa webbsidor används kontrollen `webbläsarfält` (Figur 2 detalj 8), som är av typen `AxSHDocvW.AxWebBrowser`. Denna kontroll hanterar majoriteten av alla

nätverkskommunikation och all rendering av webbinnehåll. Eftersom den använder teknologi från Internet Explorer kan funktionalitet från plug-in program som t.ex. Adobe Flash användas.

3.2.3. Navigation till och mellan webbsidor

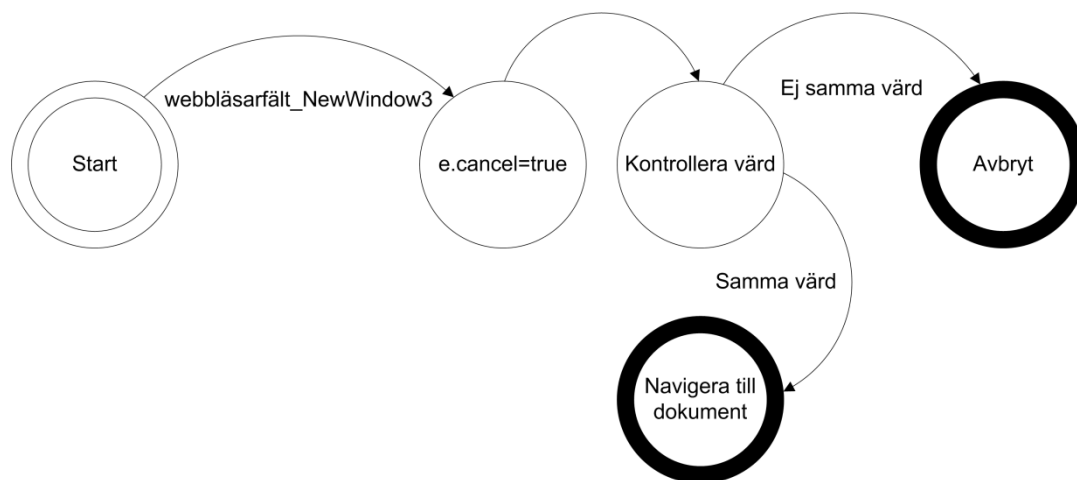
Navigation till en webbsida eller webbdokument kan påbörjas på tre sätt

- Användaren klicka på en favorit (del i Figur 2 detalj 5)
- Användaren skriver in en webbadress i adressfältet (Figur 2 detalj 6) och trycker "Retur"
- Användaren klickar på en länk i ett webbdokument

I de två första fallen påbörjas navigeringen genom att metoden `webbläsarfält.Navigate` anropas med en URL som argument.

Navigering initierad av att användaren klickar på en länk i ett webbdokument kontrolleras alltid av metoden `webbläsarfält.BeforeNavigate2` som anropas genom att `webbläsarfält` skapar en `BeforeNavigate2`-händelse. Om programmet konfigurerats så att enbart webbplatser som finns som favorit får visas genomförs kontroll, detta kontrolleras av flaggen `inställningar.navigationsSpärr`. Kontrollen består i att värd delen av målwebbsidans URL jämförs med värd delen av alla favoriters URL:er. Om ingen av favoriternas värd matchar målwebbsidans blockerats navigationen.

