

MDI – Multiple Document Interface

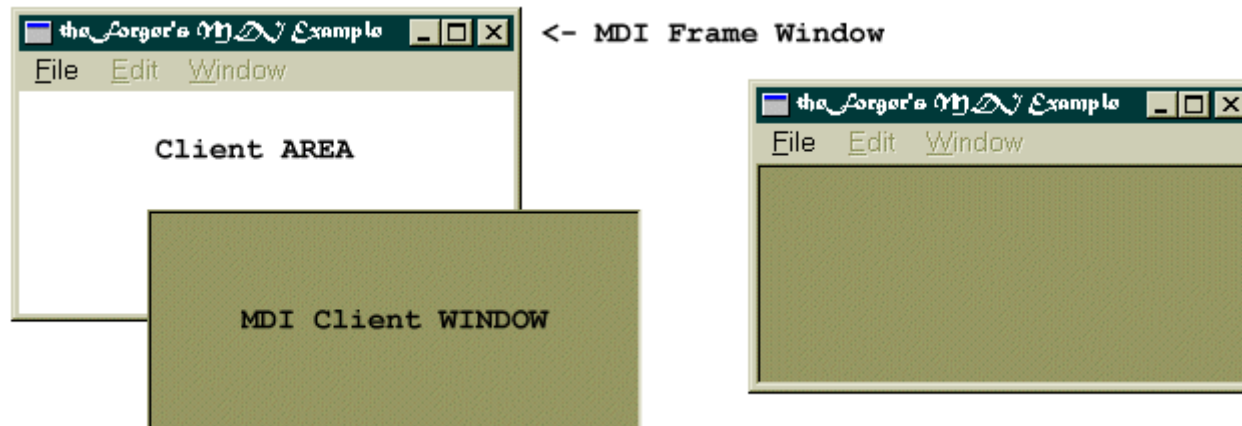
Sučelje za višestruke dokumente

Maja Štula

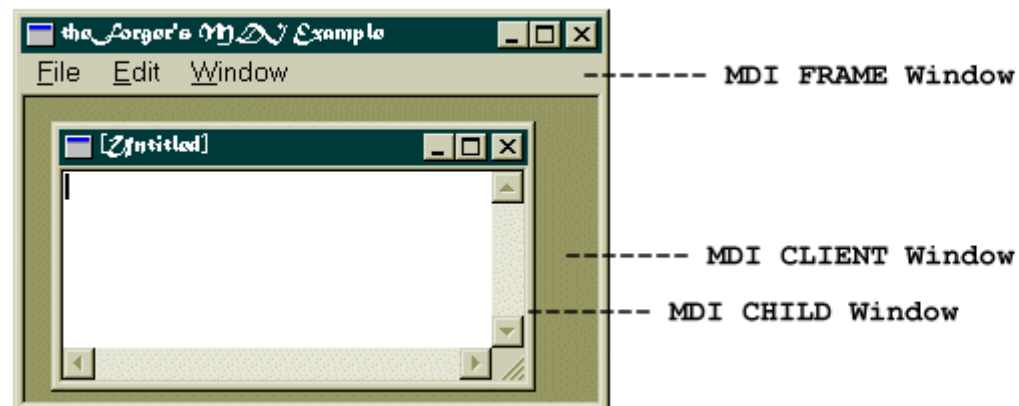
Ak. God. 2010/2011

MDI (*Multiple document interface*)

- Svaki prozor ima klijentsko područje (*Client Area*).
- Ponekad je čitav prozor klijentsko područje i ništa drugo, ponekad je klijentsko područje manje od prozora u kojem su još smješteni izbornici, trake sa botunima, naslovi i sl.



Complete MDI window ^
 The client AREA is covered up by the MDI
 Client WINDOW



MDI (*Multiple document interface*)

- Kod višestrukih dokumenata glavni prozor se naziva okvir (*Frame*). To je obično jedini prozor kod programa sa sučeljem za jednostruki dokument (SDI – *Single Document Interface*).
- Kod sučelja za višestruke dokumente postoji dodatni prozor koji se zove MDI klijentski prozor (*MDI Client Window*) i taj prozor je dijete (*child*) od glavnog prozora (*Frame*).

