Medii vizuale de programare
Curs 8

Conf. dr.ing. GENGE Béla

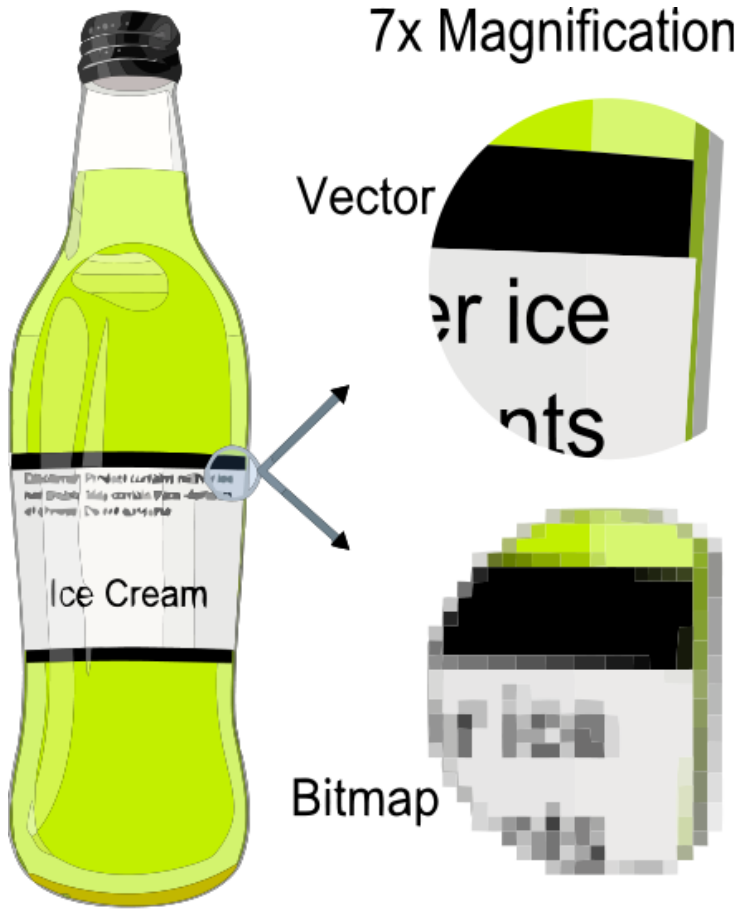
Universitatea "Petru Maior", Departamentul de Informatica
Tîrgu Mureș, Romania
bela.genge@ing.upm.ro

Windows Presentation Foundation (WPF)

- Generația următoare pentru crearea interfețelor utilizator (UI) în .NET.
- A fost introdus începând cu .NET framework versiunea 3.0.
- Permite crearea de conținut bogat în multi-media.
- Asigură separarea între UI (XAML) și logica aplicației (codul C#).
- Utilizează accelerarea hardware disponibilă.

Despre WPF

- Aplicațiile WPF sunt aplicații Direct3D (parte a DirectX).
- DirectX: O colecție de softuri și API-uri pentru implementarea aplicațiilor multi-media (jocuri în special).
- WPF utilizeaza placa grafica pentru crearea de UI moderne.
- Grafica în WPF este bazata pe grafica vectoriala.
  - Grafica vectoriala utilizeaza forme geometrice pentru descrierea imaginii.



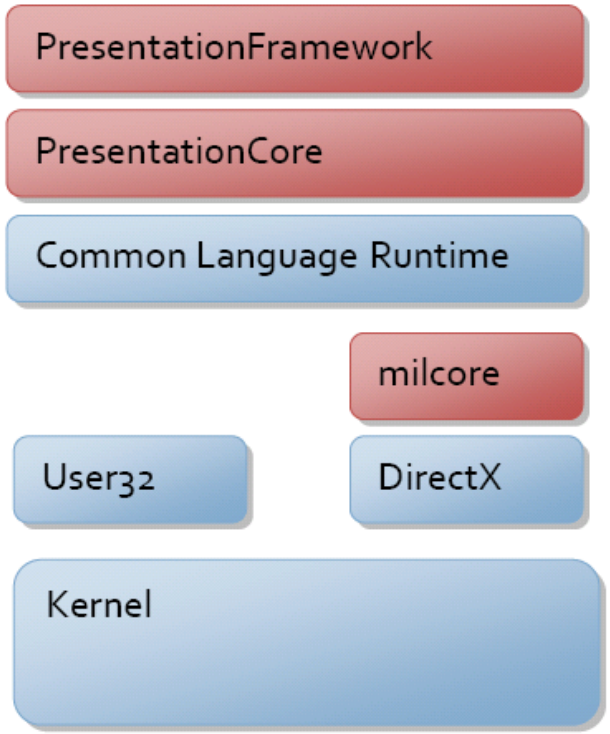
Sursa: Wikipedia.