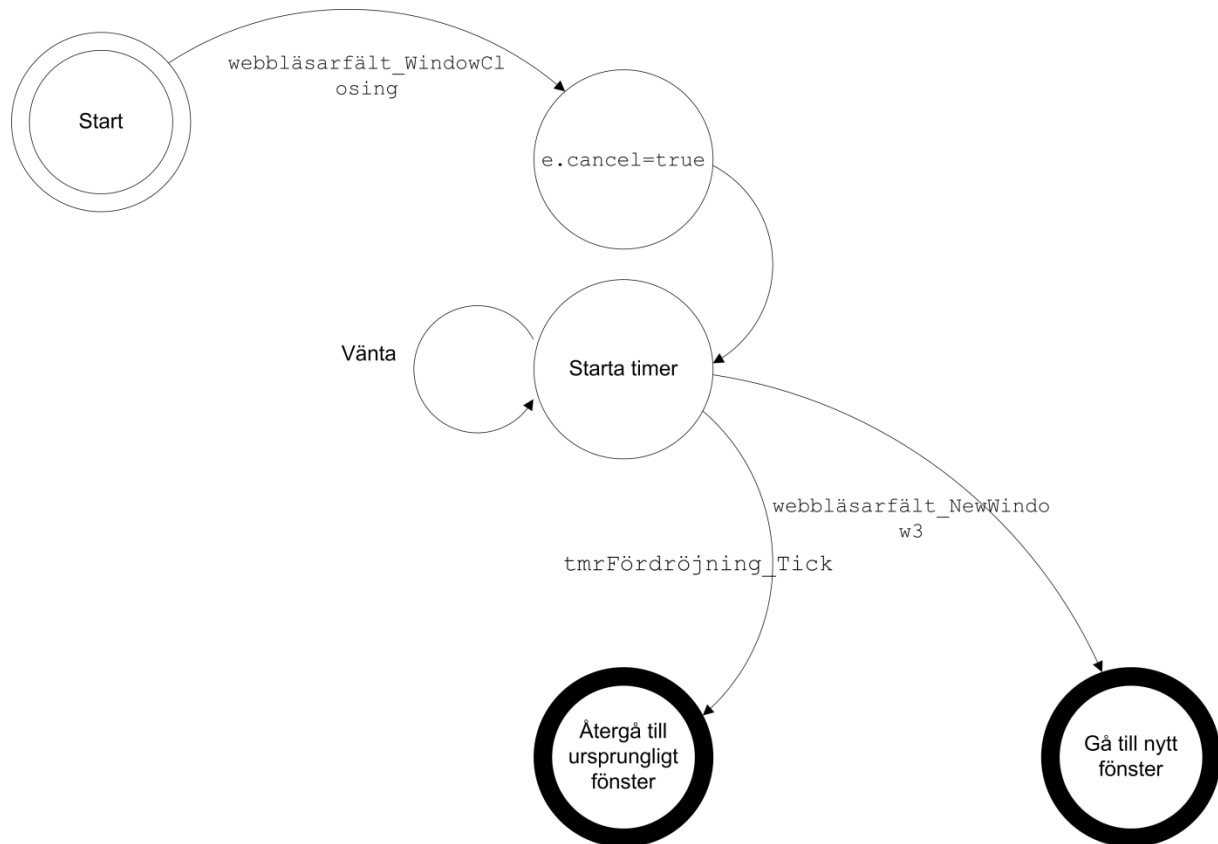
Nya fönster



Figur 3

Om en webbsida vill öppna ett nytt fönster generera `webbläsarfält` en `NewWindow3`-händelse, varpå metoden `webbläsarfält_NewWindow3` anropas. Metoden kontrollerar alltid att webbdokumentet som ska öppnas i det nya fönstret finns på samma värd som den ursprungliga webbplatsen. Om kontrollen ger positivt resultat påbörjas navigation till webbdokumentet genom att metoden `webbläsarfält.Navigate` anropas. Detta innebär att det nya fönstrets innehåll öppnas i samma `webbläsarfält` som ursprunget. Genom att parametern `e.cancel` sätts till `true` förhindras att ett nytt fönster alls öppnas. URL:en till det ursprungliga webbdokumentet sparas i variabeln `senasteKällaFörNyttFönster`.

Stänga fönster



Figur 4

Om innehållet för ett nytt fönster öppnats i webbläsarfält och webbdokumentet vill stänga sitt fönster genereras webbläsarfält en `WindowClosing`-händelse, varpå metoden `webbläsarfält_WindowClosing` anropas. Detta avbryts genom att parametern `e.cancel` sätts till `true`. Istället startas timern `tmrFördröjning`, som har en intervall på 1 sekund. När 1 sekund passerat aktiverar `tmrFördröjning` händelsen "Tick" varpå metoden `tmrFördröjning_Tick` anropas. Metoden anropar i sin tur `webbläsarfält.Navigate` med `senasteKällaFörNyttFönster` som argument, vilket leder till att det webbdokument som det nya fönstret öppnades ifrån öppnas.

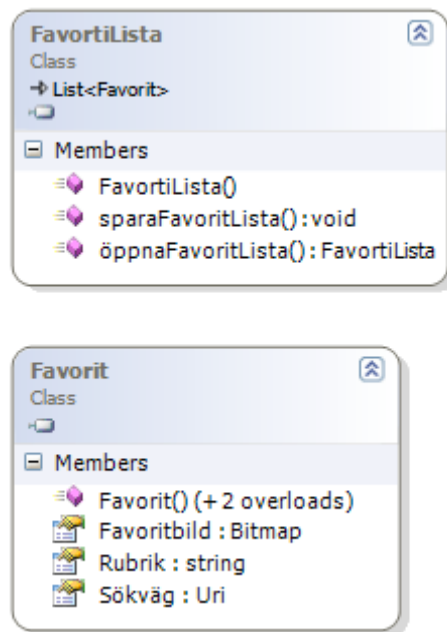
Orsaken till att en fördröjning används är pga. ett specialfall som upptäcktes vid testningen (se 4.1). En webbsida öppnade ett nytt fönster, från detta fönster öppnades i sin tur ett till fönster och det första stängdes. Genom att använda en fördröjning från det att händelsen `WindowClosing` skapas till dess att det ursprungliga webbdokumentet öppnas med hjälp av `senasteKällaFörNyttFönster` så hinner webbläsarfält påbörja navigering till det nya fönstret. Genom att i `webbläsarfält_NewWindow3` alltid anropa `tmrFördröjning.Stop` avbryts navigationen till ursprungsdokumentet.

Som tidigare nämnts användes .NET kontrollen `WebBrowser`. Denna kontroll saknar metod för att hantera att ett webbdokument vill stänga ett fönster. Kontrollen stängde då sig själv vilket ledde till att den blev oinitierad. Detta medförde att programmet kraschade. På grund av detta bytte kontrollen ut mot en `AxSHDocvw.AxWebBrowser` kontroll som ingår i COM-komponenten `Microsoft Web Browser`.

3.2.4. Favoriter

3.2.4.1. Lagring av favoriter

Favoriter lagras i programmets baskatalog. Filformatet är baserat på serialiserad binärdata och använder filändelsen SurfaLätt-favorit (slf) (Microsoft Corporation, 2012). De serialiserade objekten i filen består av ett FavoritLista-objekt innehållande Favorit-objekt.