WIN32 API MDI

- Za razliku od klijentskog područja MDI klijentski prozor je potpuno zaseban prozor koji ima sva svojstva glavnog prozora kao što je npr. vlastito klijentsko područje, okvir (*border*) i sl.
- Kada se želi prikazati neki dokument šalje se poruka MDI klijentu. Novi željeni prozor se kreira kao dijete MDI klijenta, a ne glavnog prozora. Znači taj novi prozor je MDI dijete tj. dijete od MDI klijent prozora koji je pak dijete od glavnog prozora. MDI dijete će vjerojatno imati i vlastitu djecu prozore (child windows) kao što je npr. Edit kontrola.

C# MDI aplikacija

- Klasa `System.Windows.Forms.Form` ima svojstvo `IsMdiContainer`. Kada se to svojstvo postavi u `TRUE` takva forma postaje kontejner za *multiple document interface* (MDI) child forme.
- Sve MDI forme djeca prikazuju se unutar klijentskog područja forme koja je `MdiParent`.
- Kada se zatvara ta glavna forma svoj MDI djeci se generira događaj `Close`.
- MDI dijete forma povezuju se s `MdiParent` formom preko svojstva `MdiParent` koje treba sadržavati formu za koju želimo da bude `MdiParent`:
`mdidijete.MdiParent = this;` [PRIMJER](#)

C# MDI aplikacija

- Dohvaćanje MDI formi djece ide preko svojstva MdiChildren koje za MdiParent formu sadrži niz MDI formi djece:
- Primjer zatvaranja jednog MDI djeteta.

```
if (this.MdiChildren.Length > 0)
{
    Form tempChild = (Form)this.MdiChildren[this.MdiChildren.Length - 1];
    tempChild.Close();
}
```

- Vezivanje menija uz MDI djecu ide preko svojstva MdiWindowListItem:

```
this.menuStrip1.MdiWindowListItem = (ToolStripMenuItem)this.menuStrip1.Items[2];
```

C# MDI aplikacija

- Poravnanje MDI djece u C# radi se preko metode `LayoutMdi`.
- Ako se želi npr. kaskadno poravnanje, metoda se pozove s argumentom `MdiLayout.Cascade`:
`this.LayoutMdi(MdiLayout.Cascade);`
- `MdiLayout` je enumeracija sa članovima `ArrangeIcons` , `Cascade` , `TileHorizontal` , `TileVertical`.

WINFX MDI aplikacija

- WINFX (WPF) nema ugrađenu podršku za MDI aplikacije. Ukoliko želite razvijati klasičnu MDI aplikaciju trebate sami napraviti klasu prozora s takvom funkcionalnošću.
- MS se još od Windowsa 95 udaljava od MDI dizajnerskog pristupa i uvodi nove dizajnerske pristupe problemu višestrukih dokumenata (SDI (traditional browser windows), tabbed workspaces (modern browsers and IDEs), navigational models (web or WPF apps)).

SDI MDI

- Svaki dokument se otvara u novom top-level prozoru i dodaje u taskbar za olakšavanje navigacije.
- Sve instance aplikacije se pokreću unutar jednog te istog procesa jer bi kreiranje novog procesa za svaki otvoreni dokument previše opterećivalo CLR i bilo presporo.
- Sve instance aplikacije tj. svi otvoreni dokumenti mogu međusobno komunicirati. Kada se zatvore svi prozori završava i čitav proces.
- Također je moguće u izborniku dodavati elemente prema svakom otvorenom dokumentu.
- Aplikacije poput Office rade na ovakav način.

WINFX MDI aplikacija kao *tabbed* aplikacija

- Za razvijanje *tabbed* aplikacije koristi se klasa TabControl (u imenskom prostoru System.Windows.Controls , u assembly-ju PresentationFramework). [PRIMJER](#)

```
<TabControl Name="tabControl1" >  
    <TabItem Header="Prva">  
        <Frame Source="Page1.xaml"/>  
    </TabItem>  
    <TabItem Header="Druga">  
        <Frame Source="Page2.xaml"/>  
    </TabItem>  
</TabControl>
```

WINFX MDI aplikacija sa navigacijom

- Klasa Page je jedan od osnovnih navigacijskih elemenata WPF aplikacije zajedno sa klasom Hyperlink iz imenskog prostora System.Windows.Documents i klasom Frame.
- Funkcionalnost i korištenje Page elementa odgovara HTML stranici. To znači da Page klasa implementira pomicanje po stranicama (dokumentima) aplikacije slično kao kod HTML stranica na webu.
- Klasa Page može se koristiti i kao korijenski element XAML datoteke.
- Klasa Page nije prozor i nije izvedena iz osnovne klase prozora Window stoga se kao i sve druge kontrole mora smještati na prozor.