Despre WPF

- WPF suportă animație pe bază de timp.
- Scenele sunt coordonate prin intermediul unui *storyboard*.
- Animația poate fi definită direct din XAML.
- Aplicațiile WPF pot încorpora audio și video.
- WPF permite definirea de *Template*-uri pentru schimbarea stilului UI.

Despre WPF

- WPF este de fapt o colecție de assembly-uri.
- `PresentationFramework.dll`: crează elementele superioare precum layout, stiluri, ferestre, etc.
- `PresentationCore.dll`: conține structuri de bază precum UIElement din care sunt moștenite toate elementele din `PresentationFramework.dll`.
- `milcore.dll`: Media Integration Library Core. Translatează apelurile dintre nivelele superioare și DirectX. Este practic motorul grafic al WPF.



Sursa: MSDN.

XAML

- XAML: eXtensible Application Markup Language.
- XAML este un limbaj declarativ: pune accent pe dimensiunea de proiectare a UI și ascunde detaliile de implementare.
- XAML descrie comportamentul și structura UI.
  - O diferență majoră față de Windows Forms și limbajul C#.
- Este un limbaj bazat pe XML; descrie o ierarhie de obiecte reprezentând controale grafice.

Crearea unei aplicații WPF

- MS Visual Studio: WPF Application.
- Se generează două fișiere XAML:
  - App.xaml
  - MainWindow.xaml
- Conținutul MainWindow.xaml:

```
<Window x:Class="WpfApplication1.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/pre
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="150" Width="350">
    <Grid>
    </Grid>
</Window>
```

Crearea unei aplicații WPF

- Adăugarea unui buton și tratarea evenimentelor.

```
<Window x:Class="WpfApplication1.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/pre
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="150" Width="350">
    <Button Width="250" Height="30" Click="OnClickButton1">Hel
</Window>
```

Crearea unei aplicații WPF

- Utilizarea unui Grid layout.

```
<Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="2*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="1*"/>
            <RowDefinition Height="2*"/>
        </Grid.RowDefinitions>
        <Button Grid.Row="1" Grid.Column="1" Click="OnPleaseCli
    </Grid>
```

Crearea unei aplicații WPF

- Grid.ColumnDefinitions și Grid.RowDefinitions.
- Valori disponibile pentru Width/Height:
  - Valoare: dimensiunea în pixeli.
  - Auto: dimensionare automată maximă.
  - *: dimensionare relativă față de alte componente (implicit în fața lui * se consideră val. 1).
- Exemple:
  - 0.5* și 0.5*: două coloane/rânduri de dimensiuni egale (se poate scrie și : */*, 2*/2*).
  - * și 3*: a doua coloană/rând de 3 ori mai mare.

UserControl

- Se pot crea controale complexe definite de utilizator.
- Exemplu: adăugare la proiect un user control cu un TextBox și un buton.
  - Noul control se descrie prin XML.
  - După compilare acesta este disponibil în ToolBox.
  - Se poate adăuga la fereastra principală ca și un control uzual.

ResourceDictionaries

- Permite crearea, redistribuirea și organizarea resurselor.
- Exemple de resurse:
  - String-uri.
  - Butoane.
  - Componente grafice.
  - ...
- O resursă reprezintă practic orice obiect declarat într-un XAML.
- În WPF resursele sunt identificate prin `x:Key`.
- Resursele sunt grupate în dicționare: `ResourceDictionaries`.