Figur 5 Klassdiagram FavoritLista och Favorit

När programmets startas körs klassen `SurfaLätt`:s metod `öppnaFavoriter`, därifrån anropas den statiska metoden `FavoritLista.öppnaFavoritLista`. Detta leder till att filen öppnas och innehållet deserialiseras. Metoden returnerar sedan ett `FavoritLista`-objekt innehållande filens favoriter.

När programmet stängs körs klassen `SurfaLätt`:s metod `sparaFavoriter`, därifrån anropas den statiska metoden `FavoritLista.sparaFavoritLista` med ett `FavoritLista`-objekt som argument. Denna metod arbetar åt motsatt håll som när filen favoriter laddas. Inparametern serialiseras och binär data skrivs till filen.

Under ett skede av utvecklingsarbetet fanns planer på att istället för att serialisera och spara `Favorit`-objekt använd `FavoritKnapp`-objekt. Eftersom klassen `FavoritKnapp` ärver av klassen `Button`, och `Button` inte är serialiserbar, så var inte detta möjligt.

Datamedlemmar i Favorit

Klassen `Favorit` har tre data medlemmar; `Favoritbild`, `Rubrik` och `Sökväg`. `Favoritbild` lagrar en bild som visas som en ikon för favoriten, `Rubrik` lagrar favoritens rubrik och `Sökväg` lagrar webbdokumentets URL. Alla dessa medlemmar lagras när favoriter sparas till fil. Eftersom även favoritens ikon sparas till filen uppstår aldrig några problem med sökvägar till separata bildfiler.

Exportera och importera favoriter

För att göra det lättare att flytta lagrade favoriter mellan olika datorer finns det en funktion för export och import av favoriter. Detta utförs på samma sätt som när startas och

stängs. Det är klassen `InställningarDialog` som initierar detta genom att anropa `öppnaFavoriter` respektive `sparaFavoriter`.

3.2.4.2. Skapa favorit

När användaren klickar på "Lägg till" (Figur 2 detalj 3) anropas metoden `btnLäggTillFavorit_Click` och ett nytt objekt av klassen `FavoritKnapp` skapas med webbsidans titel och URL som argument till konstruktorn. Konstruktorn skapar i sin tur ett objekt av typen `Favorit` genom att klassens konstruktor anropas med webbsidans URL och rubrik som argument. Dessa används för att tilldela datamedlemmarna `Rubrik` och `Sökväg` samt för att hämta en favoritikon.

Hämta favoritikon

De flesta av dagens webbplatser har en favicon. Det finns två metoder för att hitta en webbplats favicon, antingen finns dess sökväg angiven i ett webbdokumentets källkod eller så finns den i roten på webbplatsens filsystem (exempel: `http://www.dn.se/favicon.ico`). För att kunna hitta en favicon via ett webbdokumentets källkod krävs analys eller tolkning av dokumentets källkod. Då `SurfaLätt` inte implementerar någon egen HTML-tolk utan förlitar sig på en tillhandahållen av Internet Explorer skulle det krävas av åtminstone en sök och analys funktion enbart för att hitta en favicons sökväg via webbdokumentets källkod, vilket skulle addera exekveringstid. För att inte öka tidsåtgången för att hitta och hämta en favicon valdes metoden att hämta den från webbplatsens rot.

Med hjälp av ett objekt av klassen `WebClient` [9] (`System.Net.WebClient`) hämtas faviconen som ett fält av byte som sedan omvandlas till en `MemoryStream`. Eftersom en favicon ofta är av bildformatet `Ico` används klassen `Icon` för att först omvandla dataströmmen till `Icon` och sedan till en `Bitmap` som lagras för favoriten. Orsaken till att inte en `Icon` lagras är att användaren senare kan byta ut favoritens ikon. Genom att alltid konvertera till och lagra som `Bitmap` uppnås användning av ett enhetligt bildformat internt i programmet.

Om en webbplats saknar en favicon eller om något annat fel uppstår används en standardikon föreställande en jordglob (Figur 6).



Figur 6 Standardikon för favorit

3.2.4.3. Visa och hantera favoriter

Visa favoriter

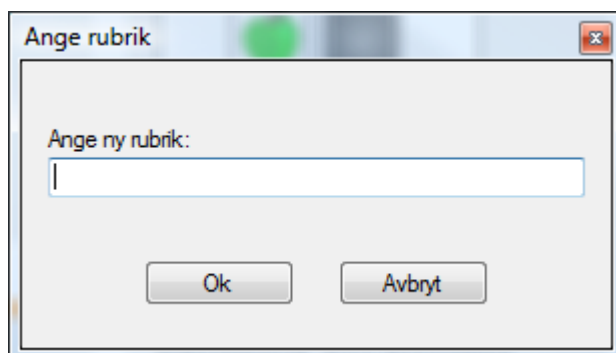
Favoriter visas i Favoritfältet (Figur 2 detalj 5) som är en kontroll typen Panel. Med hjälp av Panels egenskap `Panel.Controls` lagras Favoritknapparna, som är objekt av klassen `FavoritKnapp`. När programmet startas läggs Favoritknappar till en och en i Favoritfältet med hjälp av metoden `läggTillFavoritKnappIpanel`.

Hantera favoriter

För att hantera favoriter används en högerklicksmeny för varje `FavoritKnapp`. Menyn ger åtkomst till att ta bort favorit, ändra favoritens rubrik och ändra favoritens ikon. När metoden `SurfaLätt.organiseraGränssnitt` körs tilldelas egenskapen `ContextMenuStrip` för varje `FavoritKnapp` i `panelFavoriter.Controls` en referens till högerklicksmenyn `cmenuFavorit`. Detta sker dock enbart om programinställningarna medger att användaren får redigera favoriterna.