WINFX MDI aplikacija sa navigacijom

- Navigacijski elementi, poput Page elementa imaju svojstvo `NavigationService`, tipa klase `NavigationService`, kojim se implementira navigacija.
- Klasa `NavigationService` (u imenskom prostoru `System.Windows.Navigation`, u assembly-ju `PresentationFramework`) sadrži metode, svojstva i događaje koji podržavaju i omogućavaju navigaciju. Ova klasa se ne može nasljeđivati.

Navigacija

- Sadržaj kojem se pristupa navigacijom, može se dohvaćati preko instance navigacijskog objekta, poput stranice ili okvira i pozivanjem metode `Navigate` na navigacijskom objektu ili preko adrese sadržaja koji se dohvaća tj. njegovog URI-ja.
 - `NavigationService.Navigate(Object)`
 - `NavigationService.Navigate(Uri)`
 - `NavigationService.Navigate(Uri, Object)`
 - `NavigationService.Navigate(Uri, Object, Boolean)`

WINFX MDI aplikacija sa navigacijom

- Klasa NavigationService uključuje sposobnost dohvaćanja sadržaja koji može biti .NET Framework 3.x objekt i HTML datoteka. Uobičajeno je da se za navigaciju koristi objekt tipa Page.
- [PRIMJER:](#)

```
public partial class Page1 : Page
{
    public Page1()
    { InitializeComponent(); }
    void Idi(object obj, EventArgs ev)
    { Page2 druga = new Page2();
      this.NavigationService.Navigate(druga); }
}
```

WINFX MDI aplikacija sa navigacijom

- Kod XBAP aplikacije koja se izvodi u pregledniku, preglednik pamti povijest navigacije kroz dohvaćene stranice, pa se korisnik može pomicati naprijed ili nazad po već dohvaćenim stranicama.
- Međutim povijest navigacije u samostalnim aplikacijama treba biti implementirana u elementu aplikacije. Klasa Page nema implementiranu povijest navigacije. Kao navigatori, sa implementiranom poviješću navigacije, koriste se klasa Frame i klasa NavigationWindow iz imenskog prostora System.Windows.Navigation.
- Klasa NavigationWindow je izvedena iz klase Window te se može koristiti za glavni prozor aplikacije.

WINFX MDI aplikacija sa navigacijom

- Primjer:

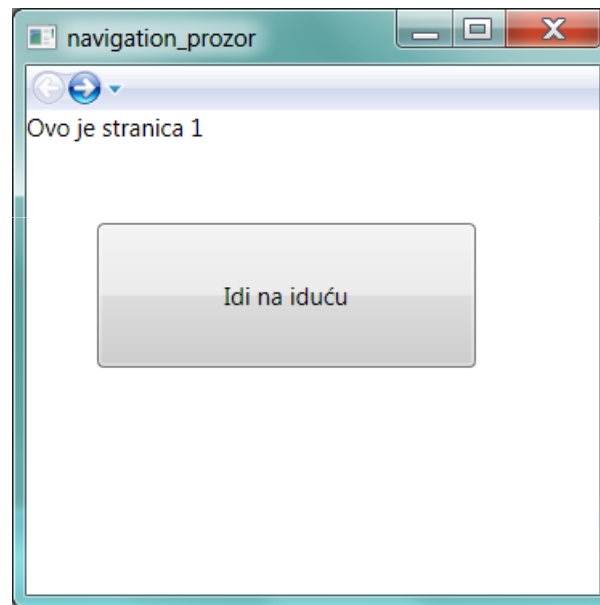
```
<Window x:Class="navigation_prozor.Window1"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="navigation_prozor" Height="300" Width="300">
  <Frame Content="Frame" Name="frame1"
        NavigationUIVisibility="Visible" Source="Page1.xaml">
    </Frame>
  </Window>
```

- Glavni prozor aplikacije sadrži okvir identificiran sa identifikatorom *frame1*. S obzirom da obični prozor tipa klase Window nema podržanu povijest navigacije, povijest navigacija je u ovom primjeru implementirana objektom tipa klase Frame.