ResourceDictionaries

- Un dicționar se crează automat prin adăugarea la proiect (click dreapta pe numele proiectului).
- Mai multe dicționare se adaugă prin "ResourceDictionary.MergedDictionaries".
- Pentru a accesa dicționarele definite acestea trebuie adăugate la `App.xaml`:

```
<Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="MyDictionary.xaml"><
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
</Application.Resources>
```

Definirea stilurilor proprii

- Exemplu de definire a unui Brush.
- În fișierul de resurse se definesc următoarele:

```
<SolidColorBrush x:Key="MyBrush1" Color="#FF5A5A5A"/>
<SolidColorBrush x:Key="MyBrush2" Color="AliceBlue"/>
<SolidColorBrush x:Key="MyBrush3" Color="Magenta"/>
```

- Legarea (”Binding”) resurselor de controale:

```
<Window x:Class="WPFLabor.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/prese
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid Background="{Binding Source={StaticResource MyBrush2}}"
```

Utilizarea unei chei

- Exemplu de modificare a stilului butoanelor:
  - Se creează/descarcă/cumpără un stil (Metro Style for WPF Button): `https://gist.github.com/alimbada/3083937`
  - Se adaugă la un dicționar.
  - Se adaugă la buton:

```
<Button Style="{Binding Source={StaticResource MetroButton}}
```

- StaticResource vs. DynamicResource.

Aplicarea unui anume stil pentru toate controalele

- Modificarea stitului se poate realiza și global.
- Exemplu (MSDN): `https://msdn.microsoft.com/en-us/library/cc278069(VS.95).aspx`