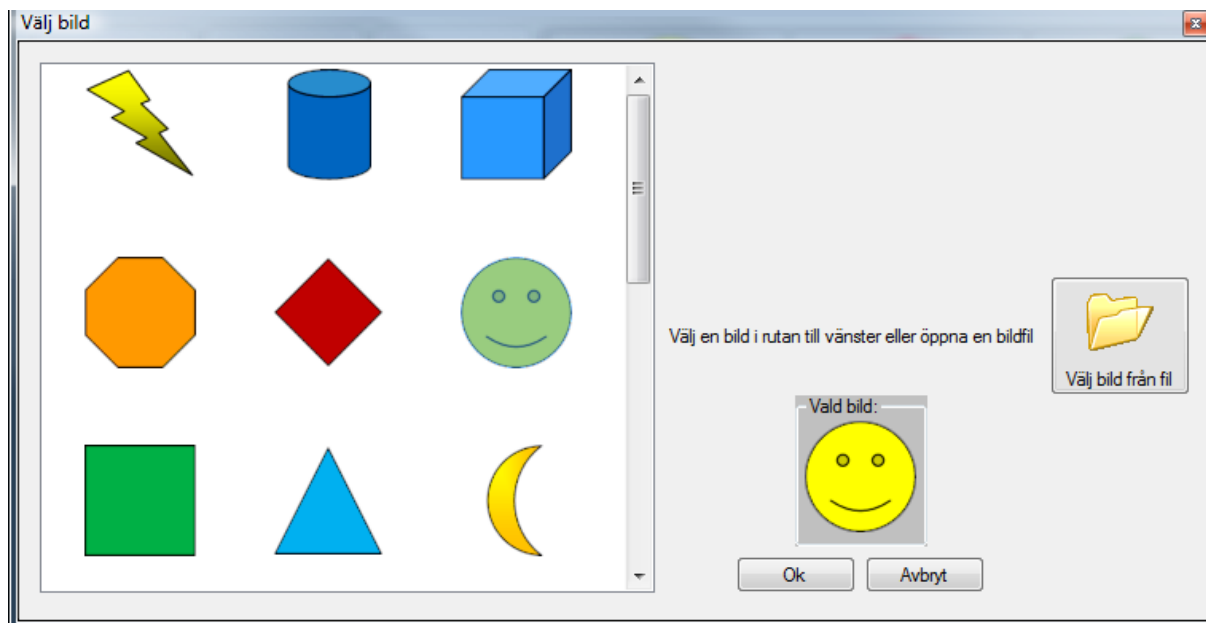
Vid borttagning av en favorit raderas ett `FavoritKnapp` objekt från `panelFavoriter.Controls` med hjälp av metoderna `Panel.Contains` och `Panel.Remove`.

Vid ändring av en favorits rubrik visas en dialogruta för att mata in ny rubrik. Dialogrutan är en kontroll av klassen `RubrikDialog` (Figur 7) som ärver av `Form`. För att garantera att alla rubriker som skrivs in får plats på Favoritknappen är textinmatningen i `RubrikDialog` begränsad till åtta tecken.



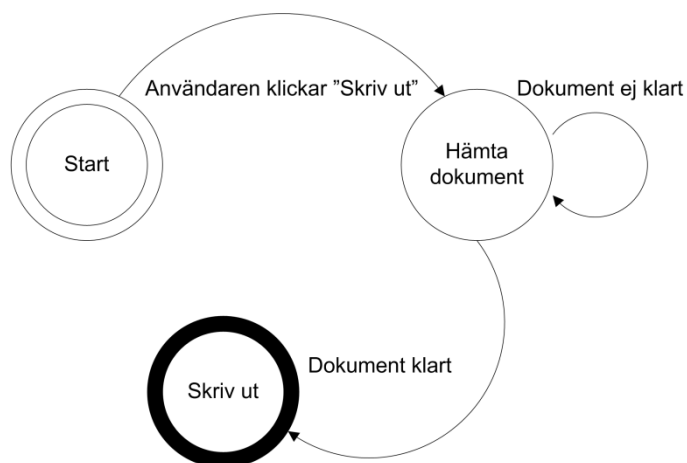
Figur 7 Inmatning av ny rubrik

Vid ändring av en favorits ikon visas en dialogruta (Figur 8) för att välja en ny bild. Dialogrutan är en kontroll av klassen BildDialog som ärver av Form. Via BildDialog går det att antingen välja en färdig ikon ur en bildbank inbyggd i programmet eller välja en bild från valfri bildfil som kan vara av formaten bmp, jpg, png, gif eller tif. Om en bild hämtad från fil är i någon dimension större än 70 pixlar genomgår den en omskalning neråt med bibehållet höjd/bredd förhållande. En förhandsgranskning av vald bild visas i rutan "Vald bild".



Figur 8 Dialogruta för att välja ikon

3.2.5. Utskrift



Figur 9

Genom att klicka på knappen Skriv ut (Figur 2 detalj 4) kan användaren skriva ut webbsidan. Då System.Windows.Forms.WebBrowser har en mycket lättanvänd metod för utskrift, WebBrowser.Print, används ett sådant objekt för utskrift. För att utskrift ska kunna ske måste WebBrowser-objektet först ladda webbdokumentet avsett för utskrift. Detta gör genom att anropa metoden WebBrowser.Navigate. Ett problem som dock kan uppstå är att om WebBrowser.Print anropas innan webbdokumentet laddats färdigt sker ingen utskrift.