WINFX MDI aplikacija sa navigacijom

- Okvir sadrži prvu stranicu definiranu sa *Page1.xaml* datotekom. Stranica se u okvir smješta preko svojstva `Source` kojim se postavlja i dohvaća URI trenutnog sadržaja kontrole.
- Povijest navigacije, podržana u kontroli `System.Windows.Controls.Frame`, ima grafičko sučelje kojim se korisnik može pomicati po stranicama aplikacije naprijed i nazad. Sučelje izgleda slično kao kod Internet preglednika, a smješta se na vrhu prozora ispod naslova.

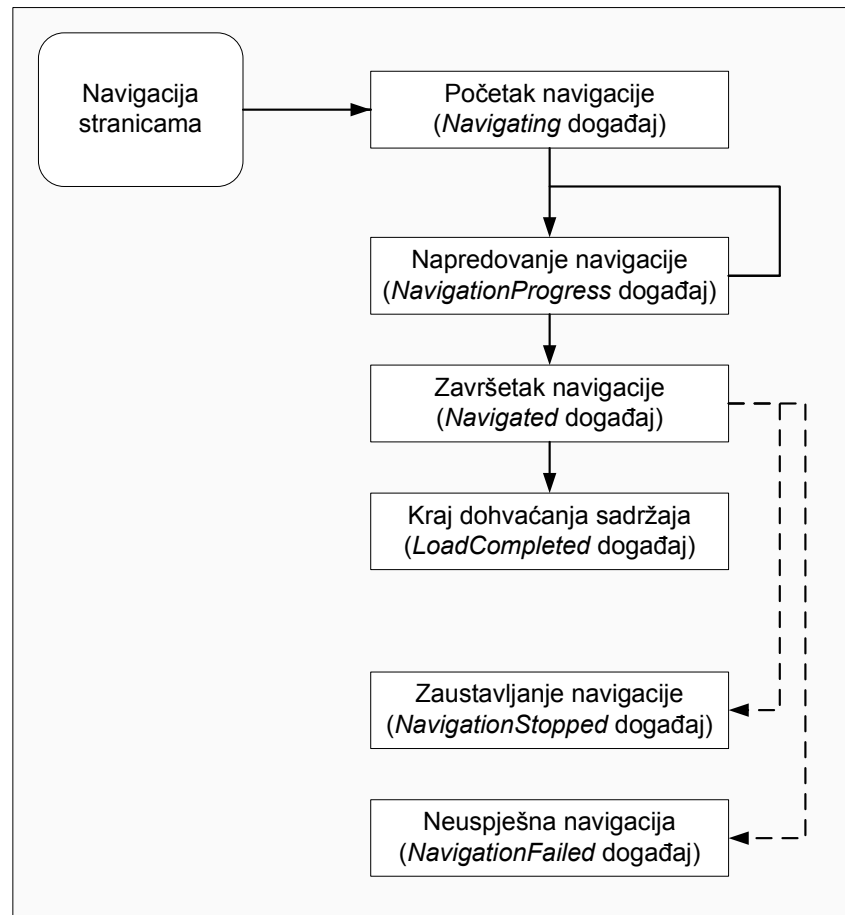
WINFX MDI aplikacija sa navigacijom



WINFX MDI aplikacija sa navigacijom

- Botuni za navigaciju naprijed i nazad po stranicama su automatski omogućeni ili onemogućeni u skladu sa mogućnostima navigacije tj. ako ne postoji sljedeća stranica u navigaciji, botun za dohvat sljedeće stranice je onemogućen, a ako postoji sljedeća stranica u navigaciji, botun za dohvat sljedeće stranice je omogućen.
- Svojstvo okvira NavigationUIVisibility može se postaviti na vrijednost Hidden ukoliko se ne želi prikazati sučelje za navigaciju.

WINFX MDI aplikacija sa navigacijom



- Događaji navigacije su prikazani na slici.