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/pre
                    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Style TargetType="Button">
        <Setter Property="Background" Value="#FF1F3B53"/>
        <Setter Property="Foreground" Value="#FF000000"/>
        <Setter Property="Padding" Value="3"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="BorderBrush">
            <Setter.Value>
                <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                    <GradientStop Color="#FFA3AEB9" Offset="0"/>
                    <GradientStop Color="#FF8399A9" Offset="0.375"/>
                    <GradientStop Color="#FF718597" Offset="0.375"/>
                    <GradientStop Color="#FF617584" Offset="1"/>
                </LinearGradientBrush>
            </Setter.Value>
        </Setter>
    </Style>
```

Modern User Interface

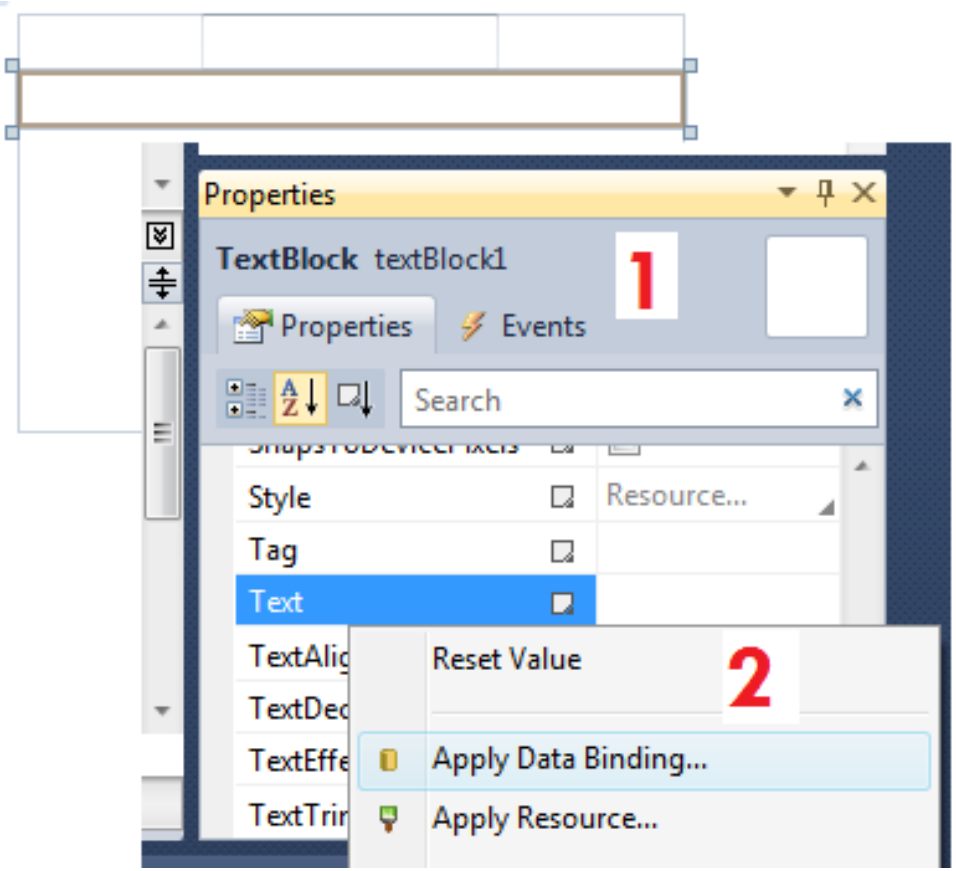
- Modern User Interface: O suită de stiluri pentru dezvoltarea aplicațiilor cu stil modern.
- Instalare: `https://visualstudiogallery.msdn.microsoft.com/7a4362a7-fe5d-4f9d-bc7b-0c0dc272fe31`
- După restart VS, se poate verifica la Tools -> Extensions and Updates dacă s-a instalat.
- În caz afirmativ se pot crea două tipuri de aplicații:
  - Modern UI WPF Application.
  - Modern UI WPF Navigation Application.

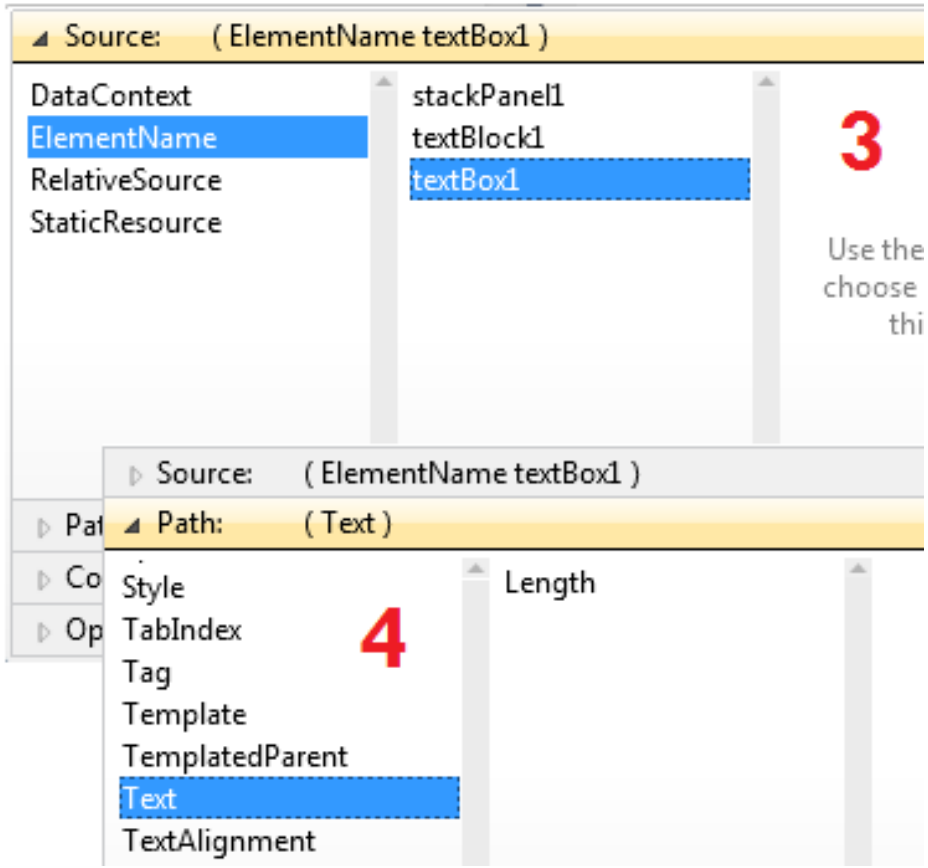
Binding

- În forma de dezvoltare tradițională a aplicațiilor .NET legarea obiectelor (sursă-destinație) se realiza prin evenimente și tratarea evenimentelor.
- WPF oferă o altă soluție ce pune accent pe cele două entități (sursa-destinația) și mai puțin pe eveniment.
- Soluția: **binding**. Componente:
  - **binding**.
  - **ElementName** - binding între controale (sursa).
  - **Source** - binding între obiecte.
  - **Path** - calea către proprietatea sursei.
  - **XPath** - calea către proprietatea sursei (ptr. surse XML).

Binding - exemplu 1

- Creare *StackPanel*.
- Adăugare *TextBox* și *TextBlock*.





Binding - exemplu 2

- Citire date dintr-un XML.
- Definire XML studenți (input.xml), binding cu *ListBox*.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Studenti>
        <nume>Ionescu</nume>
        <nume>Popescu</nume>
        <nume>Gheorghe</nume>
</Studenti>
```