För att lösa detta förses WebBrowser-objektet med en event-handler för händelsen DocumentCompleted. Först när event-handlern anropas påbörjas utskrift eftersom det då går att vara säker på att webbdokumentet laddats färdigt.

3.2.6. Inställningar

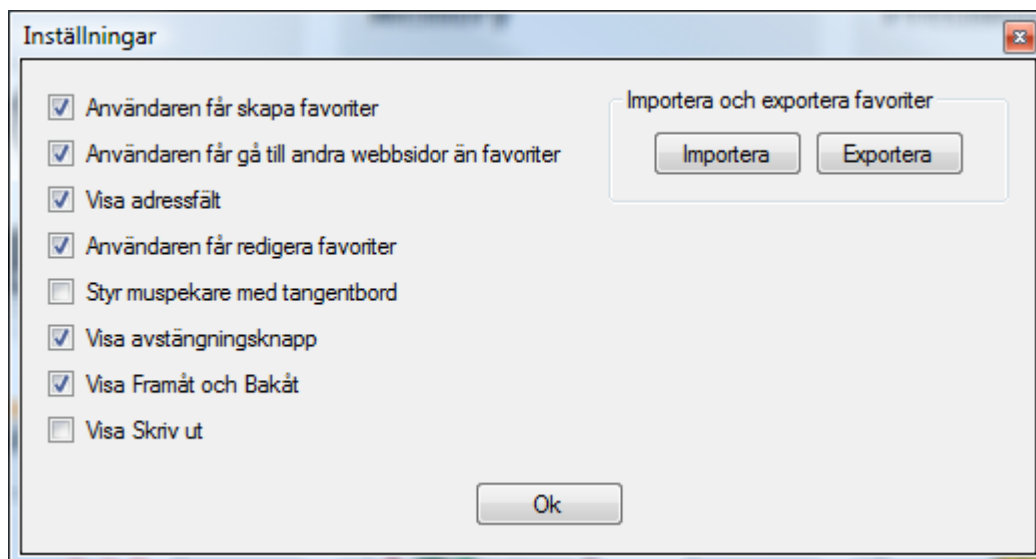
Programmets inställningar lagras i ett objekt av typen ProgramInställningar. Varje inställning representeras av en boolesk datamedlem i ProgramInställningar.

Lagring av inställningar

Programmets inställningar lagras som serialiserad binärdata i en fil med ändelsen dat. Inställningar öppnas och sparas till filen vid programstart respektive programslut.

Ändra inställningar

Ändring av inställningar görs via en dialogruta (Figur 10) som är en kontroll av klassen InställningarDialog som ärver av Form. För att dialogrutan ska visas krävs att användaren trycker tangentkombinationen "ALT+I". Detta gör det svårare att komma åt inställningar av missatag men samtidigt går det snabbt att komma åt dem när så önskas.



Figur 10 Dialog för programinställningar

Kapitel 4

RESULTAT

4.1 Prov med testgrupp

Ett test av programmet utfördes på Handikapcentrum i Västerås den 5 december 2011.

Testet utfördes tillsammans med personal och en grupp på 4 testanvändare. Användarna var i åldern 45-65 år och hade funktionsnedsättning i form av lätt till måttlig utvecklingsstörning. Gruppens medlemmar hade sedan tidigare använt dator i liten utsträckning och hade liten eller ingen vana av internetanvändning. Varje användare fick tillgång till en dator. Alla datorer som användes hade Windows XP som operativsystem samt .NET installerat. Inga särskilda hjälpmedel användes förutom i ett fall där en användare föredrog ett tangentbord med extra tydliga tecken.

Innan testanvändarna anlände fick personalen bekanta sig med programmet och ta del av dess dokumentation. Tillsammans med personalen fick testanvändarna sedan prova programmet. Två av de webbsidor som testades var <http://papunet.net/svenska/> och <http://www.youtube.com/>.

Reaktionerna på programmet var positiva. Kommentarer från användarna var:

- ”jättebra!”
- ”Så borde det vara på alla”

En observation som gjordes var att tack vare programmets enkelhet bemästrade personalen det mycket snabbt och kunde därför ge bra handledning till testanvändarna. Denna aspekt är inte att förbise, utan måste ses som viktig för ett bra resultat.

Fel upptäckta vid testning

Under testningen upptäcktes två fel, båda med anknytning till skapande och stängande av nya fönster. Detta föranledde förändringar i implementationen (se 3.2.3).

Önskemål om nya funktioner

Vid testningen uppkom från personalen önskemål om att knapparna för navigation framåt och bakåt skulle kunna döljas om så önskas. Detta implementerades senare som ett val i inställningsmenyn.

4.2 Slutprodukt

I samband med testet av programmet skapades en användarhandledning för programmet samt installationsmedia i form av en CD-ROM. När utvecklingsarbetet för programmet avslutats skapades en reviderad användarhandledning (se Bilaga A) och ett installationsmedium med en slutgiltig programversion.