- Definire resursă:

```
<XmlDataProvider x:Key="StudentiXML" Source="full-path/input.x
```

- Binding (pentru *ListBox*):

```
ItemsSource="{Binding Source={StaticResource StudentiXML},
 XPath=/Studenti/nume}"
```

Binding - exemplu 2

- Alternativ, definire XML studenți ca resursă.

```
<StackPanel.Resources>
    <XmlDataProvider x:Key="StudentiXML" >
        <x:XData>
            <Studenti>
                <nume>Ionescu</nume>
                <nume>Popescu</nume>
                <nume>Gheorghe</nume>
            </Studenti>
        </x:XData>
    </XmlDataProvider>
</StackPanel.Resources>
```

Răspunsul la evenimente

- În cazul în care există mai multe componente care doresc să fie notificate la apariția unui eveniment se poate utiliza *ObservableCollection* și interfața *INotifyPropertyChanged*.
- *ObservableCollection*: Va conține o listă de obiecte ce doresc să fie notificate.
- Interfața *INotifyPropertyChanged* trebuie implementată în clasele ce doresc să fie notificate.

Comenzi

- În programarea tradițională același cod poate fi executat din mai multe locuri. Ex: adăugare utilizatori din meniu, buton, etc..
  - Pentru fiecare va trebui tratat evenimentul și apelată aceeași metodă.
- WPF permite definirea metodei de tratare o singură dată și apelul acesteia din diverse locații prin *Commands* și *CommandBindings*.
- *Command* implementează interfața *ICommand* ce conține două metode:
  - *Execute()*;
  - *CanExecute()*;
- Există peste 100 de comenzi predefinite