Kapitel 5

SLUTSATSER

5.1 Datavetenskaplig aspekt

Utvecklingsarbetet för programmet SurfaLätt visar att en enkel webbläsare kan utvecklas uteslutande med .NET och COM teknologier. Det visar också att kontrollen `System.Windows.Forms.WebBrowser` har begränsad flexibilitet när det gäller hantering av nya fönster, vilket leder till att den inte är lämplig till mer avancerade tillämpningar.

5.2 Användarvänlighetsaspekt

Ett enkelt och tydligt gränssnitt i webbläsarprogrammet kan hjälpa personer med vissa funktionsnedsättningar att använda internet. Eventuell handledning från annan person blir också lättare om programmet har ett enkelt och lättförklarat gränssnitt.

Kapitel 6

REFERENSER

1. Adobe Systems Incorporated. (2012). *application programming | Adobe Flash Platform*. Hämtat från <http://www.adobe.com/flashplatform/> den 10 Januari 2012
2. Backlund, E., & Lundström, A. (2011). *Vi får inte skaffa Internet*. Högskolan i Borås.
3. EasyBits Group. (2012). *Magic Desktop - Låt din PC vara barnvakt!* Hämtat från <http://www.magicdesktop.com/sv/> den 10 Januari 2012
4. Inquisitive Minds, Inc. (2012). *Zoodles: A safe Kid Mode™ for every device*. Hämtat från <http://www.zoodles.com/home/marketing/index> den 10 Januari 2012
5. KIDO'Z Ltd. (2012). *KIDO'Z - Safe & Fun Internet For Kids*. Hämtat från <http://kidoz.net/> den 10 Januari 2012
6. Microsoft Corporation. (2011). *COM: Component Object Model Technologies*. Hämtat från Microsoft Corporation: <http://www.microsoft.com/com/default.aspx> den 30 December 2011
7. Microsoft Corporation. (2012). *About the Browser*. Hämtat från MSDN: [http://msdn.microsoft.com/en-us/library/aa741313\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa741313(v=vs.85).aspx) den 11 Januari 2012
8. Microsoft Corporation. (2012). *System.Windows.Forms*. Hämtat från MSDN: [http://msdn.microsoft.com/en-US/library/system.windows.forms\(v=VS.80\).aspx](http://msdn.microsoft.com/en-US/library/system.windows.forms(v=VS.80).aspx) den 12 Januari 2012
9. Microsoft Corporation. (2012). *WebClient Class*. Hämtat från MSDN: [http://msdn.microsoft.com/en-us/library/system.net.webclient\(v=VS.80\).aspx](http://msdn.microsoft.com/en-us/library/system.net.webclient(v=VS.80).aspx) den 10 Januari 2012
10. Microsoft Corporation. (2012). *Visual Studio 2008 Editions - Microsoft Visual Studio*. Hämtat från <http://www.microsoft.com/visualstudio/sv-se/products/2008-editions> den 10 Januari 2012
11. Skansholm, J. (2008). Grafiska användargränssnitt. i J. Skansholm, *Skarp programmering med C#* (s. 295). Studentlitteratur.
12. Skansholm, J. (2008). Vad är C# och .NET? i J. Skansholm, *Skarp programmering med C#* (ss. 5-7). Studentlitteratur.
13. The Internet for Kids Inc. (2012). *KidZui - The Internet for Kids*. Hämtat från <http://www.kidzui.com/> den 10 Januari 2012
14. W3Counter. (2012). *Global Web Stats*. Hämtat från W3Counter: <http://www.w3counter.com/globalstats.php?year=2011&month=10> den 2 Januari 2012