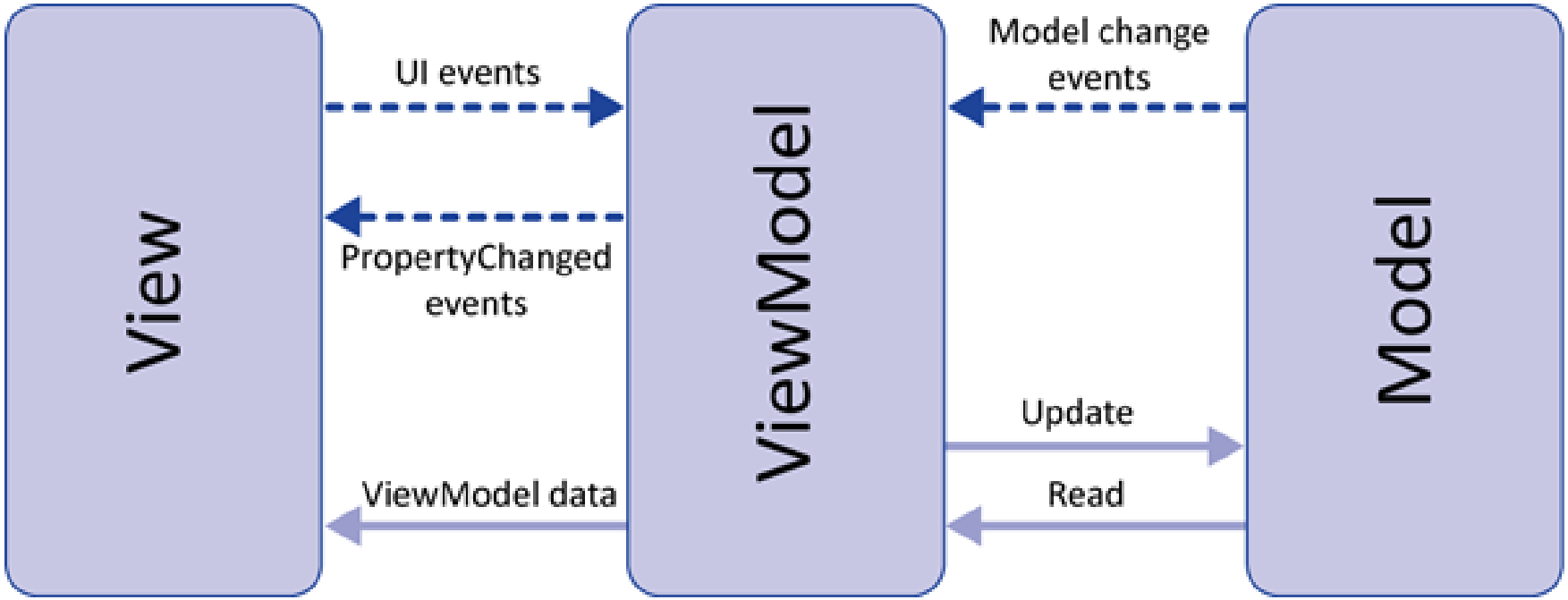
Comenzi

- Există peste 100 de comenzi predefinite grupate în 5 categorii:
  - ApplicationCommands (ex. Cut, Paste, New).
  - NavigationCommands.
  - MediaCommands.
  - EditingCommands.
  - ComponentCommands.
- Definirea de comenzi proprii presupune implementarea celor două metode (*Execute()* și *CanExecute()*).
- Ex: binding buton-textbox pentru operația "Cut". Butonul se activează doar dacă există un text selectat.

```
<Button Content="Cut" Height="23" ... Command="Cut"
 CommandTarget="{Binding ElementName=textBox2}"/>
```

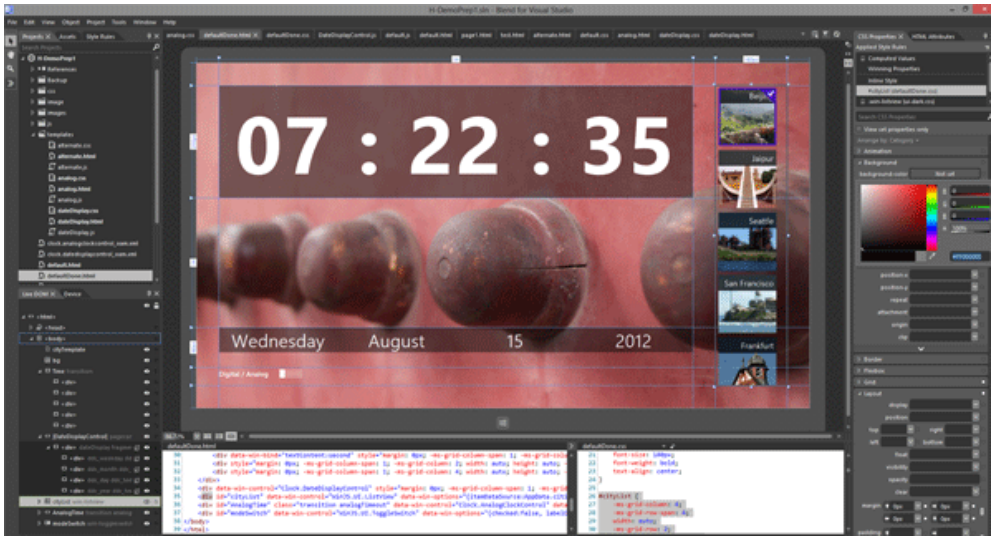
Binding la obiecte .NET și MVVM

- Binding-ul se poate realiza și al obiecte .NET.
- Se utilizează *ObjectDataProvider*.
- MVVM - Model View ViewModel.



Alte tehnologii

- Microsoft Blend.



- Microsoft Silverlight.