Bilaga A

Användarhandledning för SurfaLätt

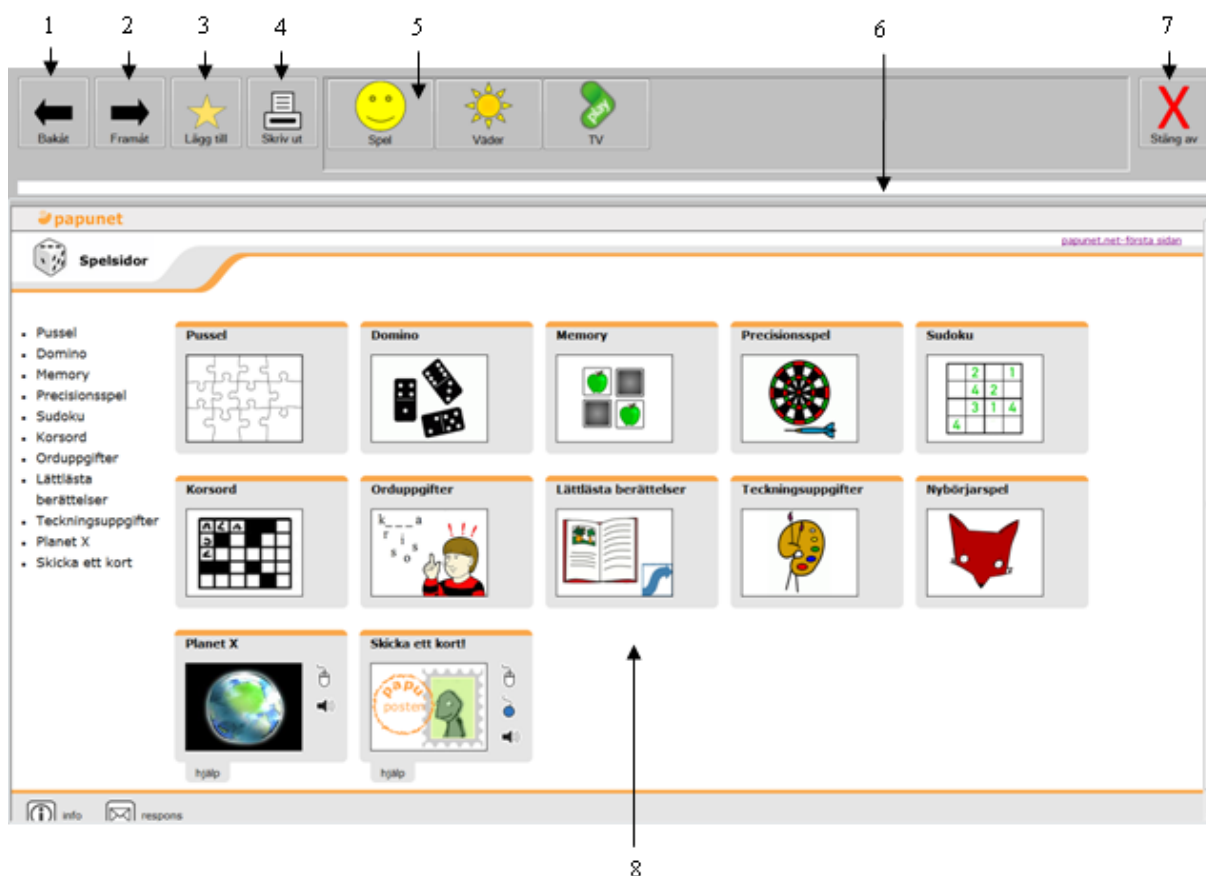
Starta SurfaLätt

Hitta programgruppen SurfaLätt på Startknappen, klicka på SurfaLätt-genvägen.



Programmets innehåll

Programmets utseende kan variera beroende på inställningar. Så här kan programmet se ut när det används.



1. Bakåt, går tillbaka till förra sidan
2. Framåt, går framåt igen till senaste sidan
3. Lägg till favorit, skapar en ny favorit för aktuell webbsida i Favoritfältet(4)
4. Skriv ut, skriver ut webbsidan på datorns standardskrivare
5. Favoritfält med favoritknappar
6. Adressfält, används för att skriva in webbadresser
7. Avstängningsknapp, stänger av programmet
8. Webbält, visar webbsidan

Använda SurfaLätt

Gå till webbsida

För att gå till en webbsida, klicka på en favoritknapp eller skriv en webbadress i adressfältet och tryck enter.

Lägg till favorit

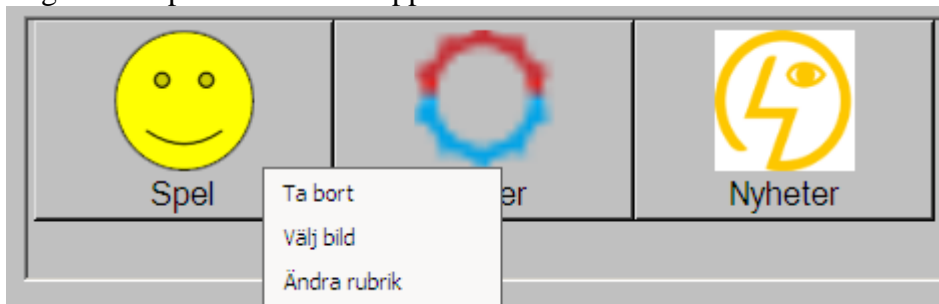
1. Gå till webbsidan
2. Klicka på knappen ”Lägg till”



Den nya favoritknappen får en bild och rubrik som automatiskt. Se längre ner i beskrivningen för att se hur man ändrar bild och rubrik

Ta bort favoritknapp

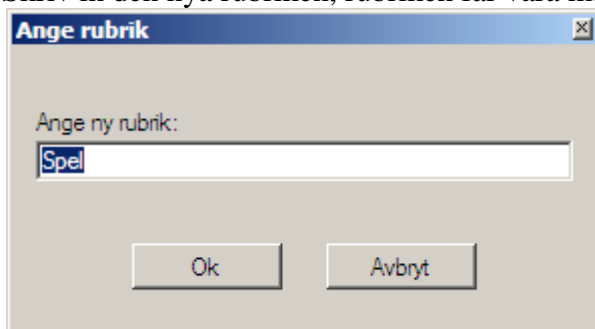
1. Högerklicka på den favoritknapp som du vill ta bort



2. Klicka på ”Ta bort” i menyn

Ändra rubrik på favoritknapp

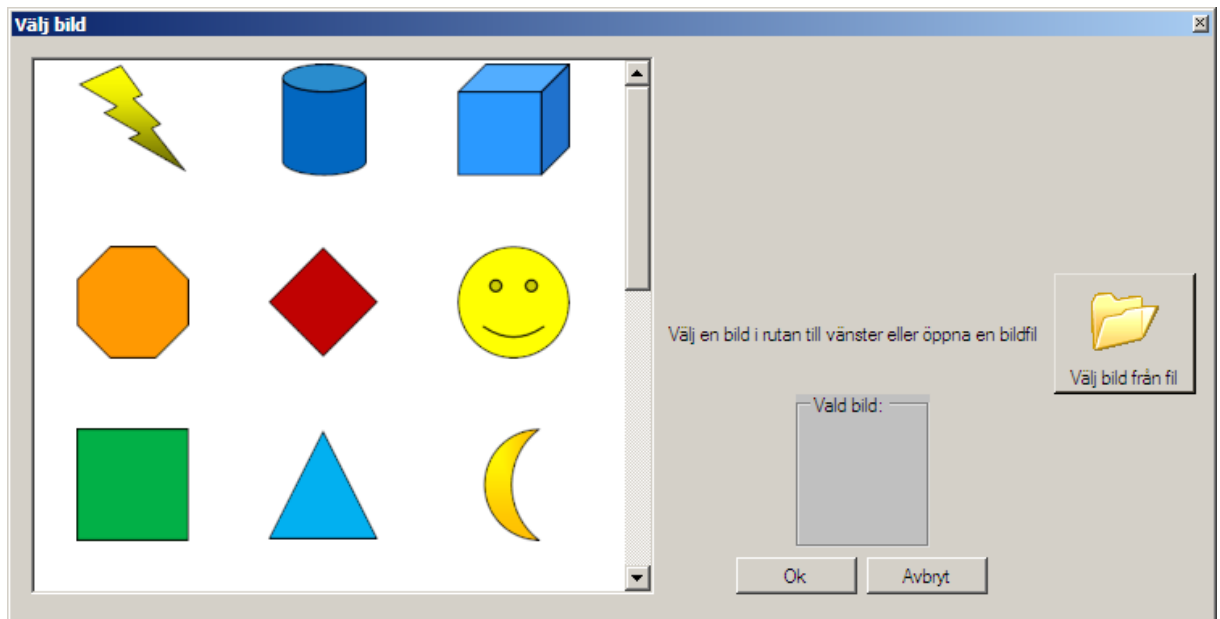
1. Högerklicka på den favoritknapp du vill ändra rubrik på
2. Välj ”Ändra rubrik” i menyn
3. Skriv in den nya rubriken, rubriken får vara max 8 tecken lång



4. Tryck enter eller klicka på ”Ok”

Ändra bild på favoritknapp

1. Högerklicka på den favoritknapp du vill ändra bild på
2. Klicka på "Välj bild" i menyn
3. Välj en bild i den vänstra rutan eller välj en bild från en fil. Den valda bilden visas i rutan "Vald bild"

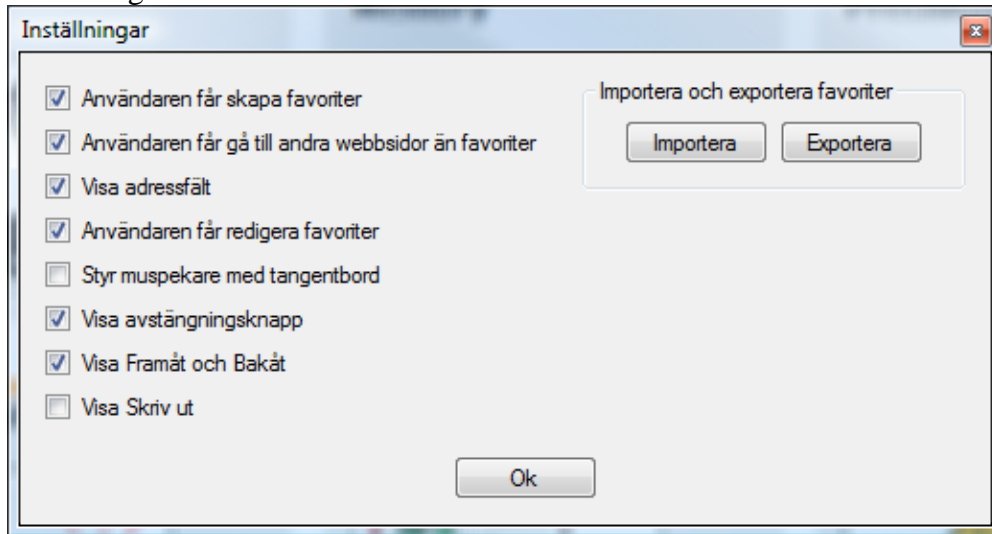


4. Klicka på "Ok"

Inställningar

Visa inställningar

För att visa inställningarna, tryck ned tangenterna Alt och I samtidigt en gång. Inställningsrutan visas då:



Beskrivning av inställningar

”Användaren får skapa favoriter”: Knappen ”Lägg till” är synlig och det går att skapa nya favoriter.

”Användaren får gå till andra webbsidor än favoriter”: Om alternativet är **avmarkerat** så går det inte att gå till andra webbplatser än de som finns i favoritfältet.

”Visa adressfält”: Adressfältet är synligt.

”Användaren får redigera favoriter”: Om ikryssat så visas högerklicksmenyn om man högerklickar på en favoritknapp.

”Styr muspekare med tangentbord”: Muspekaren går att styra med hjälp av piltangenterna, tangenten Ctrl fungerar som vänster musknapp.

”Visa avstängningsknapp”: Avstängningsknappen för programmet är synlig.

”Visa Framåt och Bakåt”: Framåt och Bakåt knapparna är synliga

”Visa Skriv ut”: Utskriftknappen är synlig

Importera och exporterar favoriter

Via inställningsrutan går det att spara eller öppna favoriter för att exempelvis kopiera dem till en annan dator.

Klicka på Importera för att öppna en fil med favoriter.

Klicka på Exportera för att spara favoriterna till en annan fil.